



HAL
open science

A replay driven model of spatial sequence learning in the hippocampus-prefrontal cortex network using reservoir computing

Nicolas Cazin

► **To cite this version:**

Nicolas Cazin. A replay driven model of spatial sequence learning in the hippocampus-prefrontal cortex network using reservoir computing. Neuroscience. Université de Lyon, 2018. English. NNT : 2018LYSE1133 . tel-01986362

HAL Id: tel-01986362

<https://theses.hal.science/tel-01986362>

Submitted on 18 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N°d'ordre NNT :

2018LYSE1133

THESE de DOCTORAT DE L'UNIVERSITE DE LYON

Opérée au sein de
L'Université Claude Bernard Lyon 1

**Ecole Doctorale 476
(Neurosciences et Cognition)**

Spécialité de doctorat : Neurosciences
Discipline : Informatique

Soutenue publiquement le 12/07/2018, par :
(Nicolas Jules-Henri CAZIN)

A replay driven model of spatial sequence learning in the hippocampus-prefrontal cortex network using reservoir computing

Devant le jury composé de :

FOURNERET Pierre	Professeur des universités – Praticien hospitalier	Président
DAMBRE Joni	Professeure/GHENT University	Rapporteure
ALEXANDRE Frederic	Directeur de recherche INRIA/CNRS UMR 5293	Rapporteur
BEN HAMED Suliann	Directrice de recherche ISC/CRNS UMR 5229	Examinatrice
FOURNERET Pierre	Professeur des universités – Praticien hospitalier	Examineur
POUCET Bruno	Directeur de recherche LNC/ CNRS UMR 7291	Examineur
DOMINEY-Peter Ford	Directeur de recherche / INSERM U1208	Directeur de thèse

UNIVERSITE CLAUDE BERNARD - LYON 1

Président de l'Université

Président du Conseil Académique

Vice-président du Conseil d'Administration

Vice-président du Conseil Formation et Vie Universitaire

Vice-président de la Commission Recherche

Directrice Générale des Services

M. le Professeur Frédéric FLEURY

M. le Professeur Hamda BEN HADID

M. le Professeur Didier REVEL

M. le Professeur Philippe CHEVALIER

M. Fabrice VALLÉE

Mme Dominique MARCHAND

COMPOSANTES SANTE

Faculté de Médecine Lyon Est – Claude Bernard

Directeur : M. le Professeur G.RODE

Faculté de Médecine et de Maïeutique Lyon Sud – Charles Mérieux

Directeur : Mme la Professeure C. BURILLON

Faculté d'Odontologie

Directeur : M. le Professeur D. BOURGEOIS

Institut des Sciences Pharmaceutiques et Biologiques

Directeur : Mme la Professeure C. VINCIGUERRA

Institut des Sciences et Techniques de la Réadaptation

Directeur : M. X. PERROT

Département de formation et Centre de Recherche en Biologie Humaine

Directeur : Mme la Professeure A-M. SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies

Directeur : M. F. DE MARCHI

Département Biologie

Directeur : M. le Professeur F. THEVENARD

Département Chimie Biochimie

Directeur : Mme C. FELIX

Département GEP

Directeur : M. Hassan HAMMOURI

Département Informatique

Directeur : M. le Professeur S. AKKOUCHE

Département Mathématiques

Directeur : M. le Professeur G. TOMANOV

Département Mécanique

Directeur : M. le Professeur H. BEN HADID

Département Physique

Directeur : M. le Professeur J-C PLENET

UFR Sciences et Techniques des Activités Physiques et Sportives

Directeur : M. Y.VANPOULLE

Observatoire des Sciences de l'Univers de Lyon

Directeur : M. B. GUIDERDONI

Polytech Lyon

Directeur : M. le Professeur E.PERRIN

Ecole Supérieure de Chimie Physique Electronique

Directeur : M. G. PIGNAULT

Institut Universitaire de Technologie de Lyon 1

Directeur : M. le Professeur C. VITON

Ecole Supérieure du Professorat et de l'Education

Directeur : M. le Professeur A. MOUGNIOTTE

Institut de Science Financière et d'Assurances

Directeur : M. N. LEBOISNE

Contents

Abstract.....	8
Acknowledgments.....	9
Glossary	10
1. Introduction.....	11
1.1. Prefrontal cortex and hippocampus in rodent navigation.....	14
1.2. Representing sequential behavior in neural networks.....	16
1.3. Hypothesis and objectives.....	17
2. Contributions.....	24
2.1. Proof of concept.....	26
2.1.1. C++/QT simulator.....	27
2.1.2. Complex sequence learning	30
2.1.3. Learning from random replay	33
2.1.4. Overlap.....	34
2.1.5. Noisy conditions	36
2.1.6. Reliability over multiple time scales.....	36
2.1.7. Sparse snippets replay.....	37
2.1.8. Consolidation of complex sequences.....	37
2.2. Towards a more realistic model	39
2.2.1. Use trajectories performed by a rat.....	39
2.2.2. Use of a place-cell model.....	40
2.2.3. Generate a set of snippets from multiple sequences based on reward.....	42
2.2.4. Defining a sensory-motor loop	44
2.3. High performance computing simulator	47
2.3.1. Use case	50
2.3.2. Design	51
2.3.3. Components	58
2.3.4. Language bindings	59
2.3.5. Deployment.....	60
2.4. Prefrontal cortex reservoir network learns to create novel efficient navigation sequences by concatenating place-cell snippets replayed with spatial credit assignment in hippocampus.....	63
2.5. Real-time sensory-motor integration of hippocampal place-cell replay and prefrontal sequence learning in a simulated rat robot (Experimental results to come).....	144
2.6. Neurodynamic properties of the semantic network.....	179
3. Discussion.....	212
Appendix.....	219

Communications	219
Chicago, IL-U.S.A 2015 Poster (SFN)	220
Chicago, IL-U.S.A 2015 Poster (SNL)	221
Paris-FRANCE 2016 Poster (CRCNS)	222
San Diego, CA-U.S.A 2016 Dynamic Poster (SFN)	223
Rome-ITALY 2017 Presentation	224
Related work	245
Bibliography	263

Figures

Figure 1: Panel A shows a picture of the rat performing the TSP related task and panel B represents a two dimensional sketch of the optimal trajectory and the encountered place fields 12

Figure 2: Place cell sequences experienced during behavior are replayed in both the forward and reverse direction during awake SPW-Rs. Spike trains for 13 neurons with place fields on the track are shown before, during and after a single traversal. Sequences that occur during running (center) are reactivated during awake SPW-Rs. Forward replay (left inset, red box) occurs before traversal of the environment and reverse replay (right inset, blue box) after. The CA1 local field potential is shown on top and the animal’s velocity is shown below. From (Carr, Jadhav, and Frank 2011). 16

Figure 3: A real life representation of the traveling salesman problem: minimizing the distance travelled while joining main U.S. cities..... 17

Figure 4: A simplified instance of the TSP problem as an undirected graph. Five cities represented by letters A, B, C, D and E are linked by routes, represented by a segment with a width proportional to the distance between its boundaries..... 18

Figure 5: An optimal trajectory between feeders ABCDE is represented in panel A. Panel B, C and D display non-optimal trajectories that contain a sub trajectory of the ABCDE trajectory. The sub trajectory shared with the ABCDE trajectory is displayed in red and the non-optimal parts in blue. Panel B contains the ABCED, panel C the EBCDA trajectory and panel D the BACDE trajectory. 19

Figure 6: Organization of the different functional modules used for defining an animat supposed to mimic the rat’s behavior while incrementally solving the TSP problem..... 20

Figure 7 : OFFLINE modes model learning through STDP during SPW-R between each trial when the animal is decoupled from the action-perception cycle and ONLINE modes model generation of a trajectory based on prior OFFLINE learning and action perception cycles. Panel A shows the hippocampus model learning recursively to generate snippets related to a reward, based on forward and reverse replay of recently experienced place cell activation sequences. Panel B shows the synaptic weights being adjusted in order to allow Striatum model to trigger the appropriate action relative to a stimuli sequence encoded by the Prefrontal cortex model. Panel C and D shows the online modes used during navigation. 23

Figure 8: Internal Representations in the reservoir. Panel A shows the raster of simulated hippocampus place-cell neurons activation over time. Panel B represents the raster of the reservoir’s neurons. Panel C depicts the hippocampus, prefrontal cortex and striatum as we modelled them and features a supplementary artificial neuron group called equivalent reservoir whose raster is represented in panel D. An equivalent reservoir neuron represents in fact a principal component, as computed by the PCA algorithm. Panel E represents the raster observed in panel D as a 3D trajectory resulting from the 3 first principal components..... 25

Figure 9: Fréchet Distance. Panel A shows the free space diagram used during the computation of the Fréchet distance between ABCED and ABCDE trajectories. X and Y axis represent the discrete steps between points of ABCED and ABCDE trajectories. Z axis relates in meter the coupling distance between two points. The coupling sequence is represented with a blue line following the optimal alignment between the two trajectories (i.e. sequences of points). Panel B represents the paired points of ABCDE and ABCED trajectories. During the common prefix ABC, curves are relatively similar and the paired points are located close to each other. This is related by the diagonal part of the coupling sequence in panel A. Then the diverging parts of ABCDE and ABCED are still paired but we can clearly observe a one to many relationship, near point (0.6,-0.1) and the free space diagram relates it by variations along the ABCED steps axis and no variation along the ABCDE steps axis. 26

Figure 10: The early version of the reservoir model. The snippet replay algorithmic module displayed in red generates dynamically a dataset used for training the model. The blue arrow represents the feedback link used in autonomous generation mode. 27

Figure 11: An extract from a scenario XML file describing a simulation.....	28
Figure 12: Main panel of the LiveViewer application and widgets controlling model parameters and displaying data plots in real time.....	29
Figure 13: Use case diagram of the RRNN Simulator.....	30
Figure 14:A complex and long stimuli sequence of 44 neurons over 540 timesteps	31
Figure 15: Representation of the transition matrix between symbols of the stimuli sequence.....	31
Figure 16: Reservoir's neurons states and transitions are represented as a 2D trajectory by displaying the first two principal components of the neurons activation raster.	32
Figure 17: Raster of the readout neurons during training, priming and autonomous generation	33
Figure 18: Raster of stimuli sequences. Random subsequences during the model training and the whole sequence during generation.....	33
Figure 19: Reservoir neurons when exposed to a random replay of snippets. States are displayed as a 2D trajectories whose coordinates are the two first principal components as computed by the PCA algorithm ..	34
Figure 20: Raster of the readout neurons when exposed to random replay of snippets during training and primed by the whole sequence.....	34
Figure 21: Heat map representing the relationship between snippet length (Y-axis), snippet number (X-axis) and snippet overlap (color)	35
Figure 22: Model performances represented as a surface. Snippets overlap and training epochs have an influence on prediction error.....	35
Figure 23: Input (Panels A and C) and output (Panels B and D) neurons rasters during model training and testing with an additive noise.....	36
Figure 24: Demonstration of the model's sequence generation ability over multiple timescales while trained at a faster timescale.....	37
Figure 25: Raster of the readout neurons when trained with snippets sparse snippets and the resulting generated sequence.....	37
Figure 26: Raster of input neurons when exposed to the whole sequence and to a random replay of snippets of the whole sequence with an additive noise	38
Figure 27: Raster of readout neurons generating the whole sequence and learning the whole sequence from a random replay of sparse, noisy and complex snippets. White circles emphasize ambiguous transitions present in the training set. It reflects the difficulty of this sequence learning task	38
Figure 28: Trajectory performed by a rat (Panel A) and an idealized version (Panel B)	39
Figure 29: Trajectory performed by a rat as captured by the visual tracker. Panel B represents the corresponding place-cell activation pattern raster, panel C shows the gradient modulus of the position, panel D represents the condensed and resampled trajectory and panel E shows The corresponding simplified place-cell activation sequence.	41
Figure 30: Place field model applied to a trajectory and representation of a snippet.....	42
Figure 31: Trajectories performed by the rat over successive trials	42
Figure 32: Snippet replay likelihood used in the snippet generation procedure.....	43
Figure 33: BACDE trajectory and the frequency of different parts being represented through random replay	44
Figure 34 : Successive trajectories of the rat in configuration 40.....	45
Figure 35: Mean square error histograms for different conditions evaluated in non-autonomous mode.....	46
Figure 36: Mean square error histograms for different conditions evaluated in autonomous mode	47
Figure 37: Use case diagram of the TRN solution.....	50
Figure 38: The Model View Presenter architecture used by TRN.....	51
Figure 39: Different interfaces organized as different namespace	52
Figure 40: Class diagram of the overall model simulating multiple animats	54

Figure 41: An example of the monitor plugin displaying performances information on the standard output.	55
Figure 42: An example of the Display plugin rendering multiple trajectories computed by several simulators using 4 GPUs	56
Figure 43: Class diagram of the Presenter layer and related artifacts.....	57
Figure 44: Component diagram of the TRN solution	58
Figure 45: The TRN4MALTAB component provide to a MATLAB script a limited set of TRN4CPP interfaces	59
Figure 46: TRN4JAVA allows one to interface a third party simulator (SCS, a robotic simulator from USF Tampa)	59
Figure 47: Deployment on a standalone computer	60
Figure 48: Deployment of a remote Backend for offloading simulations	60
Figure 49: Cluster deployment of TRN with a TCP/IP Server	61
Figure 50: Deployment of the heterogeneous Cluster used in this thesis	62

Abstract

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 HC (hippocampus) and prefrontal cortex (PFC) are taking place. Recent studies (De Jong et al. 2011; Carr, Jadhav, and Frank 2011) show that spatial navigation in the rat hippocampus involves the replay of place-cell firing during awake and sleep states generating small contiguous subsequences of spatially related place-cell activations that we will call “snippets”. These “snippets” occur primarily during sharp-wave-ripple (SPW-R) 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 propose to develop an integrated model of HC and PFC that is able to form place-cell activation sequences based on snippet replay. The proposed collaborative research will extend existing spatial cognition model for simpler goal-oriented tasks (Barrera and Weitzenfeld 2008; Barrera et al. 2015) with a new replay-driven model for memory formation in the hippocampus and spatial sequence learning and recall in PFC.

In contrast to existing work on sequence learning that relies heavily on sophisticated learning algorithms and synaptic modification rules, we propose 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. This reservoir computational model will consolidate snippets into larger place-cell activation sequences that may be later recalled by subsets of the original sequences.

The proposed work is expected to generate a new understanding of the role of replay in memory acquisition in complex tasks such as sequence learning. That operational understanding will be leveraged and tested on an embodied-cognitive real-time framework of a robot, related to the animat paradigm (Wilson 1991). The originality and contribution of our proposed work include:

- the use of awake hippocampal replay to create place-cell activation sequences of valid trajectories (snippets)
- the use of reservoir computing to learn place-cell activation sequences using inputs generated by the hippocampus model
- the constraining of the model using electrophysiological data in rats
- the use of the resulting model in an animat
- the use of behavioral data for training the model and comparing the generated trajectories

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Peter-Ford Dominey for the continuous support of my Ph.D study and related research, for his patience and motivation. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Joni Dambre, Dr Suliann Ben Hamed, Dr Bruno Poucet, Prof. Pierre Fournieret and Dr Frederic Alexandre for their insightful comments and encouragement, but also for the difficult and wise questions, which incited me to widen my research from various perspectives.

I thank my fellow labmates in for the stimulating discussions, and for all the fun we have had in the last four years. Also I thank my colleagues Weitzenfeld & al from university of south Florida/Tampa and Fellous & al from university of Arizona/Tucson. The discussions we had and their comments played an important role.

Last but not the least; I would like to thank my relatives, my friends and family for supporting me spiritually throughout writing this thesis and my life in general.

Glossary

API	Applications programming interface
CPU	Central Processing Unit
CLI	Command Line Interface
ECC	Error Correcting Code
EEG	ElectroEncephaloGraphy
FPGA	Field Programmable Gate Array
Gflops	Giga floating operations per second
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HC	Hippocampus
HHMM	Hierarchical Hidden Markov Model
HMM	Hidden Markov Model
MKL	Math Kernel Library
MPI	Message Passing Interface
MVP	Model View Presenter
PCA	Principal Component Analysis
PFC	Prefrontal Cortex
SPW-R	SharP Wave ripple during Rest state
ST	STriatum
STDP	Spike-Timing Dependent synaptic Plasticity
TBPTT	Truncated Back Propagation Through Time
TCP	Transfer Control Protocol
TRN	Temporal Recurrent Network
TSP	Traveling Salesperson Problem

1. Introduction

The work accomplished during this thesis focuses on aspects of recurrent neural networks and their dynamics, and how these dynamics can encode aspects of sequential behavior.

We first propose to consider the problem of an agent exhibiting an efficient behavior and to reduce it as the problem of generating a correct sequence of sensory-motor association of an animat (Wilson 1991).

We chose to use the reservoir computing framework (Dominey 1995) as a neurobiologically plausible implementation of a sensory-motor sequences learning capability of a situated agent (Hendriks-Jansen 1996). Reservoir computing is a computational metaphor of cortico-cortical loops in prefrontal cortex (PFC) and its principle is based on maintaining a spatio-temporal high dimensional representation of input sequences through reverberations caused by recurrent connections of PFC neurons. Sensory motor associations are learnt by a model of Striatum (ST) that acts as a simple readout layer by selecting correlated spatiotemporal features provided by the PFC model in order to reconstruct a prediction of the input patterns (Rigotti et al. 2010, 2013; Enel et al., n.d.).

We thus propose in section 0 to evaluate the standard complex sequence online learning paradigm of this joint PFC-ST model and to extend it by demonstrating that it is possible to concatenate a complex sequence by learning only from a random set of smaller subsequences of stimuli called snippets. It is possible to explain this new dynamical property by observing a reduced dimension representation of the PFC model dynamics when exposed to the whole sequence and subsequences. In (Sanger 1989), the author demonstrates that generalized hebbian learning computes in fact the linear PCA and it suggests a biologically plausible implementation for the formation of subspaces relevant for a particular task. We will see in section that it is possible to model an equivalent reservoir as a linear combination of reservoir's neurons with a PCA. Learning readout synaptic weights in this space is equivalent to find the correlation coefficients between the raster of principal components neurons and expected neurons. It is also a convenient manner of representing Reservoir's neurons activations over time. Activation raster of equivalent reservoir neurons are represented as a trajectory along the two first components of a principal component analysis (PCA). It appears that trajectories and sub-trajectories of the same stimuli sequence share common sub-trajectories in the reduced state space and suggest a stationary property of the high dimensional time series provided by the PFC allowing the sequence learning to occur.

At the same time, in the lab, experiments were being done related to human cortical dynamics during a visual and narrative comprehension task. We hypothesized that the human brain states observed through an EEG signal display the same mixed selectivity property as the reservoir and succeeded to explain with the same PCA analysis technique that it is possible to learn a reliable decoding of sequential coherence for one modality (images or sentences) only from training data of the other modality. This is explained in more detail in section 2.6.

The key point in both experiments is that projecting the reservoir's neurons activation time series in a subspace congruent to a particular task allows one to represent and modify efficiently spatio-temporal data streamed by a recurrent neural network. It encourages us to formulate the following working hypothesis:

If the recurrent network (or EEG signal) is performing highly similar operations over two successive trials, then the spatiotemporal patterns of activity (as revealed by PCA, a linear combination a neurons) will be similar as well.

In (Preston and Eichenbaum 2013), the authors study the interplay between Hippocampus and Prefrontal cortex and their roles in the memory consolidation process. Their understanding entails the notion of schema, ‘introduced to cognitive psychology by (Piaget 1967) and (Bartett 1932) in their efforts to understand how new information is integrated with pre-existing knowledge.’ A schema can be viewed as ‘any organized network of overlapping representations that has the following properties: first, new information is better remembered when it fits within a pre-existing schema; second, new information that challenges schema organization may cause modification of the existing schema or development of a new schema; and third, schemas support novel inferences between indirectly related events and their generalization to new situations.’ Consolidation is understood as a ‘process in which hippocampal networks can link indirectly related elements (A and C) via the invariant common element (B) and, guided by the prefrontal strategic control of conflicting associations to create a schema (A-B-C). During subsequent memory expression, a memory cue (‘Are A and C related?’) engages prefrontal cortex to select the correct schema (A-B-C) within which the hippocampus retrieves the relevant associations (A-C via B).’

We propose to explain this understanding of memory consolidation with a computational joint model of hippocampus, prefrontal cortex and striatum (HC-PFC-ST) by revisiting an interesting experiment where during a navigation task involving a rat searching for multiple sources of food in a closed environment (see Figure 1, panel A), it has been observed (De Jong et al. 2011) a convergence of the trajectories performed by the rat towards a short route between rewarded food cups.

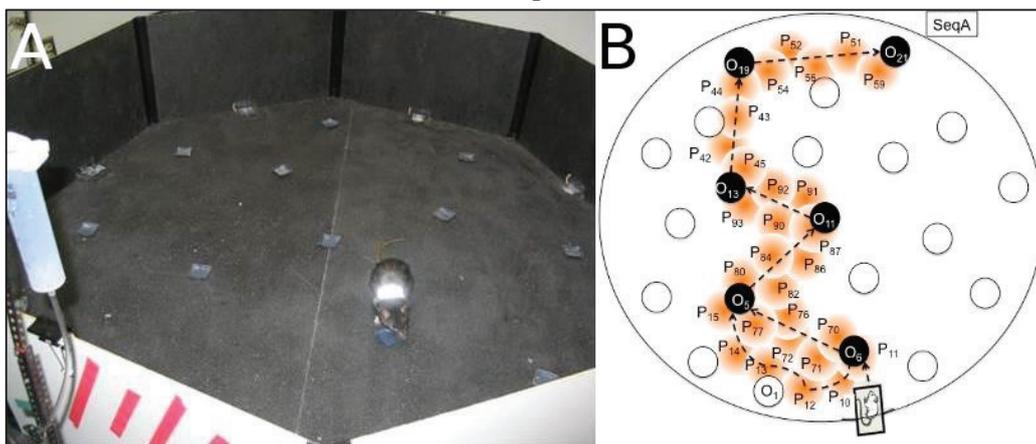


Figure 1: Panel A shows a picture of the rat performing the TSP related task and panel B represents a two dimensional sketch of the optimal trajectory and the encountered place fields

As illustrated in Figure 1, panel B, place-fields¹ are traversed in a particular order, which is characteristic of the shortest trajectory between baited feeders². This trend to generate efficient trajectories is referred to as the traveling salesperson problem (TSP). The TSP is a classical algorithmic NP-hard problem that requires an agent to visit a fixed set of locations once, minimizing the total distance traveled.

Sharp wave ripple complexes (SPW-R) occur during awake and sleep states and include the replay of subsequences of place-cells activation sequences related to the trajectories observed in previous trials (Barrera and Weitzenfeld 2008; De Jong et al. 2011). In (Gupta et al. 2010), the authors characterize the content of hippocampus replay and suggest that the activation subsequences generated by the hippocampus during SPW-R are derived from immediate and remote experiences, feature forward and reverse order of previously encountered place-cells activations and the reward plays a modulatory role in occurrence of subsequences contained in a during in SPW-R events. We focus on the role of hippocampus place-cell replay during the awake state, as the rat generates increasingly efficient trajectories between reward sites across multiple trials.

¹ Noted P_n, where n is the corresponding place-cell number.

² Noted O_k, where k is the feeder number

The indirectly related elements mentioned in (Preston and Eichenbaum 2013) corresponds to the rewarded food cups observed through the associated place-cell activations, the relationship between rewarded food cups corresponds to sub-trajectories between them. We hypothesize that:

- Hippocampus links rewarded food cups through SPW-R replay of associated place-cell activation patterns in order to form efficient sub-trajectories
- Prefrontal cortex reconciles the conflicts in associations of efficient sub-trajectories sharing a common part in order to create an efficient trajectory between rewarded food cups (schema).

We extend our scope to a HC-PFC-ST joint model where the hippocampus model emulates different features of the place-cell subsequences replay phenomenon observed during awake SPW-R between trials. It aims at:

- Demonstrating dynamical properties induced by the hippocampal replay and the effect of its different features (Foster and Wilson 2006a; Gupta et al. 2010)
- To assess if an animat embedding the HC-PFC-ST model is sufficient to explain the behavior of the rat observed in experiments (Wilson 1991; Ball et al. 2010; Foster and Knierim 2012; De Jong et al. 2011; Barrera and Weitzenfeld 2008).

A preliminary work had been done during my master's thesis about the control of a mobile robot with a chaotic random recurrent neural network with continuous time (article in preparation, see section Related work). The learning rule used for associating an action to a sequence of stimuli is related to the recursive estimation of the covariance matrix between perception and action. It emphasizes the importance of learning sensory-motor associations upon the detection of a salient event. In that particular case, a salient event is detected by considering the neuron activation that diverges from the same neuron's running average. An experiment shows that a mobile robot controlled by a random recurrent neural network can learn sensory motor associations from its previous experience and generate motion sequences in compliance with the serial structure of the stimuli by being exposed to the same stimuli sequence without having necessarily the same temporal structure (i.e. the same delay between two different stimuli). It suggests that learning during salient events only allows the model to learn and generate sensory-motor sequences with the same serial structure whilst at the same time removing the requirement of experiencing the same temporal structure. Learning the salient part of a trajectory relies in this case on the running estimate of the covariance matrix. Hippocampal replay during SPW-R occur at a faster time scale than the experienced stimuli sequences and the temporal structure is not a feature of hippocampal replay (Davidson, Kloosterman, and Wilson 2009; Nádasdy et al. 1999). It suggests that our joint HC-PFC-ST model will learn salient part of stimuli sequences and hippocampal replay will enhance several features of the stimuli sequences replayed.

In addition, conceptors (Jaeger 2014) are based on PCA which also rely on the estimate of the covariance matrix and “help explaining how conceptual-level information processing emerges naturally and robustly in neural systems”. The principle is essentially the same: Different stimuli entails differently shaped reservoir state clouds (i.e. different linear combinations of reservoir's neurons). “The ellipsoid envelopes of these clouds make conceptors”. “After driving patterns have been stored in the network, they can be selected and stably re-generated by inserting the corresponding conceptor filters in the update loop. A conceptor could be implemented by projecting observed neurons into linear neurons that represent conceptors by projecting along synaptic weights representing the PCA linear combination coefficients. Thus a conceptor is task dependent and requires a prior training on tasks being evaluated. It remains a very powerful tool that allows a single neural system to “learn, store, abstract, focus, morph, generalize, de-noise and recognize a large number of dynamical patterns”.

Estimating the covariance matrix allows one to generate appropriate representations of stimuli time-series useful, either for explaining neuron dynamics or for altering them in order to solve difficult learning problem.

It encourages us to use the PCA as a tool for explaining how the dynamics of a recurrent neural network allows the model to learn multiple and complex sensory-motor associations and help to solve a challenging problem: the TSP.

1.1. Prefrontal cortex and hippocampus in rodent navigation

The hippocampus stores information during the acquisition of new memories and these memories are replayed (in this document, we use ‘replay’ and ‘reactivate’ interchangeably) during sleep as part of a memory consolidation process (Marr 1971; Stickgold and Walker 2007). Consolidation is believed to involve synaptic changes in the neocortex reflecting the integration and refinement of memory representations (McClelland, McNaughton, and O’Reilly 1995; Schwindel and McNaughton 2011) This replay involves neural populations that were active during a task immediately preceding the sleep period. In (Jadhav et al. 2012), a specific performance deficit was observed in SPW-R disrupted animals, providing “a causal link between awake hippocampal SPW-Rs and the spatial memory requirements of outbound trials”. It is possible to characterize hippocampus replay through the following features:

- Occurrence: Reactivations of specific neural activity patterns during sleep have been observed in several brain areas including the hippocampus, amygdala, neocortex and striatum (Bendor and Wilson 2012; Carr, Jadhav, and Frank 2011; Euston, Tatsuno, and McNaughton 2007; Foster and Wilson 2006b; Hoffman and McNaughton 2002; Ji and Wilson 2007; Karlsson and Frank 2009; Kudrimoti, Barnes, and McNaughton 1999; Lee and Wilson 2002; Nádasdy et al. 1999; Pennartz 2004; Peyrache et al. 2009; Popa et al. 2010; Ribeiro et al. 2004; Sutherland and McNaughton 2000; Tatsuno, Lipa, and McNaughton 2006; Cutsuridis and Hasselmo 2011; Cutsuridis and Taxidis 2013). 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, Jadhav, and Frank 2011; Davidson, Kloosterman, and Wilson 2009; Diba and Buzsáki 2007a; Gupta et al. 2010; Jadhav et al. 2012; K. Friston, Breakspear, and Deco 2012; Karlsson and Frank 2009).
- Place-cells: 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. Some of the most important models developed in the past years include those of (Burgess, Recce, and O’Keefe 1994; Sharp, Blair, and Brown 1996; Redish and Touretzky 1997; Guazzelli et al. 1998; Arleo and Gerstner 2000; P. Gaussier et al. 2002; Filliat and Meyer 2002; Arleo, Smeraldi, and Gerstner 2004; Milford and Wyeth 2010; Dollé et al. 2010; Alvernhe, Sargolini, and Poucet 2012; Caluwaerts et al. 2012; Barrera and Weitzenfeld 2008)
- Time cells: In (Eichenbaum 2014), the author distinguishes hippocampus place-cells from hippocampus time-cells. The latter are characterized by their ability to “parse temporally defined periods into representations of specific moments (‘time fields’), much as place-cells parse spatially defined environments into representations of specific locations (place fields)”. Finally, the author suggests that the fundamental function of the hippocampus is to establish spatio-temporal frameworks for organizing memories. Place-cells and time-cells are not distinct ‘cell-types’ but instead, the same population of hippocampal neurons encodes both the spatial and temporal regularities of experience. Hippocampus place-cells and time-cells are seen as features of hippocampal neurons associated with spatial or temporal dimensions of the context in which learning occurs.
- Time scale: 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 (SPW-R). Different subsets of cells reactivate in different SPW-Rs, each cell emitting only a few spikes. The inter-spike 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 and offline synaptic modifications that will eventually

lead to the consolidation and integration of specific memory items. (Davidson, Kloosterman, and Wilson 2009)

- Reward: Interestingly, the presence of rewards increases replay in hippocampus and ventral striatum (Lansink et al. n.d.; Singer and Frank 2009), suggesting an interaction between reinforcement learning and replay. This enhanced reactivation in response to reward could be a mechanism to bind rewarding outcomes to the experiences that precede them. Awake SPW-R reactivates coherent elements of the experiences that are associated with the paths to and from the rewarded location. Their findings, in contrast, indicate that reward plays a special role in modulating the reactivation of cells associated with recent experiences. Their findings suggest that in the rodent hippocampus, activity following a reward specifically relates to the sequence of locations the animal traversed on the way to the reward. These observations indicate that reward increases the likelihood of reactivation for all cells. Therefore, the specific spatial sequence the animal traversed strongly influences which cells will be active during SPW-Rs, while the presence or absence of reward modulates the amount and strength of reactivation.
- Direction: 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 (Foster and Wilson 2006b; Diba and Buzsáki 2007b)
- Remote: Hippocampal replay is not limited to recent sensory experience and might include events experienced less recently (Karlsson and Frank 2009; Foster and Wilson 2006b; Davidson, Kloosterman, and Wilson 2009)

Underlying Neuroanatomy:

The model developed in this thesis provides a possible explanation of mechanisms that allow PFC and hippocampus to interact to perform path optimization to solve the TSP. This implies functional connectivity between these two structures. In a recent review of hippocampal–prefrontal interactions in memory-guided behavior (Shin and Jadhav 2016) outlined a diverse set of direct and indirect connections that allow bi-directional interaction between these structures. Principal direct connections to PFC originate in the ventral and intermediate CA1 regions of the hippocampus (Cenquizca and Swanson 2007). Indirect connections between hippocampus and PFC pass via medial temporal lobe (subiculum, entorhinal cortex, peri- and post-rhinal cortex) (Delatour and Witter 2002) and the nucleus reuniens (Vertes et al. 2007). Thus while we will not provide a more extensive review, these studies and reviews allow us to consider that there are anatomical pathways supporting bi-directional interaction between PFC and hippocampus. The model developed in this thesis also demonstrates the crucial role of SPW-R replay of “snippets” of previously experienced trajectories in this sequence consolidation and optimization. While SPW-R replay has more traditionally been examined during sleep, there is now increasing evidence of its vital role in the awake state, between trials, in spatial learning tasks (reviewed in (Carr, Jadhav, and Frank 2011)). Figure 2 illustrates awake SPW-R

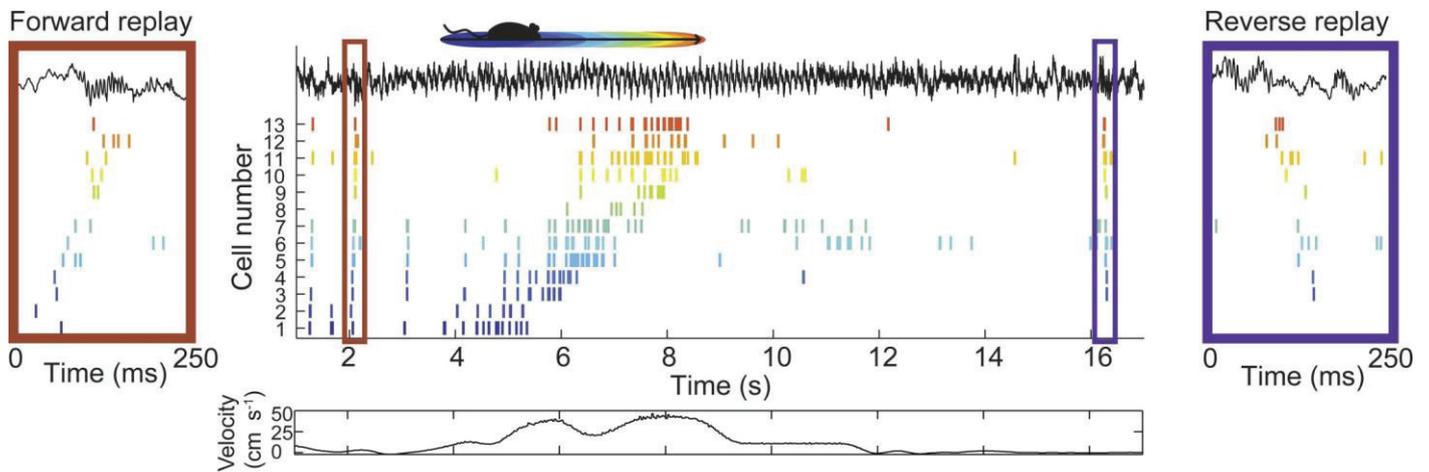


Figure 2: Place cell sequences experienced during behavior are replayed in both the forward and reverse direction during awake SPW-Rs. Spike trains for 13 neurons with place fields on the track are shown before, during and after a single traversal. Sequences that occur during running (center) are reactivated during awake SPW-Rs. Forward replay (left inset, red box) occurs before traversal of the environment and reverse replay (right inset, blue box) after. The CA1 local field potential is shown on top and the animal's velocity is shown below. From (Carr, Jadhav, and Frank 2011).

1.2. Representing sequential behavior in neural networks

Reservoir computing refers to a class of neural network models in computational neuroscience and machine learning (Mantas Lukoševičius and 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 task contexts (e.g. sequence recognition, prediction, classification). The first instantiation of such models was by (Dominey 1995; Dominey, Arbib, and Joseph 1995) with the reservoir corresponding to recurrent prefrontal cortical networks, and the modifiable readout connections corresponding to the corticostriatal projection, with dopamine-modified synapses. 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, 1998b; Dominey and Ramus 2000; Dominey, Inui, and Hoen 2009; Hinaut and 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 and Haas 2004). Interestingly, these reservoir properties appear to be found in cortex. Electrophysiological studies have revealed that cortical neurons in primary sensory areas (e.g. V1) have reservoir properties of fading memory (Nikolić 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, Arbib, and Joseph 1995; Rigotti et al. 2010). 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 1998a), including serial order regularities that are expressed in sequence segments (Dominey and Ramus 2000). However, so far, reservoir computing has not 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.

The most prevalent theories of how memories are formed and consolidated rely on dynamic changes in synaptic strengths and the creation of strongly connected neural assemblies. This hebbian (Hebb 1949) view

of memory has dominated the field for decades. While there is strong experimental evidence for several aspects of the theory, there are also significant difficulties that include:

- The unclear nature of fast induction and exquisite control of synaptic modification in the presence of realistic amount of neuronal and synaptic noise and unreliability
- The low probability that 2 cortical neurons are in fact connected by synapses
- The pluri-functionality of cortical neural networks whereby small groups of cells may be involved in various and seemingly distinct neural computations in addition to memory formation (how these multiple functions co-exist with memory is unclear).

1.3. Hypothesis and objectives

We propose to focus on the TSP problem, which is a classical artificial intelligence NP-hard problem that requires an agent to visit a fixed set of locations once, minimizing the total distance traveled. Figure 3 illustrates the TSP by representing the optimal trajectory linking the main cities in U.S.



Figure 3: A real life representation of the traveling salesman problem: minimizing the distance travelled while joining main U.S. cities

TSP can be modelled by a graph where vertices are cities and edges the roads linking cities. A simplified problem featuring only five feeders is represented in Figure 5:

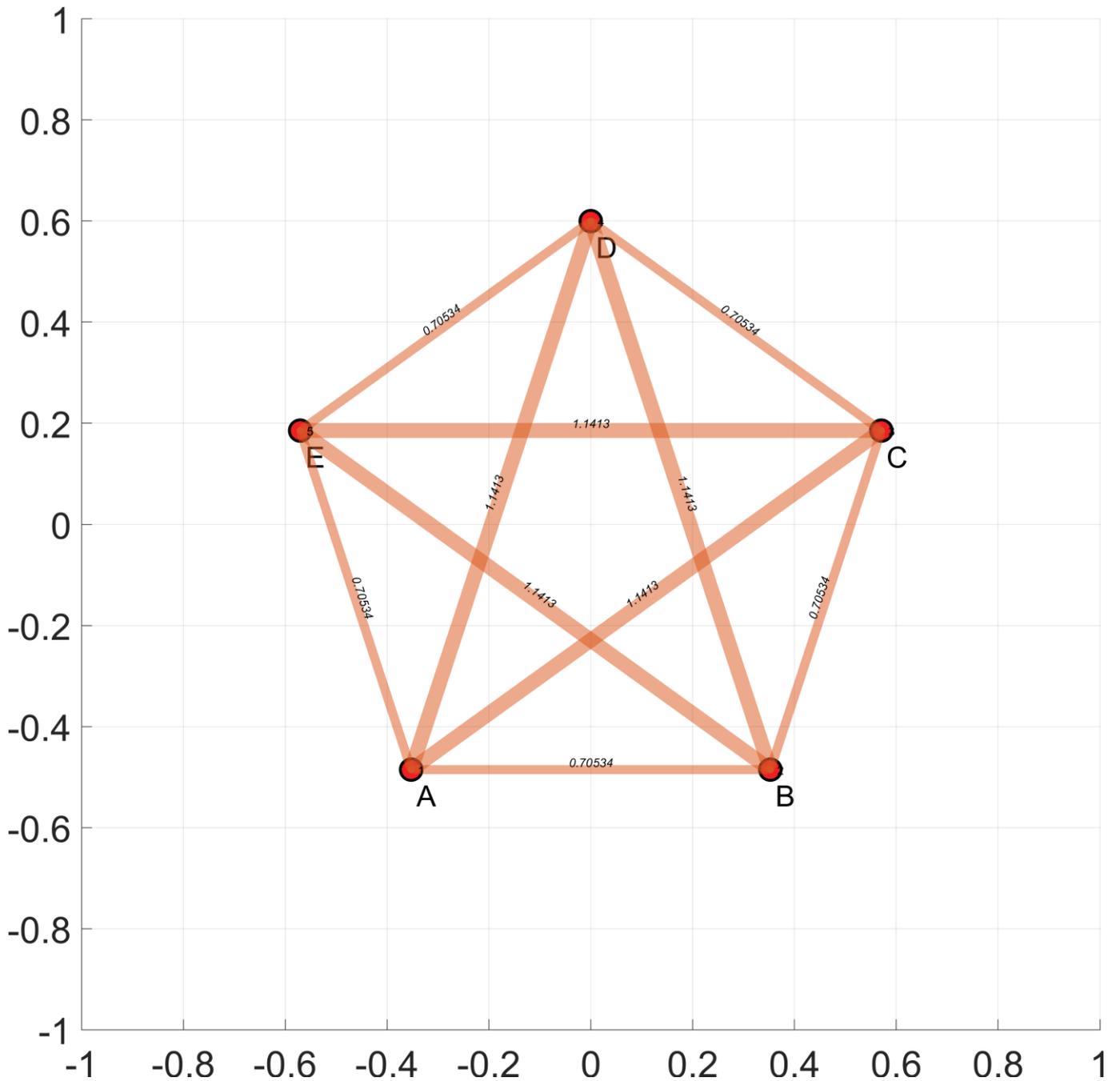


Figure 4: A simplified instance of the TSP problem as an undirected graph. Five cities represented by letters A, B, C, D and E are linked by routes, represented by a segment with a width proportional to the distance between its boundaries

In the traveling salesrat problem (De Jong et al. 2011) derived from the traveling salesman problem, distances are unknown but instead, each cup containing food (equivalent to cities in TSP) are associated to a reward information. The paths between cups are analogue to roads between cities and become available once visited by the rat across trials. It suggests an incremental update of the knowledge acquired during recent experiment, with a particular emphasis on shortest paths between cups. We propose to study this hypothesis by considering the following elementary experiment illustrated in Figure 5 and other experiments described in 2.4.

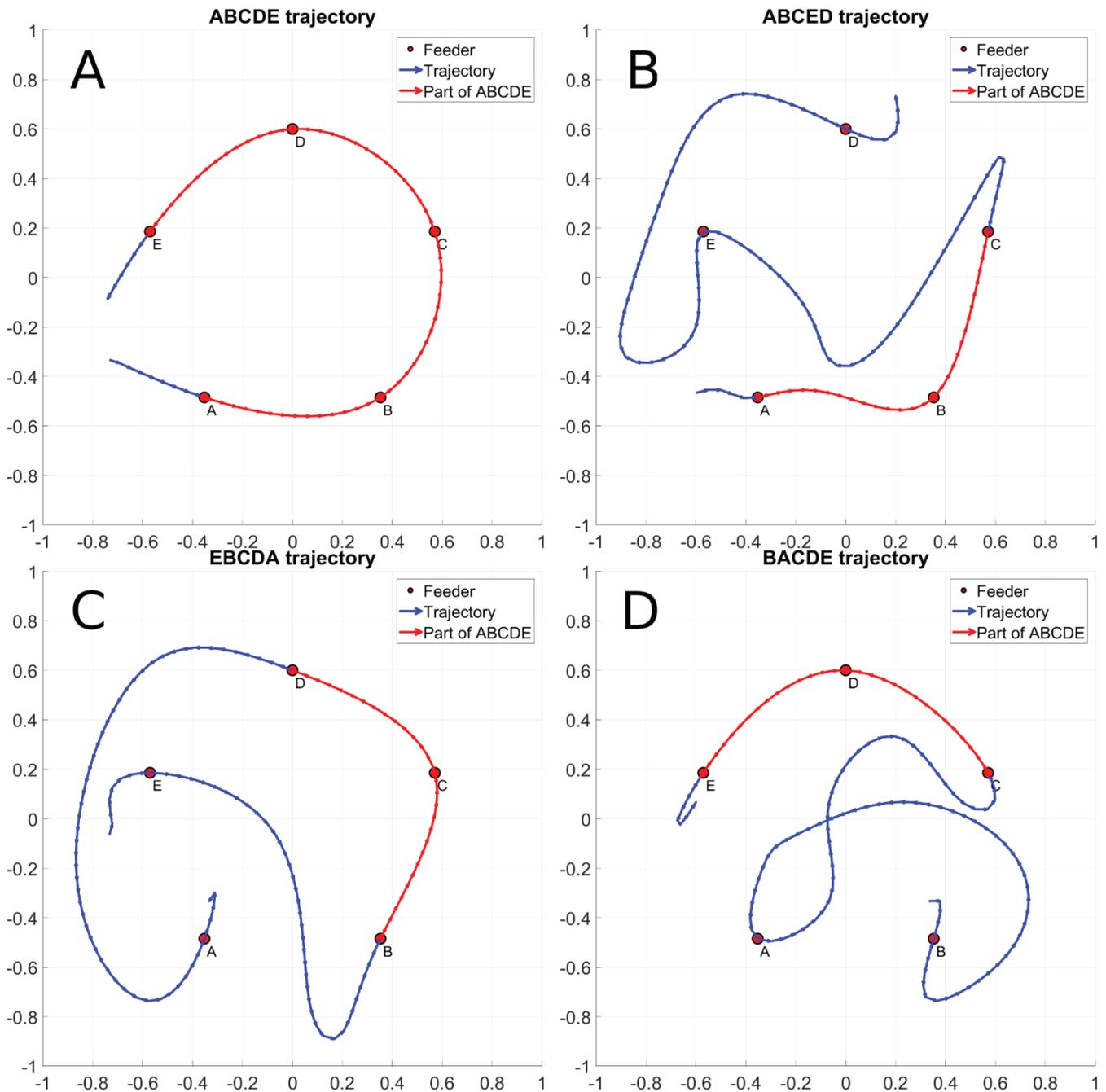


Figure 5: An optimal trajectory between feeders $ABCDE$ is represented in panel A. Panel B, C and D display non-optimal trajectories that contain a sub trajectory of the $ABCDE$ trajectory. The sub trajectory shared with the $ABCDE$ trajectory is displayed in red and the non-optimal parts in blue. Panel B contains the $ABCED$, panel C the $EBCDA$ trajectory and panel D the $BACDE$ trajectory.

Five cups containing food are laid in an open space and the rat is supposed to experience trajectories illustrated in panel B, C and D. These trajectories contain a part of the optimal trajectory illustrated in panel A. We will study how the consolidation of an optimal trajectory (Panel A) can result during awake rest state from a recombination of non-optimal trajectories containing efficient parts between reward sites (Panels, B, C, D), supported by a STDP learning mechanism occurring during SPWR.

Several navigations models based on a hippocampus model have been implemented on a mobile robot. In (Burgess, Recce, and O'Keefe 1994), the authors demonstrate the ability of a mobile robot to reach a single goal in a simple environment, based only on a hippocampus model featuring place-cells. Recurrent connections in the hippocampus are not involved nor required in the experiment. A more recent model (Barrera et al. 2015) uses hippocampus place-cells learnt by hebbian learning driven by kinesthetic information and visual landmarks. Sensory motors associations required for allowing the agent to navigate are performed with a Q-learning algorithm (Sutton and Barto 1998) which requires a discrete state space. Another robotic model

(Hirel et al. 2013) implements neural networks representing enthorinal cortex (EC), HC and PFC in order to build a cognitive map and solve a multiple goal navigation task.

We thus propose a computational model of hippocampus, prefrontal cortex and striatum that will be embedded in an animat (Wilson 1991; Filliat and Meyer 2002; Ball et al. 2010) for demonstrating the existence of a neurobiologically plausible implementation of a heuristic that solves the traveling salesrat problem. Figure 6 represents from a purely modeling point of view the organization of the different functional modules used in this thesis for simulating a virtual rat:

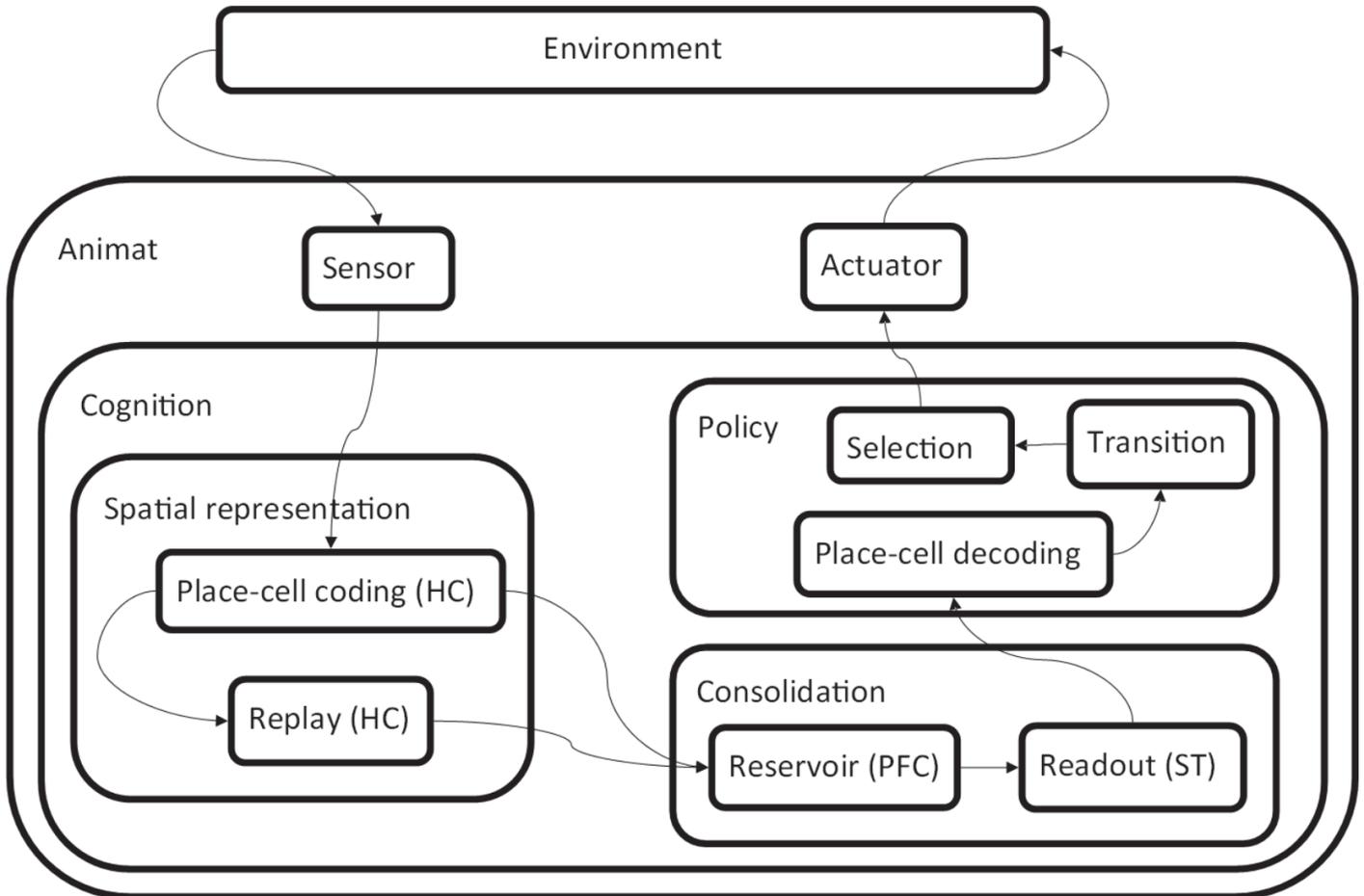


Figure 6: Organization of the different functional modules used for defining an animat supposed to mimic the rat's behavior while incrementally solving the TSP problem

The global model is hierarchical composition of other models whose structuration derives from the PerAc architecture (Philippe Gaussier and Zrehen 1995) could be described as follows:

- An environment model that consists in a 2D free space containing reward sites as illustrated in Figure 4
- An animat that is capable of perceiving the environment through its sensors and to perform an action on its environment by using its actuators. The relationship between the sensor model and the actuator model is established the cognition model, allowing the animat to perform actions given a sensory input. No reflex module linking perception to action in a straightforward manner is used in this study.
 - The sensor model is very simple in our case and consists in the perception of the current position of the animat in the environment in a global coordinate system. It might be replaced by a more elaborated and realistic model but for the sake of simplicity and a better understanding, we chose a trivial sensor model.

- The cognition model establishes dynamical relationships between perceived stimuli and possible actions. It is implemented by a set of other hierarchical models that might interact. The relationships illustrated by arrows represents strictly the sufficient model for studying memory formation mechanisms in this thesis and don't necessarily correspond to neurons projections between areas as described in neuroanatomy. A cognitive model could be implemented by an algorithm, a classifier system (Holland and Reitman 1977), a hidden Markov chain or a neural network. In our case, the cognitive model is implemented by a hybrid and hierarchical composition of algorithmic and neural network models:
 - A spatial representation model will be responsible of converting the perceived global position into internal representations suitable for solving the TSP task. It is implemented by:
 - A place-cell coding model which provides a place cell mean firing activation pattern that is characteristic of the perceived position. A 2D Gaussian place cell model is used but one can also use firing rate maps estimated experimentally and representing the spatial response of place-cells spatial response measured by electrodes. It is assimilated to a part of the hippocampus model, not represented here because it is relative to a neurobiological implementation.
 - The replay module is implemented by an algorithm that emulates only place-cells neurons rates and the replay of subsequences of place-cell activations (snippet) during awake SPW-R as a concatenation. Snippets are characterized by a duration, modelled by the number of contiguous place-cell activation patterns and a direction, forward or backward that will be modelled by the increasing or decreasing numbering of time indices contained by a snippet. A snippet will contain only place-cell activation patterns encountered during a motion supposed to occur with a constant speed. This models the time compression phenomenon observed in place-cell activation subsequences during SPW-R, derived from the place-cell activation encountered during the last trial and resulting from sensory inputs. Snippets will be drawn according to a snippet replay likelihood whose distribution might be uniform when no reward is taken into account or non-uniform and shaped by reward when available. The role of this model will be to generate subsequences of place cell activation derived from previous experience and to emphasize parts of trajectories related to a reward through a non-uniform random replay. Despite the fact that this part of the model is algorithmic, it admits an implementation based on recurrent neural networks, also observed in hippocampus (CA3). It is a part of the hippocampus model.
 - A consolidation model learns to predict the next place-cell activation, given a training dataset generated dynamically by the replay model. The generated predictions reflect the state-transitions observed in the training dataset and it is possible to emphasize a particular subset of state-transitions by over representing it in the training dataset. In our case, place-cell transitions related to a reward are emphasized by the replay module which learn and maintain a snippet replay likelihood through snippet replay. Consolidation is implemented by:
 - A reservoir model will be implemented by the reservoir-computing framework that emulates cortico-cortical loops through recurrent connections between leaky integrator neurons, representing only the mean firing rate of non-linear neurons

whose membrane potential time evolution is driven by a time constant resulting from its resistive and capacitive properties. This defines the prefrontal cortex model and its role is to combine overlapping parts of snippets replayed at random and build a transitive relationship between them by aligning the common parts in their representations.

- A readout model implements partially a striatum model. Readout model is supposed to select spatio-temporal features provided by the reservoir through learning in order to learn a desired output. In our case, the desired output is a prediction of the next place cell activation that encodes the next location of the animat.
- A policy model associates the prediction of the next place-cell activation to the action that is the most likely to realize the prediction. Possible actions are command signals compatible with the animat possible moves. It is based on three other sub models:
 - A place-cell decoding model that will provide a 2D map of probable locations given the place-cell prediction generated by the readout layer.
 - A transition model that restricts the 2D location map is restricted to areas reachable by the animat given and estimate of its current position.
 - A selection model that will select the most probable location in the restricted 2D location map.
- The actuator model is also very simple and consists in moving the agent to the predicted position.

This emulates a rudimentary sensory motor loop and might be viewed as a form of embodiment. In this model presented as a hierarchy of other models, some models are implemented by neural networks (HC, PFC, ST) and their interactions and functioning modes are not clearly specified. In (Pezzulo, Kemere, and van der Meer 2017), the authors provide a schematic illustration of different functioning modes of their model within an overall architecture for PFC-HC interactions. We propose to adapt this illustration to our joint HC-PFC-ST model in Figure 7 in order to describe the different modes used by our model for solving the TSP problem. The experimental protocol we use is inspired by (De Jong et al. 2011) and is defined as follows:

- The rat/animat performs one or more trials by making an attempt to solve the TSP problem, based on the actual state of its model. In the case of an animat, one or more already existing trajectories are simply added to a dataset that will be used as a basis for generating snippets.
- The rat/animat is put in an awake resting state at the same location before each next trial by enclosing him in a small and opaque space, preventing him from moving or seeing the arena where the next trial will occur. At this point, SPW-R events occur and small sequences of recently experienced place-cell activation patterns are replay. This contributes to the incremental update of two models and correspond to the two offline modes where recent experience is replayed through SPW-R events and models updated through a STDP mechanism:
 - The replay (snippet generation) model, implemented by our hippocampus model (HC) and illustrated in panel A by a dashed ellipsis. The reward information is propagated according to a power law to adjacent timesteps in a direction relative to the replay direction. The model is defined recursively and is described more in details in section 2.4.
 - The consolidation model implemented by the reservoir (PFC) and the readout (ST) layer and illustrated in panel C, associates a particular state to a place-cell activation sub sequence. This state is associated incrementally through an online learning rule to the next place-cell activation pattern. The dashed arrow represents the synapsis projecting reservoir neurons in readout neurons.

- Then the rat/animat behaves freely based on its newly acquired and consolidated knowledge (hypothesis) and performs one more attempt to solve the TSP problem. This online mode is illustrated in panel D and is viewed as a coordinated action perception cycle repeated several times in order to form a trajectory. The decision taken at a given time will result in a new perception.

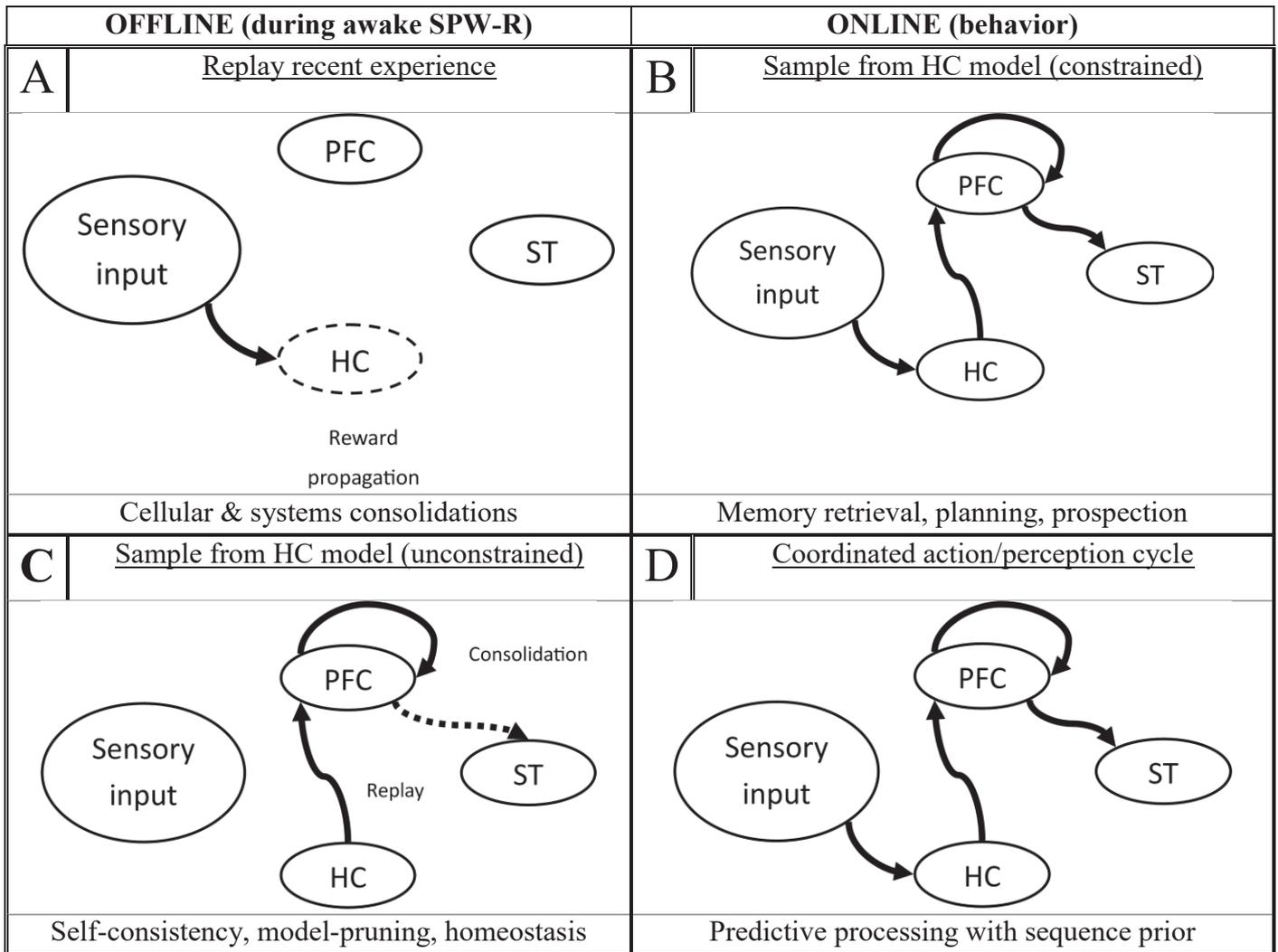


Figure 7 : OFFLINE modes model learning through STDP during SPW-R between each trial when the animal is decoupled from the action-perception cycle and ONLINE modes model generation of a trajectory based on prior OFFLINE learning and action perception cycles. Panel A shows the hippocampus model learning recursively to generate snippets related to a reward, based on forward and reverse replay of recently experienced place cell activation sequences. Panel B shows the synaptic weights being adjusted in order to allow Striatum model to trigger the appropriate action relative to a stimuli sequence encoded by the Prefrontal cortex model. Panel C and D shows the online modes used during navigation.

The mode illustrated in panel B corresponds to the preplay phenomenon observed during SPW-R when the rat is performing the task. SPW-R are shaped by current sensory input. The snippet preplayed during behavior are samples from the replay model relevant to the current task. It represents the future possible suffix trajectories, given the trajectory accomplished so far. It might be viewed as a prospective mechanism and can be used for online planning/inferencing by selecting the action associated to the most rewarding prefix snippet, suggested by preplay. We won't use this mode in this thesis and we focus on the consolidation of salient part of multiple and rewarded trajectories.

Our claim is that it is possible to observe the emergence of the optimal trajectory as a recombination of salient parts of trajectories experienced during past trials. The heuristic that supports this demonstration is biologically plausible and implementable by neural networks.

Working hypothesis are summarized as follows: An agent able to move freely within a restricted range in an environment without obstacles is controlled by a neural network model. This model includes a model of

hippocampus that will emulate forward and reverse hippocampal replay modulated by a form of reward observed during awake SPW-R. A PFC model based on reservoir computing will evaluate the hypothesis that awake hippocampus replay between trials plays a role in long term memory consolidation and allows the agent to take advantage of its previous experience from trial to trial. The ST model will allow the agent decode predictions from the PFC and contribute to an action, supposed to minimize the difference between the actual and predicted position of the agent.

2. Contributions

All the contributions in this thesis aim at investigating the role of hippocampus replay during SPW-R in memory consolidation process through simulations of models of neural networks based on neurophysiology supposed to implement a heuristic able to solve the TSP problem. These are models and do not necessarily reflect or emulate the complete set of observed anatomical features. We start from the simplest model possible of the memory consolidation process through awake SPW-R and implement only the features that allow the model to mimic the rat's behavior while solving the TSP problem. The PFC-ST model illustrated in In Figure 8 panel C is based on a recurrent neural network of leaky integrator neurons having fixed recurrent connections. Information is processed dynamically through reverberations caused by the recurrent connections. When the network is exposed to an input sequence like a place-cell activation sequence illustrated in panel A, the reservoir neurons activation pattern sequence is difficult to understand (see panel B). The readout layer associate reservoir states to an expected output through a supervised online learning rule. In this thesis, the expected output will be the place-cell activation contained by the next sample within a snippet.

We propose to use a singular value decomposition of the raster observed in panel B for implementing a principal component analysis (PCA) illustrated in panel D. It can be viewed as another layer of linear neurons whose firing rate is a linear combination of reservoir neurons. This additional layer is called the equivalent reservoir and the linear combination is chosen such that equivalent neurons provide the most orthogonal possible information. These 'orthogonal' neurons might be viewed as different axes spanning a subspace and they are called principal components. The first neuron will account for most of the information provided by reservoir neurons, while other neurons will account for a decreasing proportion of the remaining information. Thus it is possible to represent the neurons activations as a 2D/3D trajectory as illustrated in panel E and to observe them rather than the raw raster in panel B. Note that if one attempts to learn synaptic weights projecting the equivalent neurons into the readout neurons, he will find the Pearson correlation coefficients (Pearson 1896). It is an interesting equality which might be reformulated as: The readout selects iteratively spatio-temporal features of a stimuli sequence provided by the reservoir, correlated with the expected signal used in supervised learning.

We use a simulation based approach and the experimental protocol is always the same and a trial in one experiment is defined by:

- Simulate the awake SPW-R snippet replay when the rat is in a rest state
 - Instantiate a model with a fixed set of parameters
 - Define a training set based on one or more sequences representing the recent experiences of the rat
 - Train the model by exposing it to a random replay of snippets emulating different features observed and reported in literature.
- Simulate the next trial, based on the previously acquired knowledge (hypothesis)
 - Prime the model by forcing its input with the first samples of the expected sequence
 - Generate a trajectory

- Autonomously by reinjecting directly the predicted place-cell activation pattern as the input of the next simulation cycle.
- Non autonomously by injecting the expected place-cell activation pattern as the input of the next simulation cycle. This is used when the model is not sufficient for generating autonomously a sequence.
- Gather and record various observables such as:
 - Neural network states
 - Connectivity matrices
 - Time indexes generated by the replay model

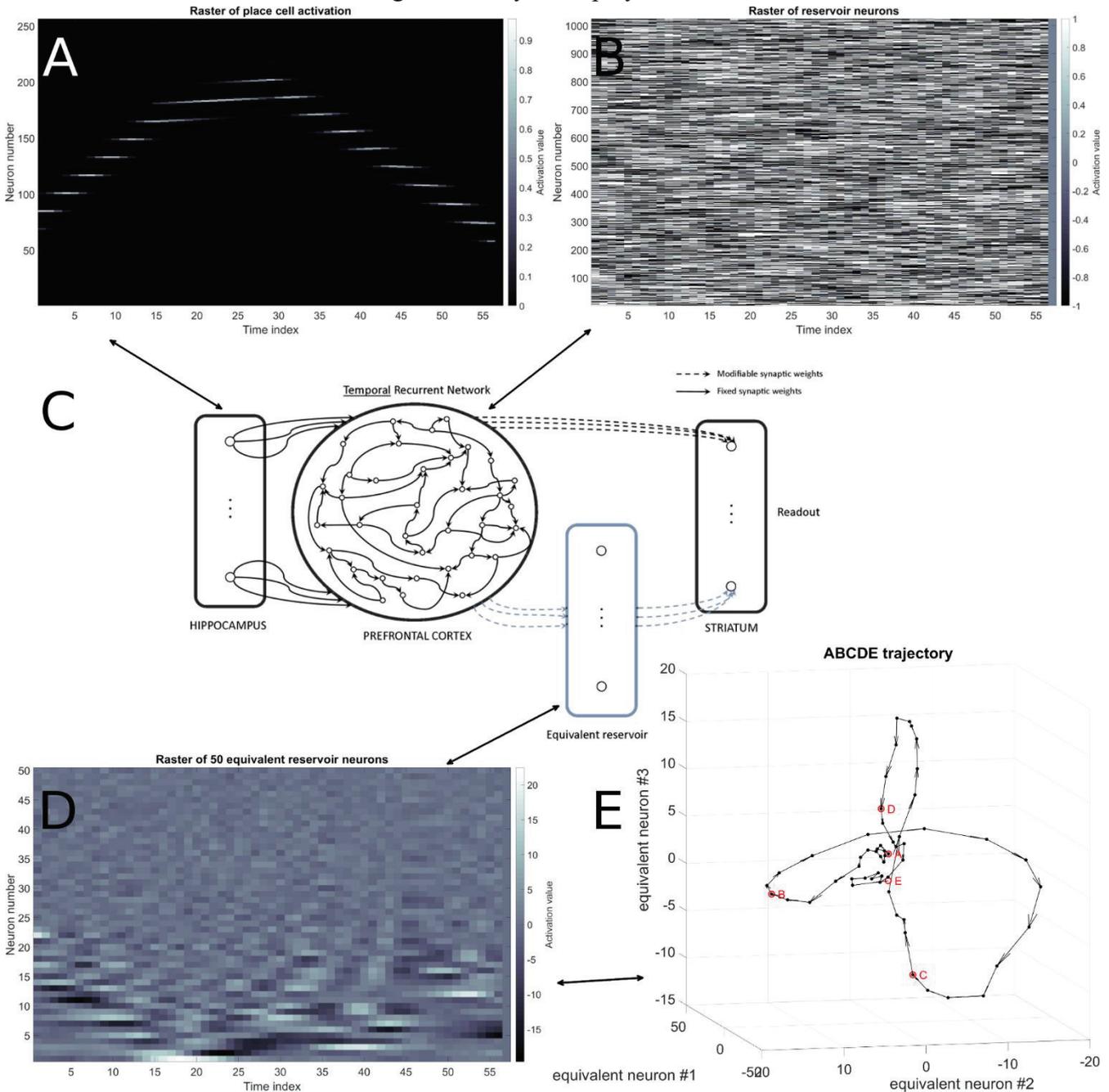


Figure 8: Internal Representations in the reservoir. Panel A shows the raster of simulated hippocampus place-cell neurons activation over time. Panel B represents the raster of the reservoir's neurons. Panel C depicts the hippocampus, prefrontal cortex and striatum as we modelled them and features a supplementary artificial neuron group called equivalent reservoir whose raster is represented in panel D. An equivalent reservoir neuron represents in fact a principal component, as computed by the PCA algorithm. Panel E represents the raster observed in panel D as a 3D trajectory resulting from the 3 first principal components.

It is thus possible to extract representations from the recorded data as depicted in Figure 8 for example. Some representations might be meaningful and will allow one to explain the phenomenon observed through the

representations. The model will be adjusted based on the numerous and informative observations until the model is able to predict and explain expected observations having a higher level of abstraction. This very general process describing the scientific approach can be as well implemented by different models in a hierarchical manner, exactly as illustrated in Figure 6. The free energy/active inference framework (K. Friston 2009) is an elegant, rigorous and unifying theoretical framework that allows one to create or extend hierarchical Bayesian models of the brain (K. Friston 2008; Pezzulo, Rigoli, and Friston 2018). We will make a particular effort to formulate our models, measures and explanations in order to be easily translatable in Bayesian equations.

From the top level of abstraction (behavior), a very useful metric for comparing trajectories regardless their phase locking and dimensionality (2D trajectory, place-cell activation sequence, neurons rasters) is the Fréchet distance. Figure 9 illustrates the Fréchet distance between sequences ABCDE and ABCED also observed in Figure 5. In each experiment, the trajectory generated by our model is compared to the optimal/idealized sequence with a Fréchet distance.

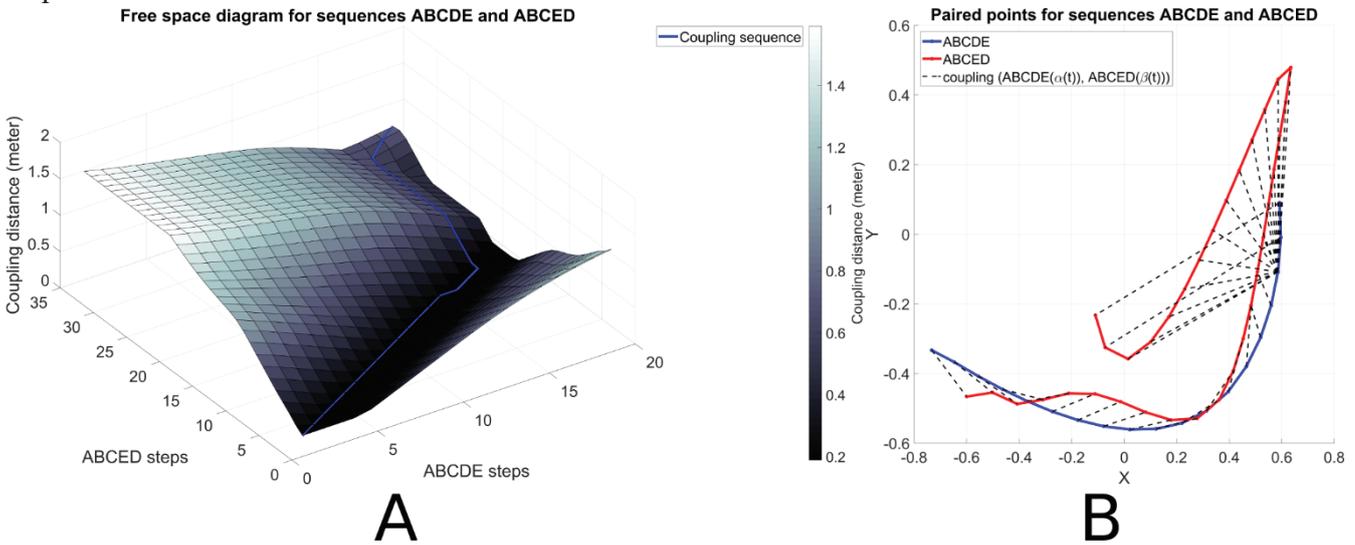


Figure 9: Fréchet Distance. Panel A shows the free space diagram used during the computation of the Fréchet distance between ABCED and ABCDE trajectories. X and Y axis represent the discrete steps between points of ABCED and ABCDE trajectories. Z axis relates in meter the coupling distance between two points. The coupling sequence is represented with a blue line following the optimal alignment between the two trajectories (i.e. sequences of points). Panel B represents the paired points of ABCDE and ABCED trajectories. During the common prefix ABC, curves are relatively similar and the paired points are located close to each other. This is related by the diagonal part of the coupling sequence in panel A. Then the diverging parts of ABCDE and ABCED are still paired but we can clearly observe a one to many relationship, near point (0.6,-0.1) and the free space diagram relates it by variations along the ABCED steps axis and no variation along the ABCDE steps axis.

It allows a global optimization algorithm to find an optimal set of parameters for instantiating an overall model able to perform in the required conditions (see section 2.4).

2.1. Proof of concept

The reservoir-computing framework allows complex sequence learning and generation at the price of several parameters that might interact between each other in manner that is difficult to characterize and exhibit unforeseen dynamic properties. Dynamic properties of snippet replay are exactly what we want to observe. Some guidelines about echo state network have been written (Mantas Lukoševičius and Jaeger 2009) and are a good starting point. For going further on the particular problem of sequence learning through random replay of snippets, we need to write a simulator of our model that provides to the user the ability to modify parameters and to observe quickly the effects of a parameter change on the screen. We propose to write an interactive simulator based on C++/QT and to explore snippet driven learning properties of a joint PFC-HC model illustrated in Figure 10.

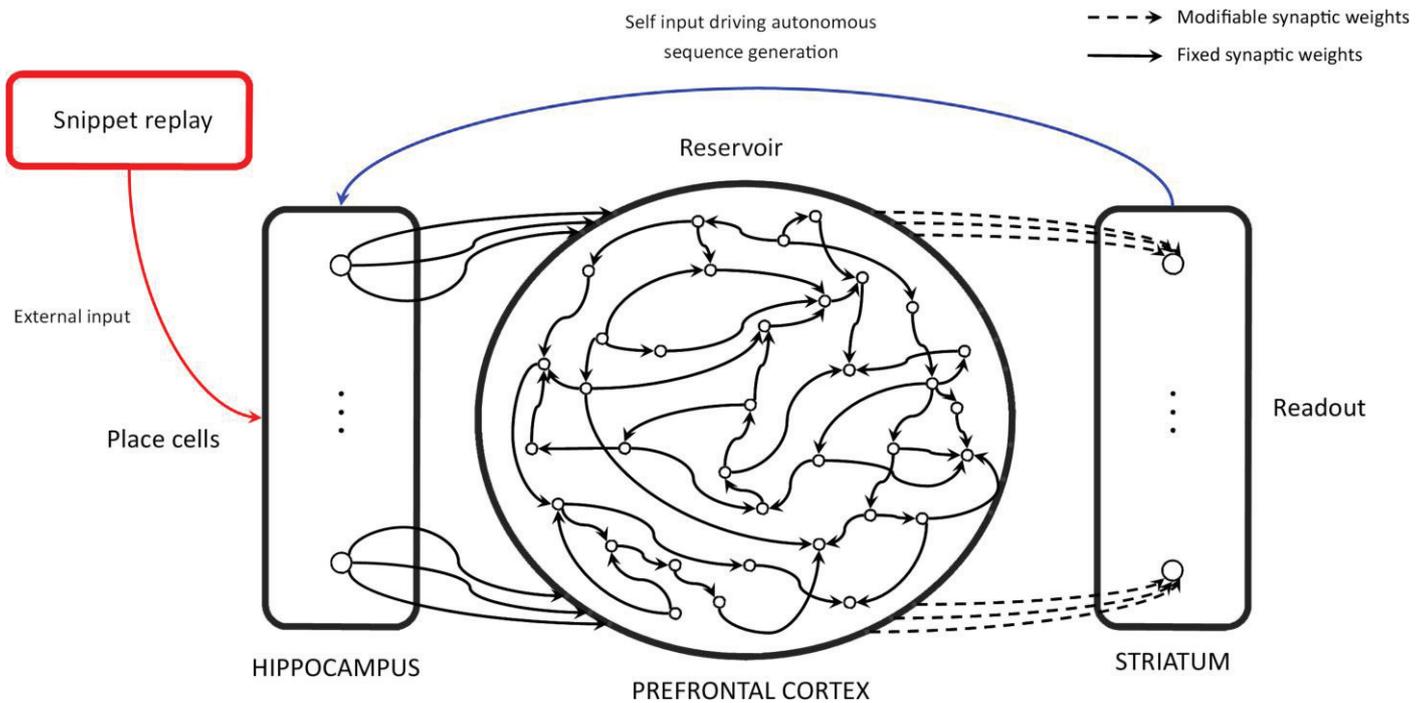


Figure 10: The early version of the reservoir model. The snippet replay algorithmic module displayed in red generates dynamically a dataset used for training the model. The blue arrow represents the feedback link used in autonomous generation mode.

2.1.1. C++/QT simulator

The idea behind this simulator was to provide an interactive and easy to simulator the user, allowing him to play with the model and its parameter in order to build an intuition on the model functioning. C++ language and the QT graphical user interface toolkit (GUI) were chosen for their robustness, performances and versatility, required for writing an interactive and responsive simulator. Boost library is also used for various utility classes.

The design is based on interdependent tasks executed in a particular order. From a modeling point of view, any simulation admits an equivalent graph representation where vertices are the tasks and the edges are the dependencies between tasks. A simulation is fully described by a scenario xml file where tasks and their dependencies are declared. An example of a simulation description can be found in Figure 11 :

```

<Process type="Weights" name="Feedforward" eta="@feedforward_gain" alpha="1">
  <Data name="pre" ref="Input"/>
  <Data name="post" ref="Reservoir"/>
  <WeightsInitializer type="Uniform" name="input_initializer" a="-1" b="1" seed="@seed"/>
  <!-- <WeightsInitializer type="Gaussian" name="input_initializer" mean="0" stddev="1" seed="@seed"/>-->
  <NeuromodulationInitializer type="Constant" constant="1" />
</Process>

<Process type="Weights" name="Recurrent" eta="@recurrent_gain" alpha="1">
  <Data name="pre" ref="Reservoir"/>
  <Data name="post" ref="Reservoir"/>
  <!-- <WeightsInitializer type="Uniform" name="recurrent_initializer" a="-1" b="1" seed="@seed" />-->
  <WeightsInitializer type="Gaussian" name="recurrent_initializer" mean="0" stddev="@recurrent_stddev" input="Reservoir" seed="@seed" />
  <NeuromodulationInitializer type="Constant" name="recurrent_neuromodulation" constant="1" />
</Process>

<Process type="Weights" name="Readout" >
  <Data name="pre" ref="Reservoir" sink="x"/>
  <Data name="post" ref="Output" sink="x"/>
  <NeuromodulationInitializer type="Constant" constant="1" />
  <WeightsInitializer name="readout_initializer" type="Uniform" a="@readout_a" b="@readout_b" seed="@seed" />
</Process>

<Process type="Weights" name="Loop" alpha="0" eta="1">
  <Data name="pre" ref="Output"/>
  <Data name="post" ref="Reservoir"/>
  <NeuromodulationInitializer type="Constant" constant="1" />
  <WeightsInitializer type="Copy">
    <Data name="src" ref="Feedforward"/>
  </WeightsInitializer>
</Process>

<Process type="Weights" name="Feedback" alpha="1" eta="1">
  <Data name="pre" ref="Output"/>
  <Data name="post" ref="Reservoir"/>
  <NeuromodulationInitializer type="Constant" constant="1" />
  <WeightsInitializer type="Uniform" a="-1" b="1" seed="@seed"/>
</Process>

<Process type="Rule" subtype="LMS" name="rule" epsilon="1" skip="1">
  <Data name="weights" ref="Readout" />
  <Data name="pre" ref="Reservoir" sink="x1"/>
  <Data name="post" ref="Output" sink="x1"/>
  <Data name="desired" ref="Stimulus"/>
</Process>

<Actor type="Trainer" name="train" threshold="@train_th" epoch="@train_epochs" stimulus="Stimulus" decrease="@decrease"/>
<Actor type="Tester" name="test" preamble="@preamble" epoch="@test_epochs" stimulus="Stimulus" >
  <Data name="measure" ref="error" sink="relative"/>
</Actor>
<Actor type="Evaluator" name="evaluate" rounds="@evaluation_rounds" >
  <Actor type="Ref" ref="train"/>
  <Actor type="Ref" ref="test"/>
</Actor>

```

Figure 11: An extract from a scenario XML file describing a simulation

A task is the base class of a process or a widget. A process processes incoming data from its input sinks and writes the result on its outputs. An example of a process is a neuron layer, a running average or a stimulus generator.

A widget represents incoming data in a particular format on the screen and is attached to the main panel as depicted in Figure 12. A very useful widget is the object explorer which allows the user to display and modify the parameters of the model while computing. Other widgets are classic XY plots, bar plots, histograms and rasters. All widgets are instantiated in containers whose layout is also configurable in the xml scenario file.

The simulator is available as a standalone version called LiveViewer. It is mainly used for fast prototyping of simulations by combining building blocks and observing the effect of a parameter change.

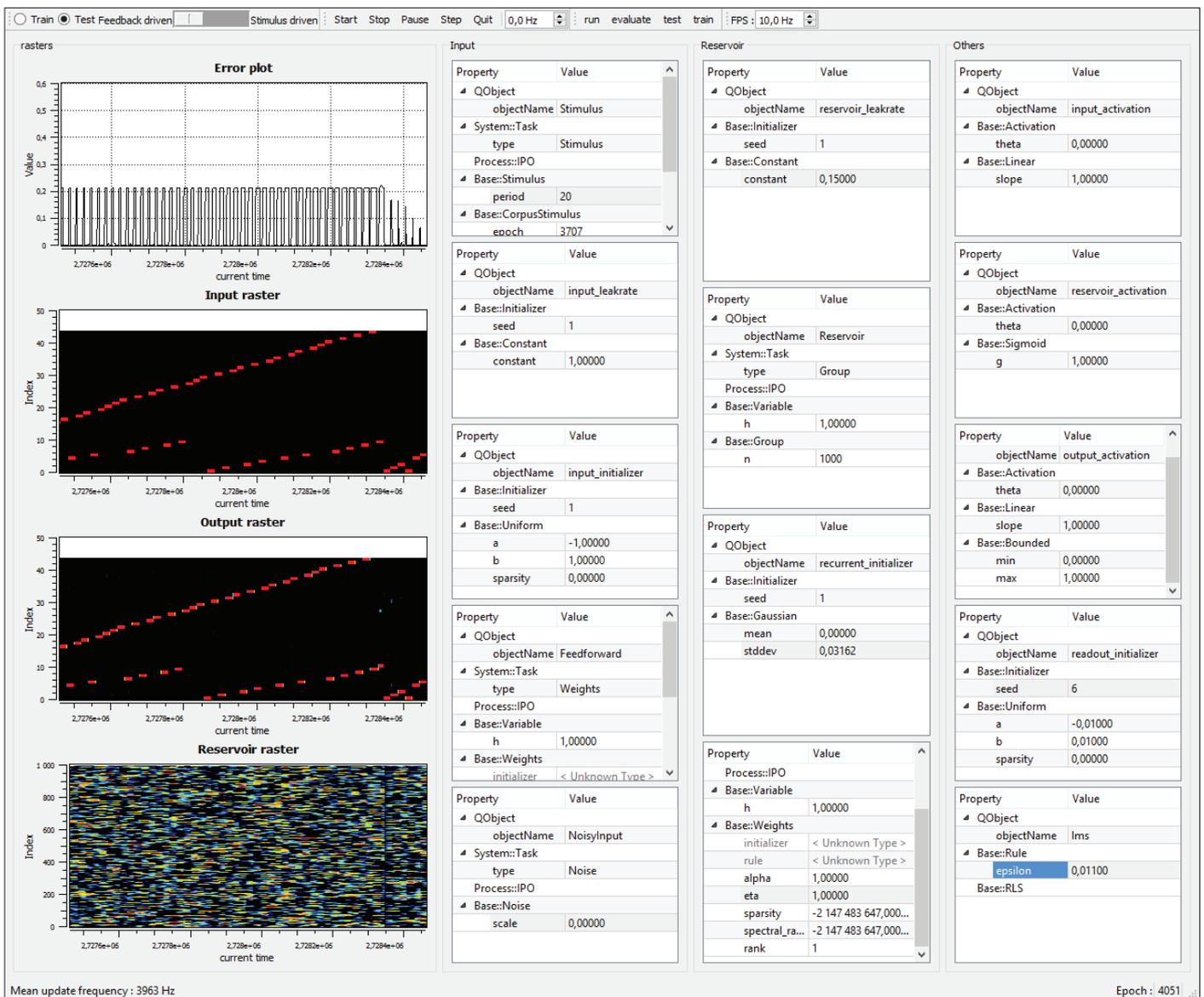


Figure 12: Main panel of the LiveViewer application and widgets controlling model parameters and displaying data plots in real time

The graphical user interface allows the user to limit the simulation speed in order to observe fast paced phenomenon and a step by step button allows one to run compute one simulation cycle each time the button is pressed. It is also possible to switch between training and test mode with a simple click at any time. Automation allows the user to specify program parts called Actors that will act on the interface like a regular user. For example, the learning rate of the model can be set to zero during the test stage and the constant controlling the amount of input signal from the stimulus and the feedback signal from the readout layer projected on the reservoir can be adjusted according to a ramp which is a function of the time. Figure 13 shows the possible uses of the simulator :

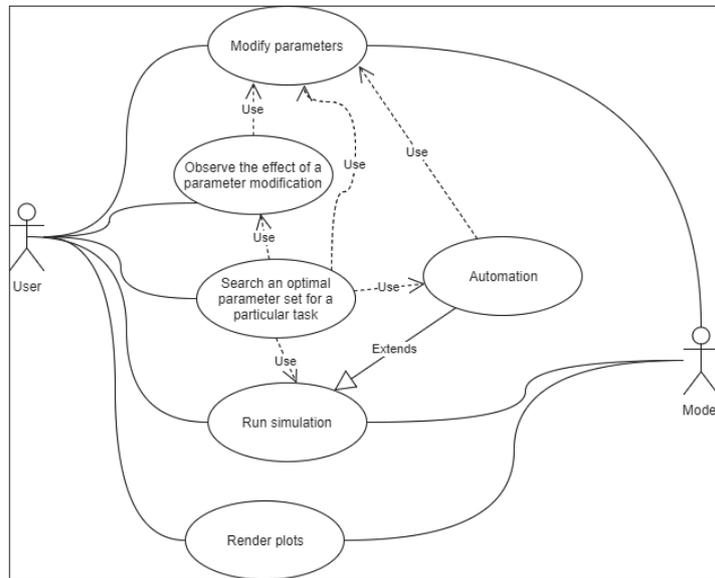


Figure 13: Use case diagram of the RRNN Simulator

A client/server version of the simulation is also available and is designed for cluster computing. The server part is made of two programs called Backend and Worker. The worker program role is to compute simulations and to send the results when terminated while the Backend program is responsible of the scheduling and life cycle of simulations. The Backend acts as a proxy between Clients and Workers. The client part is called Frontend and its role is mainly to read a scenario file, transmit its contents to the Backend and store or display results when available. Simulated annealing algorithm and grid search algorithms are used for parameter search and a graphical user interface (GUI) and a command line interface (CLI) are available. An algorithm and an interface is selected upon each invocation of the Frontend program.

Communications between Frontend and Backend use a TCP based text protocol and communications between Backend and Workers are implemented with MPI. CPU is used most of the time for computing tasks and some computing power demanding tasks are offloaded to a GPU.

Source code is available on GitHub at <https://github.com/NicolasCAZIN/RRNN>

2.1.2. Complex sequence learning

We propose to demonstrate that the reservoir computing model we use is able to learn and generate correctly long and complex stimulus sequences with a standard training method.

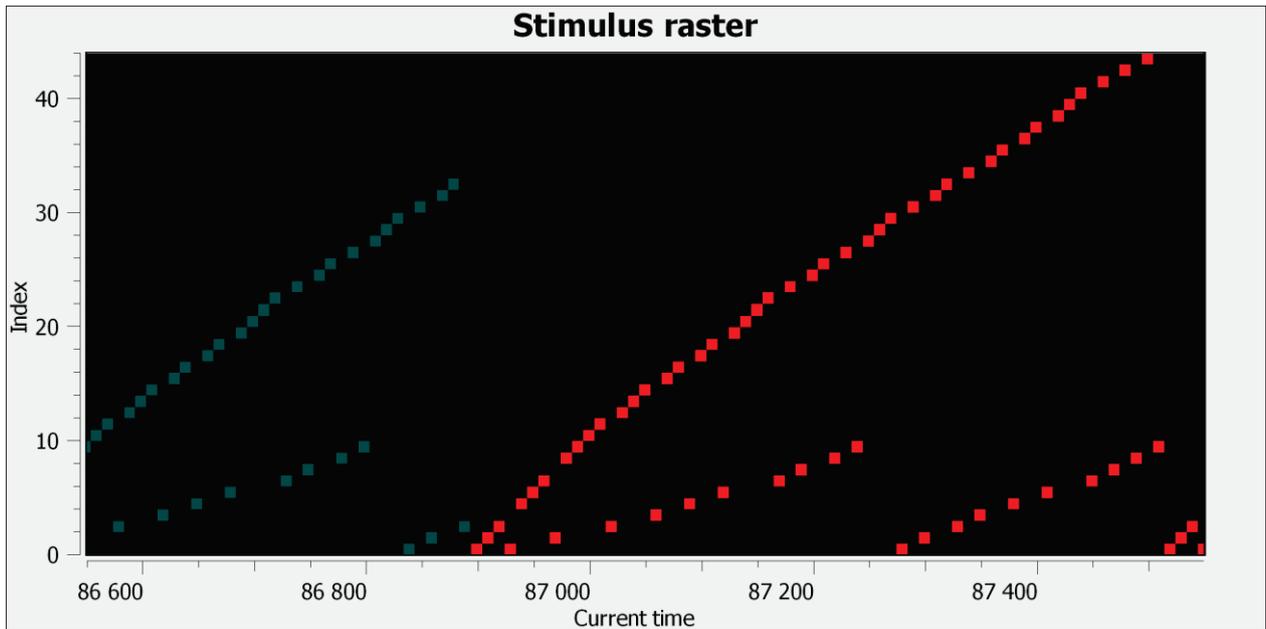


Figure 14: A complex and long stimuli sequence of 44 neurons over 540 timesteps

The stimulus at each timestep consists in 44 neurons activated with a serial temporal structure. Each neuron is activated during 10 timesteps and this elementary activation pattern is represented by a red square in Figure. The state of a given neuron is determined at least by the last 10 steps. The first 10 neurons have two different predecessor and successor neurons activated at different timesteps and the total length of the sequence is 540 steps. This sequence is called a complex sequence because it is characterized by several ambiguous transitions between neurons states. Figure 15 shows the transition probabilities between each neurons :

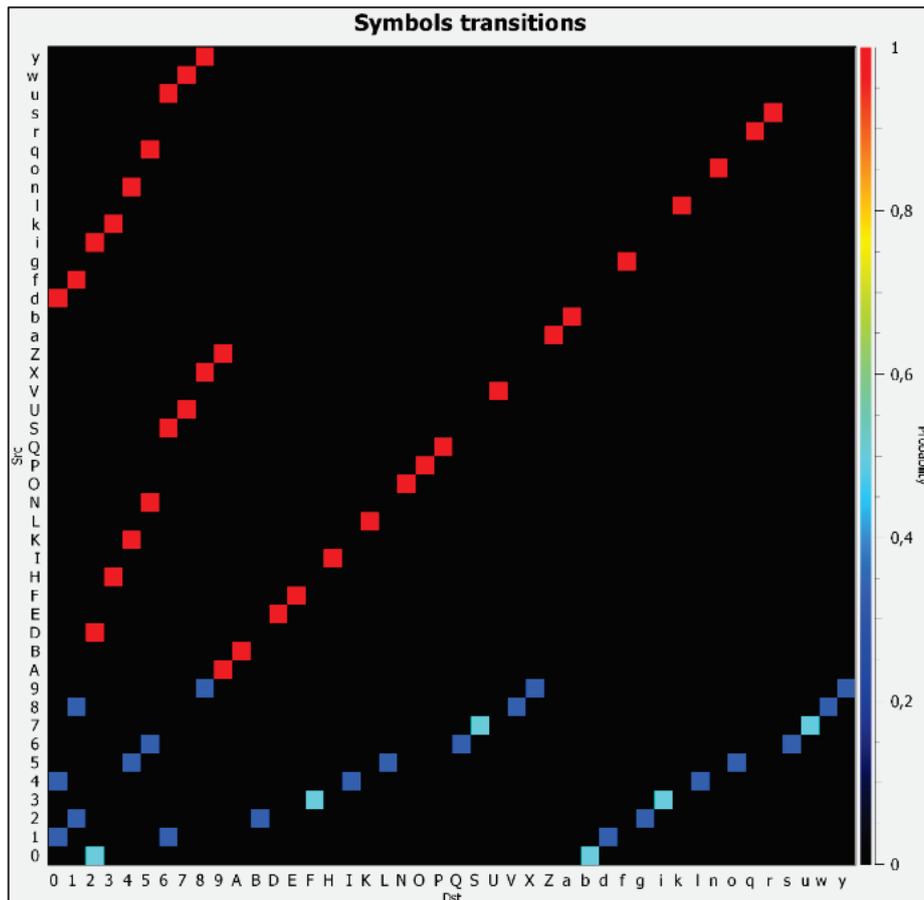


Figure 15: Representation of the transition matrix between symbols of the stimuli sequence

The activation pattern of the stimulus might be seen as a state transition function where both states and actions are activations values. The 10 first neurons exhibit ambiguous transition since their state are not uniquely

determined by their previous state. This learning problem cannot be solved without maintaining a form of context that will allow the association of a particular state of a neuron to a limited history of the states of other neurons. This is exactly the purpose of the cortico-cortical loop modelled by the recurrent connections in the reservoir computing framework. It is a form of fading memory that allow the state of the modelled prefrontal cortex to represent a context within a limited time line of sight. Intuitively, if the duration of the context required by the serial structure of the stimulus sequence fits in the short term memory implemented by recurrent connection, then the learning of a complex sequence is reduced to the learning of a simple sequence having the same number of steps but a higher number of meta-states resulting from the conjunction of successive stimulus states. This is illustrated in Figure 16:

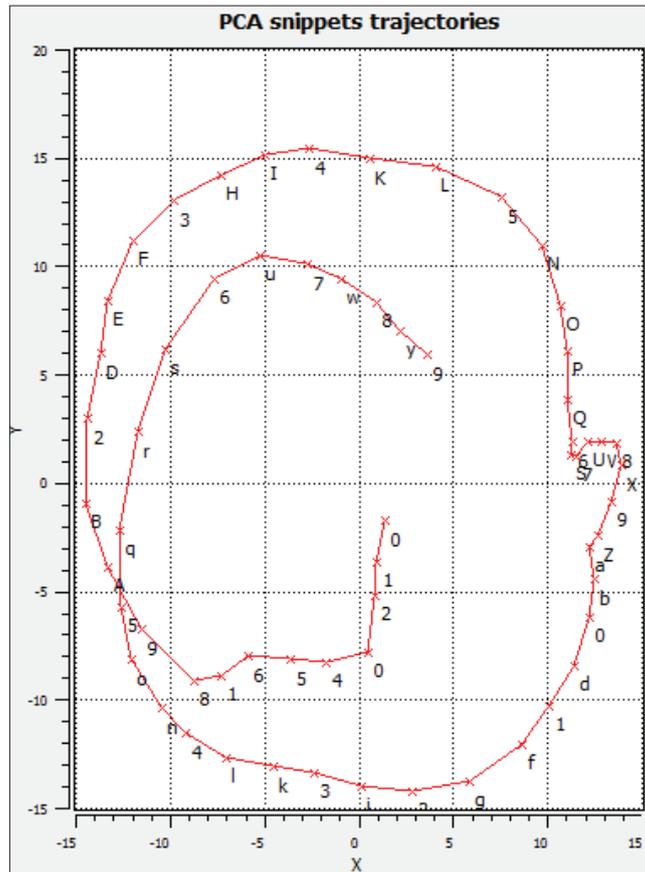


Figure 16: Reservoir's neurons states and transitions are represented as a 2D trajectory by displaying the first two principal components of the neurons activation raster.

This simpler problem is easily solved by exposing the reservoir model several times to the input activation sequence and associating incrementally the reservoir's state to the expected output activation sequence by modifying the synaptic weights of the readout layer according to the delta rule. When evaluated, the model is primed with the 30 first steps of the expected sequence and then the predicted neurons states are reinjected as the next input of the model until the end of the expected sequence is reached. The resulting activation pattern is similar to the expected sequence as depicted in Figure 17:

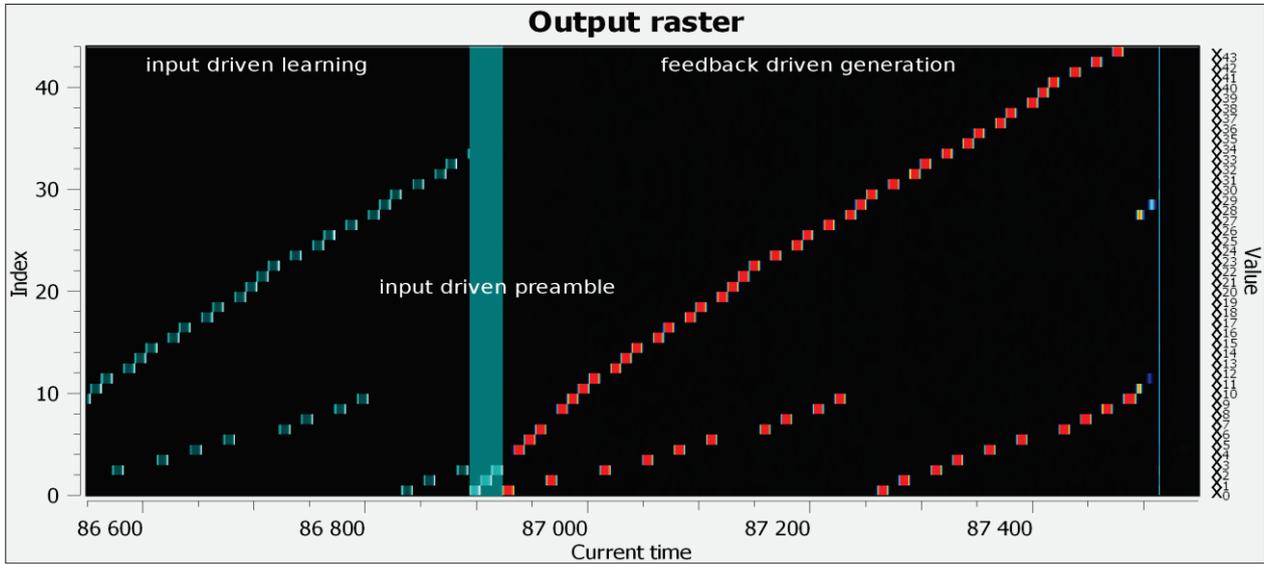


Figure 17: Raster of the readout neurons during training, priming and autonomous generation

The serial and temporal structure of the target sequence are reproduced but the sharp transitions from 0% activation to 100% activations cannot be reproduced by the model. This is explained by the time constant of the model which have been tuned for capturing the slow variations between each neuron (1 state change every 10 steps). Recurrent connections act like a running average and the side effect is to smooth the stimulus sequence along the time axis. This demonstrate the model's ability to learn and generate an arbitrarily long and complex sequences of neuron activation patterns.

2.1.3. Learning from random replay

As stated in, the main hypothesis in this work is that random replay of hippocampus place-cells subsequences during SPW-R plays a role in awake memory consolidation. We propose to model the snippet replay by exposing the model during learning to random subsequences of the target sequence. Each subsequence has a fixed length of 50 timesteps and relates the successive activation of 5 different neurons during 10 timesteps. 19 input neurons are modeled by a binary 0% or 100% mean firing rate and are labelled by letters A to S. The 1000 neurons of the reservoir are exposed to subsequences derived from the target sequence ABCDEFGHIJKLMNOPQRS selected at random according to a uniform distribution. Figure 18 relates subsequences and the target sequence:

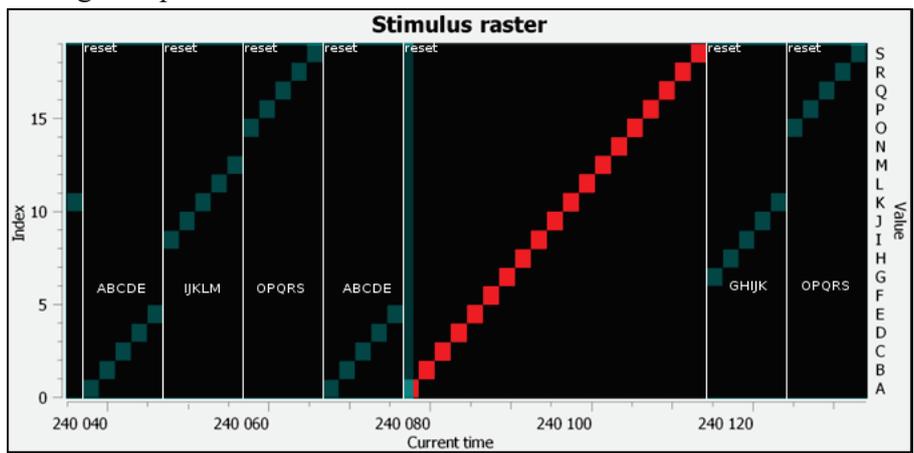


Figure 18: Raster of stimuli sequences. Random subsequences during the model training and the whole sequence during generation

We propose to compute the PCA trajectories of the reservoir's neurons when exposed to the subsequences. It appears that overlapping parts of subsequences (for example EFGHI and GHIJK) overlaps in the reservoir's

neurons state space represented along the two first principal components in figure. Even when replayed at random, each subsequence is represented or encoded in a similar manner by the reservoir. If the timescale supporting the stimuli is sufficiently slow compared to the timescale of the reservoir, then the mnemonic ability of the network will be exceeded before the end of a subsequence. The context will fade out quickly and two sequences having a different prefix will result in a similar trajectory after a sufficiently long delay. This dynamic property is illustrated in Figure 19 and results in the alignment of multiple sequences.

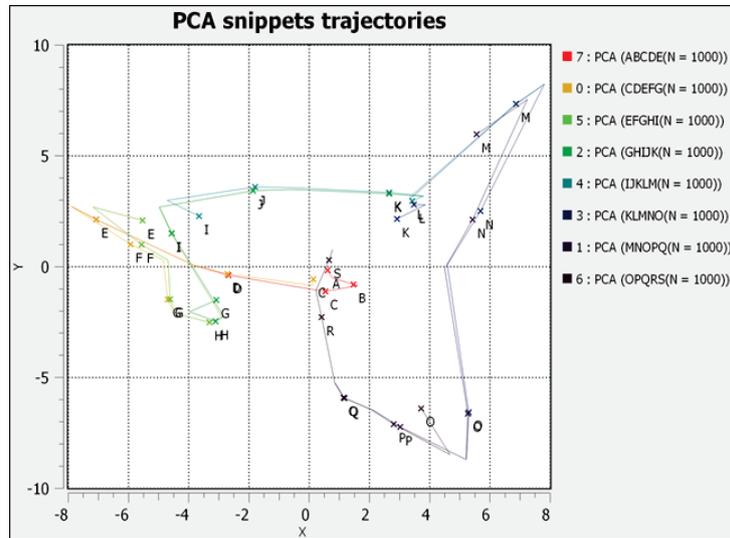


Figure 19: Reservoir neurons when exposed to a random replay of snippets. States are displayed as a 2D trajectories whose coordinates are the two first principal components as computed by the PCA algorithm

Overlapping parts of the trajectories are encoded in a similar area and this coherency in the reservoir's state coding allows the readout layer to associate successfully the reservoir's states to the prediction to the next stimulus state as demonstrated in Figure 20:

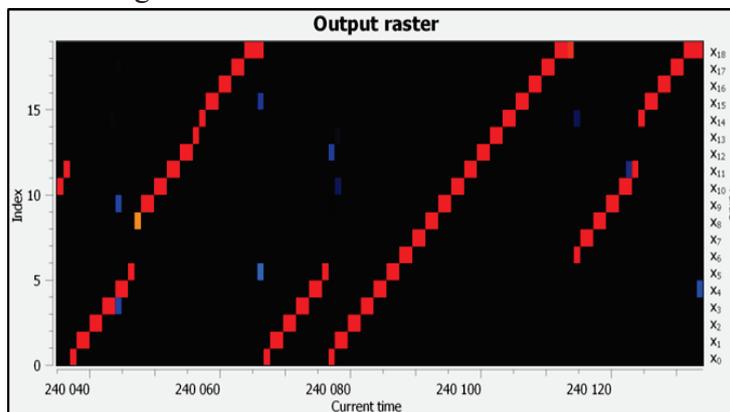


Figure 20: Raster of the readout neurons when exposed to random replay of snippets during training and primed by the whole sequence

The model is able to autonomously generate an ascending sequence, based only on a random replay of ascending subsequences. An ascending sequence is a simple sequence without any ambiguous transition. The recurrent connections are not required for solving this simple problem and it was only the first step for validating the sequence learning driven by a random replay within the reservoir computing framework.

2.1.4. Overlap

As demonstrated previously in Figure 19, the consolidation from random replay of snippets relies on the overlap of subsequences in the reservoir's neurons state space. We propose to characterize empirically the conditions on overlap that allow the snippet replay driven learning to occur. A replay episode is characterized by a duration T , the number of snippets N and their length L . We consider a fixed time budget of T time steps allocated to snippet replay. Several combination of subsequences of different length are possible and we show in Figure 21 the constrained relationship between the number of snippets of fixed length that could be replayed

during T timesteps and the resulting overlap between snippets (i.e. the number of samples shared by two given snippets)

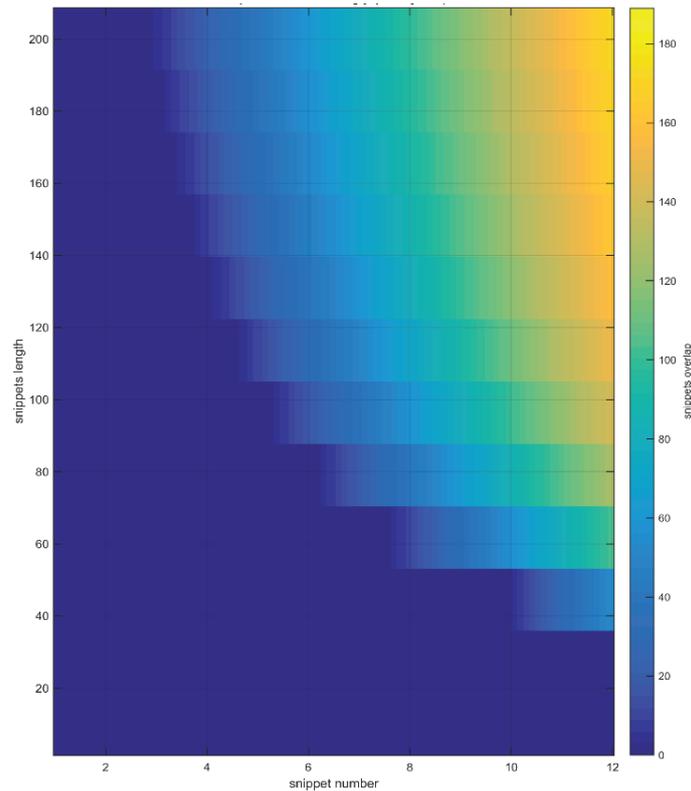


Figure 21: Heat map representing the relationship between snippet length (Y-axis), snippet number (X-axis) and snippet overlap (color)

We observe that only a restricted set of snippet length/number combinations allows a non-zero overlap between snippets. Without intrinsic overlap between contiguous snippets, learning the underlying complete sequence would be impossible because of important missing parts of the whole trajectory in the training set. We define a sequence of 20 neurons activated sequentially during 10 timesteps each and expose our model several times to different combination of snippet number during 100 cumulative learning episodes of 10 training epochs. Based on the constrained relationship between snippet length and snippet number, the snippet overlap is deduced and model performances in non-autonomous mode are displayed in Figure 22:

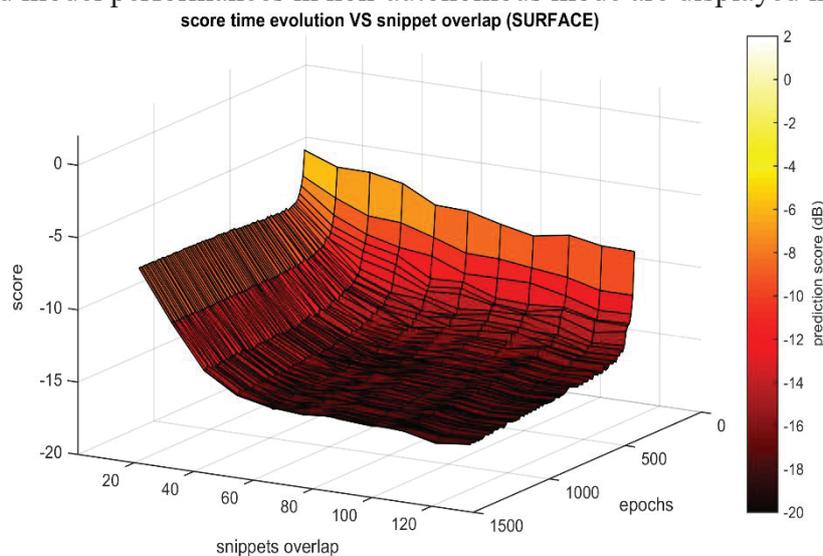


Figure 22: Model performances represented as a surface. Snippets overlap and training epochs have an influence on prediction error

It shows that a for a fixed set of parameters, a minimum exposure time to random replay and a minimum overlap between snippets are required in order to reach a low error level. The minimum is reached for a maximum exposure time and a maximum overlap. A particular case of snippet scheduling which is a unique

snippet whose duration equals the duration whole sequence. We are not interested in this case which is equivalent to the classical sequence learning paradigm.

2.1.5. Noisy conditions

The consolidation from random replay benefits from the robustness against noise of the neural networks. We demonstrate in Figure 23 that a snippet replay altered with a strong noise (-3dB in panel A and -8 dB in panel C) still allow the consolidation of the target sequence (respectively panels B and D).

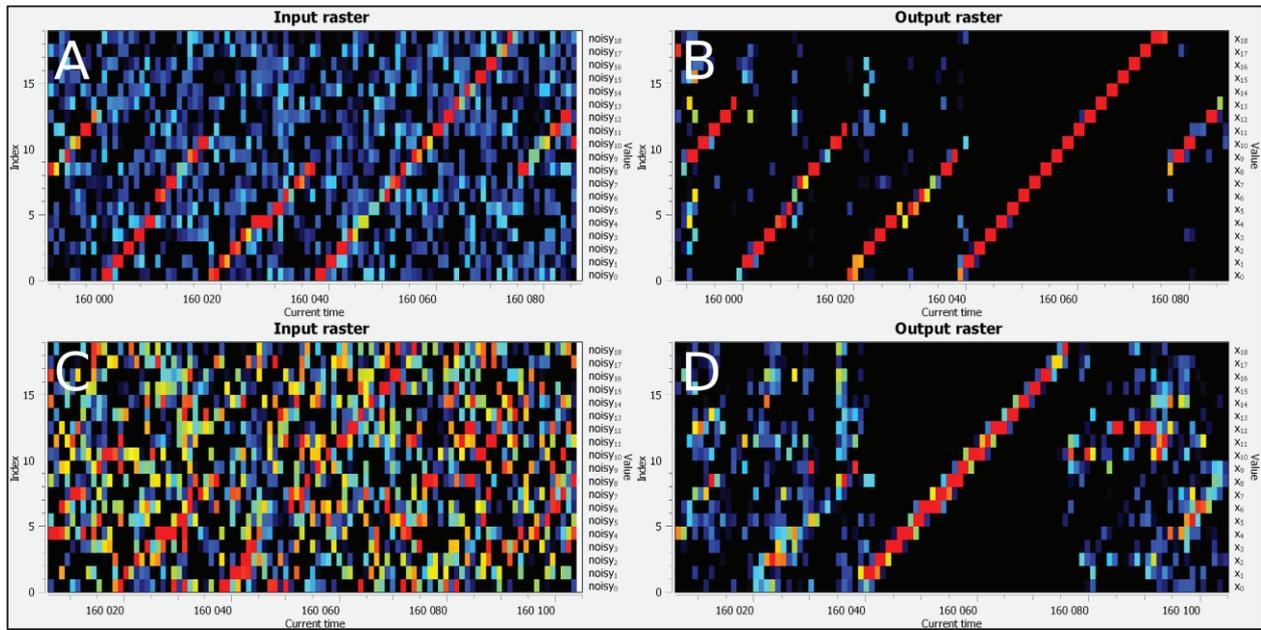


Figure 23: Input (Panels A and C) and output (Panels B and D) neurons rasters during model training and testing with an additive noise

2.1.6. Reliability over multiple time scales

Awake SPW-R replay occurs after a trial at a faster timescale than the stimuli experienced during the trial. We propose to model that by a neuromodulation mechanism. The stimulus sequence is learnt from random replay of subsequences with 10% noise added and when evaluating the model for generation, the model is primed with the beginning of the sequence. This is illustrated by Figure 24 panel A. The model is then put into autonomous mode by reinjecting the prediction of the readout layer as an input of the reservoir for the next timestep. The propagation delay between neurons is globally modified by changing the leak rate that reflects the resistive and capacitive properties of each neurons of the reservoir. In panel B we can observe the correct generation for a leak rate factor of $\left\{1, \frac{1}{2}, \frac{1}{4}\right\}$ Panel C includes factor $\frac{1}{8}$ and panel D $\frac{1}{16}$. This property has been evaluated up to factor equals to $\left(\frac{1}{2}\right)^{12}$ (i.e. 4096 times slower than the modelled random replay).

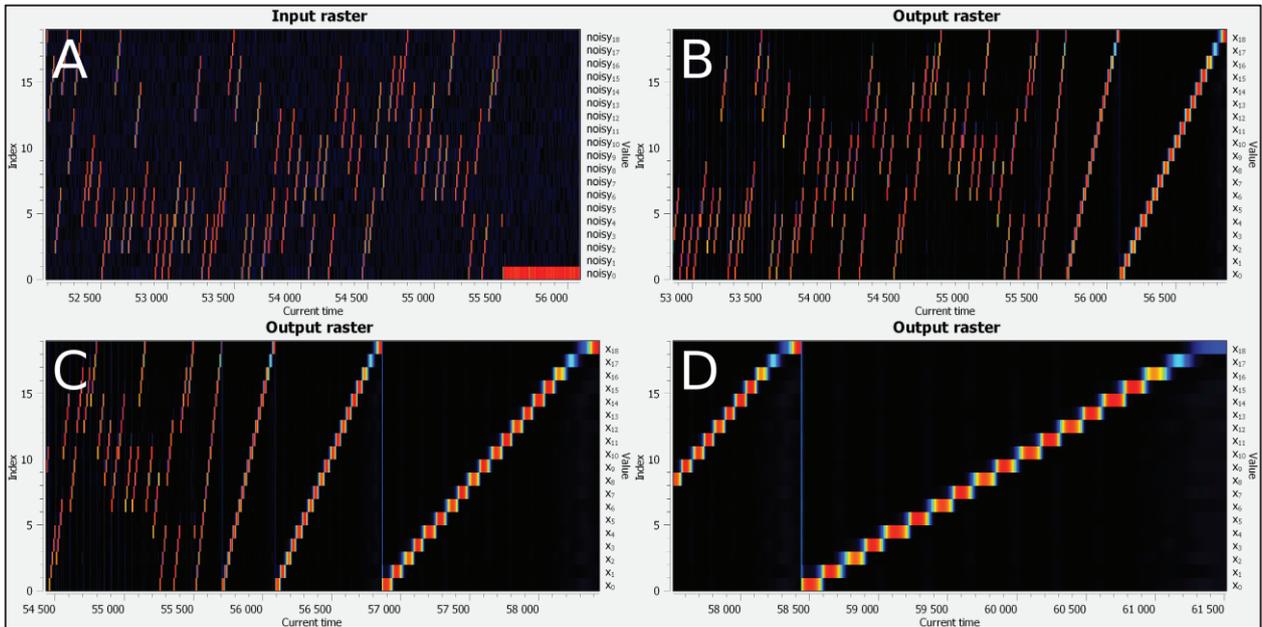


Figure 24: Demonstration of the model's sequence generation ability over multiple timescales while trained at a faster timescale

2.1.7. Sparse snippets replay

It had been observed that during SPW-R, hippocampus place-cells are replayed as a subsequence featuring the serial order encountered during the training and missing place-cells in a snippet could be observed. We propose to verify the robustness of our PFC-ST model by exposing it to a random replay of snippets featuring missing place-cells in each snippet. Figure 25 shows a raster of the readout neurons when exposed to a random replay of altered snippet and the generated ascending sequence:

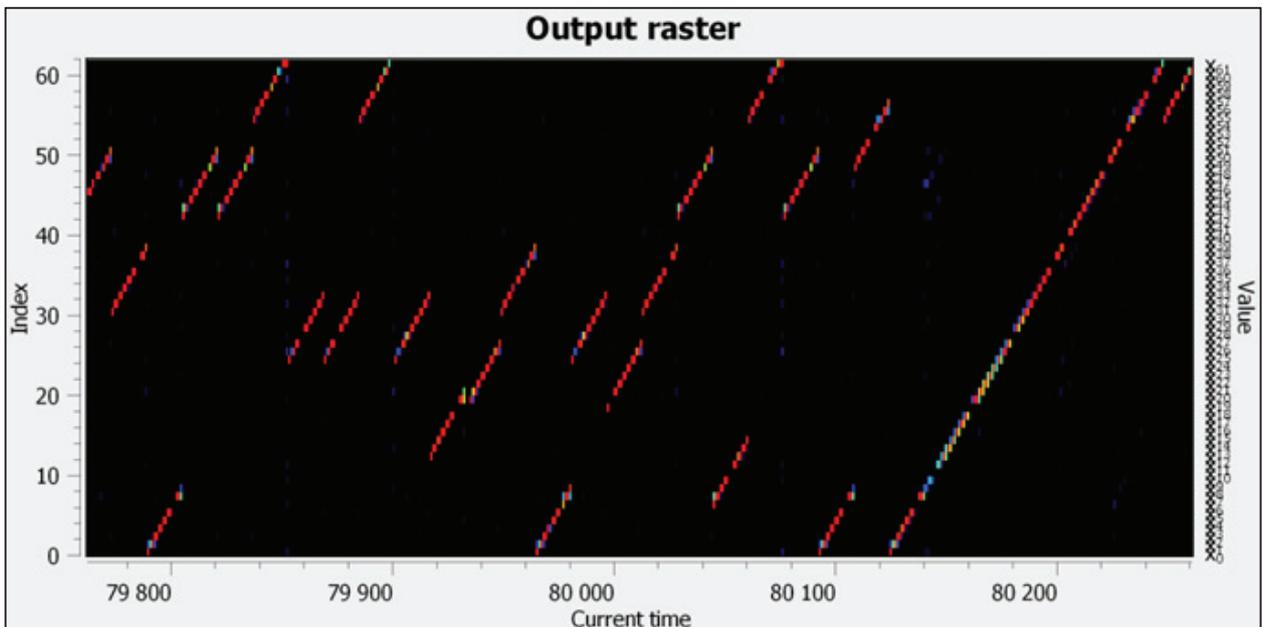


Figure 25: Raster of the readout neurons when trained with snippets sparse snippets and the resulting generated sequence

Despite the fact that the generated sequence features missing place-cells in some parts, the information conveyed through the recurrent connections implements a form of short term memory that allow the consolidation of the complete sequence and the generation to occur until the end of the sequence.

2.1.8. Consolidation of complex sequences

We have demonstrated that it is possible to learn a simple sequence from a random replay of snippets even in noisy conditions. A more challenging task for our model is to learn a complex sequence from a random replay of snippets in noisy conditions.

We use the same input sequence as in section 2.1.2 and configure a random replay of snippets with a 10% additive noise and one random missing place-cell activation during 10 timesteps for every snippet replayed. It is illustrated in Figure 26.

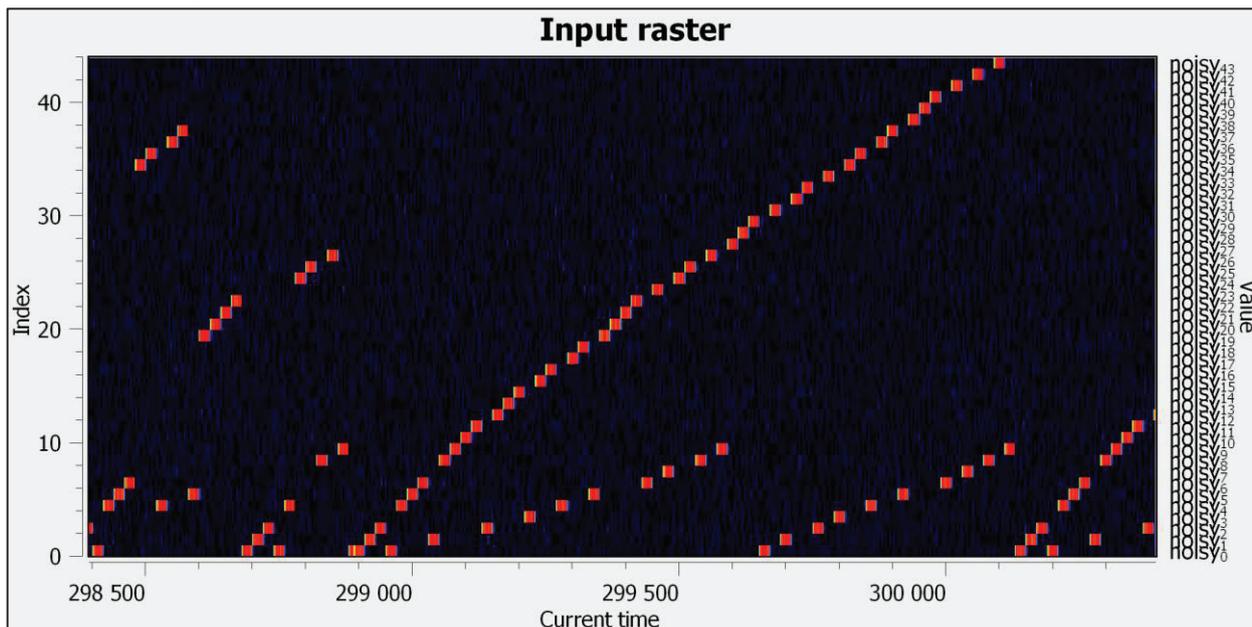


Figure 26: Raster of input neurons when exposed to the whole sequence and to a random replay of snippets of the whole sequence with an additive noise

The generated sequence is depicted in the output raster in Figure 27. The serial structure respected most of the time and we observe wrong predictions emphasized by white circles. The generation still occurs after the wrong predictions.

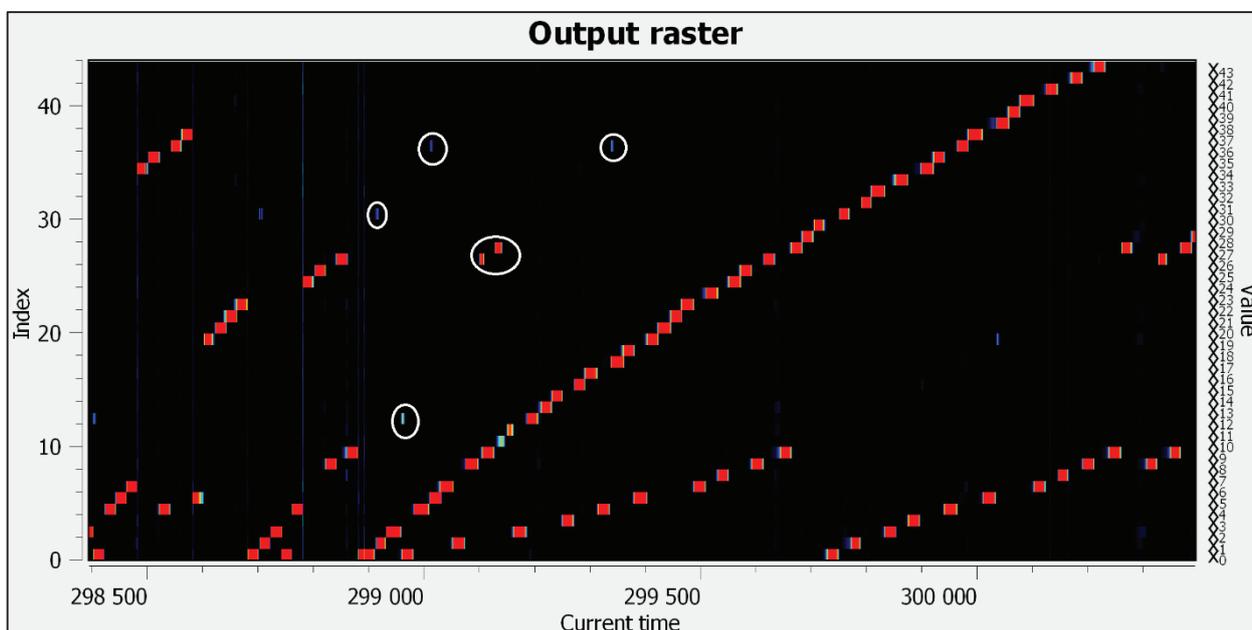


Figure 27: Raster of readout neurons generating the whole sequence and learning the whole sequence from a random replay of sparse, noisy and complex snippets. White circles emphasize ambiguous transitions present in the training set. It reflects the difficulty of this sequence learning task

The autonomously generated sequence displays a global time dilatation: the end of the generated sequence occurs at time step 300 230 instead of 300 020. This is explained by the fact that the leak rate acts as a temporal smoothing filter and consequently, a non-zero time is required for establishing a given neuron firing pattern. This delay is observed as a continuous increase and decrease of mean firing rate at the beginning and the end of a neuron activation pattern. The generated output at each timestep is never corrected and the delay is

cumulated during the generation process occurring within a closed loop between prediction and input. This points out the need of a metric that measures the similarity between sequences having arbitrary dimensions but not necessarily the same duration. We propose to use the Fréchet distance.

2.2. Towards a more realistic model

Previous investigations have demonstrated that our joint PFC-HC model is able to consolidate at least a single complex trajectory based only on random replay of snippets. The stimuli used so far were emulating hippocampus place-cell coding by affecting one binary neuron sparse orthogonal representation (learning of a competitive network) for coding one particular location. No combination of neurons in a firing pattern were allowed and all neurons fired strictly orthogonal patterns (i.e. null dot product). The purpose of the model being to explain and predict data issued from TSP related experiments with a rat, we need to extend our model along several axes:

2.2.1. Use trajectories performed by a rat

The synthetic trajectories used so far for demonstrating dynamic properties of the joint HC-PFC model related to TSP problem are designed by hand and aims at capturing particular features of the rat behavior we are trying to explain. They do not represent the trajectories performed by a rat trying to solve the TSP problem. We then use behavioral data about rat solving the TSP problem provided by Fellous & AI, CENL Lab, University of Arizona, Tucson. Experiments take place in a circular arena having a radius of 1.5m. 21 feeders are scattered according to a spiral shape. The feeder's location and label is always the same. Some feeders are baited with food whilst others are not. We call a configuration the particular state of baited/non-baited feeders. 100 different configurations of baited feeders are explored by different rats during 10 trials. Before each trial the rat is positioned on the same starting point, near the first baited feeder. Figure 28 panel A shows the trajectory performed by the rat during trial 4 for configuration 84

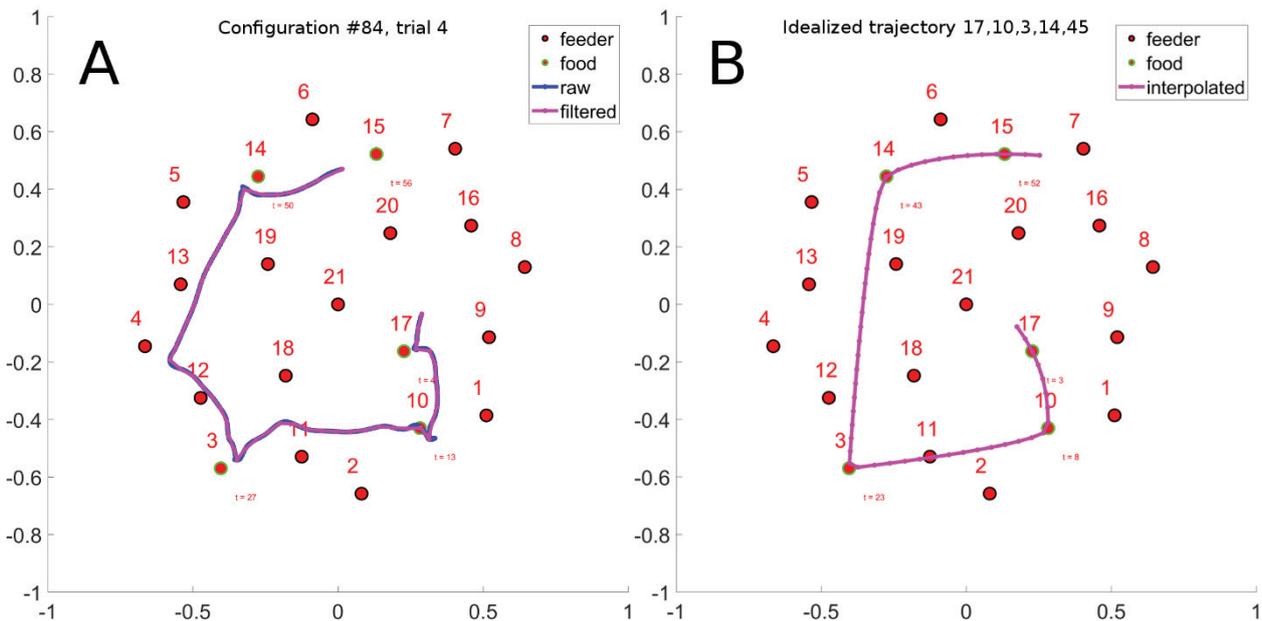


Figure 28: Trajectory performed by a rat (Panel A) and an idealized version (Panel B)

The (x, y) points of the trajectory are provided by a camera based tracker. Measures might contain noisy or missing samples and that is why we interpolate data with splines. We consider a trajectory as a graph. The vertices are the feeders and correspond to cities in the TSP problem and the edges are the paths that link the feeders. In the acquired dataset, the rat does not always find the optimal trajectory between baited feeders and 64 configurations contain at least a trial where an efficient path is found. An efficient path is found when a given path is a circular shift of the optimal path or the optimal path in reverse order. The subset of 64 configurations containing at least one efficient path each is called the converged subset. Two configurations

of the converged subset contain an efficient path but all edges of the optimal trajectory had not been encountered. We restrict the dataset to the converged configurations where every feeder pair of the optimal trajectory is encountered at least once. This subset is called the complete subset and it accounts for 22% of the acquired data. We will use this subset in our experiments. If baited feeders are visited in the correct order, it does not mean that an optimum has been reached. Subj-trajectories linking baited feeders might not describe the shortest path between these two points. We created an idealized trajectory that links optimally the baited feeders and an example is shown in Figure 28 panel B. This is required for establishing a performance criterion. Trajectories are down sampled to a spatial resolution of $20 \text{ dot} \cdot \text{m}^{-1}$. This resolution is sufficient for representing the motion of the rat, using a very high resolution would capture the noise of the camera based tracker and as demonstrated in section 2.3, there is a direct relationship between the spatial resolution of a stimuli sequence and the leak rate required by the reservoir model for capturing significant variations of the sequence.

2.2.2. Use of a place-cell model

Place-cells are a type of pyramidal neurons in the hippocampus that fire preferably when the rat is located in a particular place called place field. We propose to use an isotropic 2D Gaussian place-cell model defined by:

$$f_k(s) = e^{-\frac{\|s - c_k\|^2}{w_k}} \quad (1)$$

Where:

- k is the number of the place-cell
- $f_k(s)$ is the mean firing rate of the k^{th} place-cell
- c_k is the (x, y) coordinate of the k^{th} place-cell
- $w_k = \frac{r_k^2}{-\log(\theta)}$ is a constant that will constrain the highest activations of the place-cell to be mostly contained in a circle of radius r_k , centered in c_k
- r_k is the radius of the k^{th} place-cell
- θ is the radius threshold which controls the spatial selectivity of the place-cell?

Trajectories are defined within a $2 \times 2 \text{m}$ square space spanned by a regular grid of 16×16 2D Gaussian isotropic place-cells. Thus from a simulation point of view, a given trajectory result in a raster of 256 place-cell neurons as illustrated in Figure 29 panel A and B and is defined by the conjunction of the place-cells activations concatenated for each time step. One can observe in panel C that there exist parts in the trajectory where the trajectory first order gradient with respect to time is null. It means that the trajectory features static parts where the rat is not moving. Since the reservoir model captures the serial and temporal structure of a trajectory through the recurrent connections, a long static part might be partially captured by the PFC model. Indeed, a given sustained activation pattern as observed in panel B will allow the model to eventually exceed its mnesic ability and to produce a similar context after this point. The consequence is that the readout layer will associate several times this fixed context to the same output, resulting in the overrepresentation of a particular point of the sequence. In autonomous generation mode, this will lead to a fixed point. A practical workaround would be to modify the learning rule in order to allow a hebbian learning process to occur only when novelty is detected with the equivalent of a first order derivative filter. In fact, the formulation of the learning rule is similar to the definition of the covariance matrix between reservoir neurons and readout neurons. The

estimation of the mean activation value of a neuron over time depends on a particular timescale and this would add several hyper parameters to the learning rule. Another simple manner of allowing the learning process to occur only when novelty is detected is to algorithmically remove static parts of the stimulus where the modulus of the derivative with respect to time (illustrated in panel C) is non zero/

We hypothesize that this novelty detection mechanism is a part of the snippet formation process and we propose to model a snippet as a contiguous subsequence of the dynamic parts of a place-cell activation sequence. The regular sampling is the equivalent of a constant velocity. Panel D illustrates the condensed and resampled trajectory and panel E represents the place-cell activation sequence. It contains only dynamic parts of the trajectory and it will be used by the snippet generation procedure.

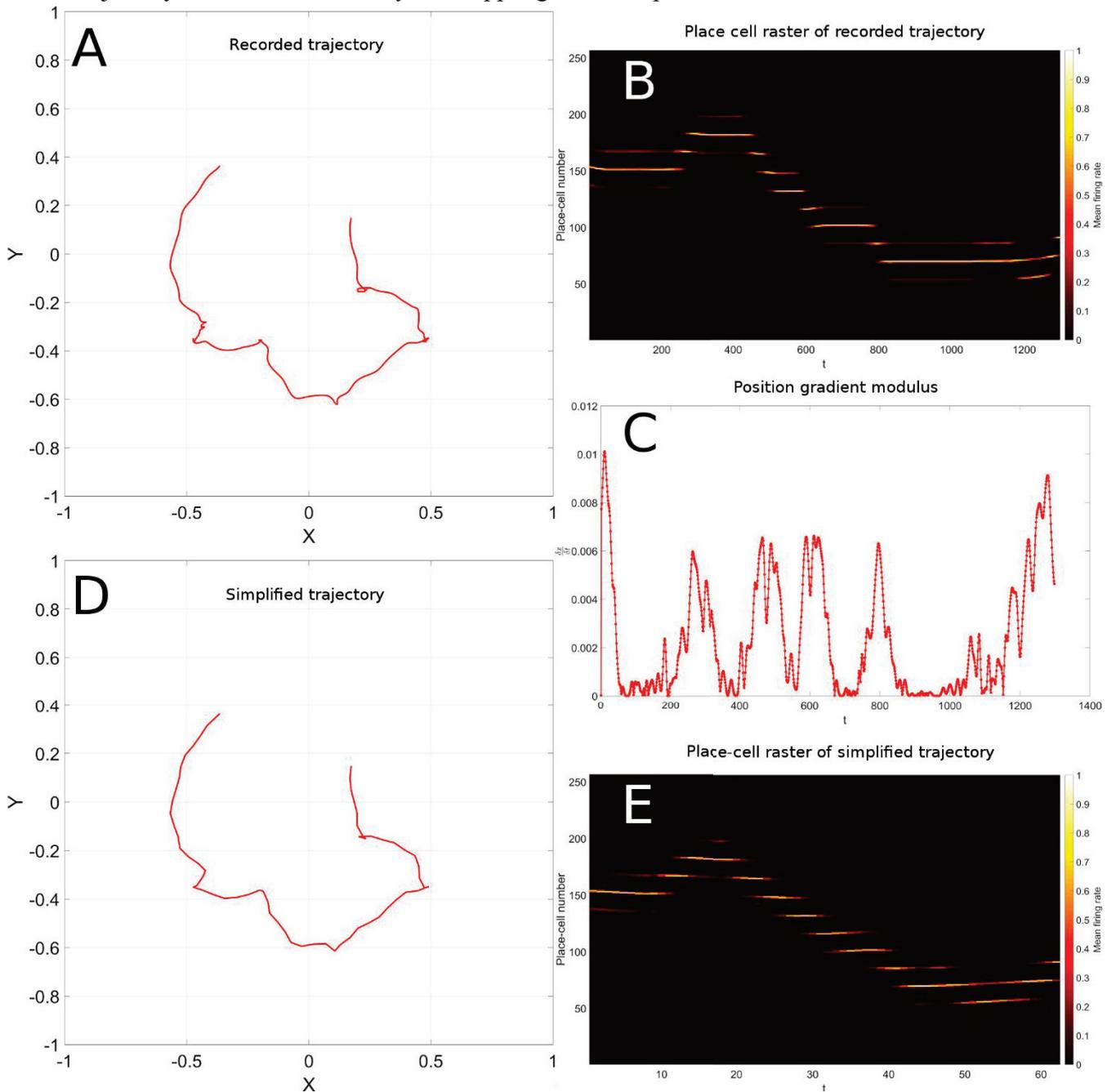


Figure 29: Trajectory performed by a rat as captured by the visual tracker. Panel B represents the corresponding place-cell activation pattern raster, panel C shows the gradient modulus of the position, panel D represents the condensed and resampled trajectory and panel E shows The corresponding simplified place-cell activation sequence.

For specific demonstration purposes, we propose to use place-cell activation sequences based on the place-cell model described above and spline interpolated trajectories satisfying a constraint on the order where the feeders are visited. Figure 30 Panel B shows a synthetic trajectory where feeders ABCED are visited. Gray

circles represent the place-cells centers and their radius. Panel C shows the corresponding place-cell activation pattern and the same snippet represented in the 2D trajectory and its place-cell extent.

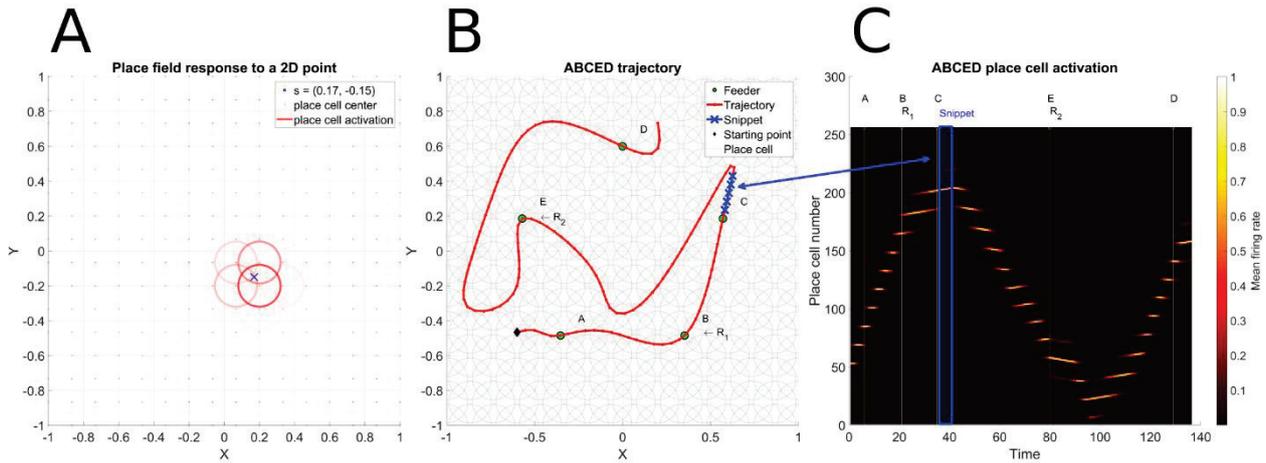


Figure 30: Place field model applied to a trajectory and representation of a snippet

2.2.3. Generate a set of snippets from multiple sequences based on reward

In the behavioral experiment described in (De Jong et al. 2011) we can observe the rat incrementally solving the TSP problem by experiencing new paths at each trial. In Figure 31 panel A, B, C, D and E, the rat describes a new trajectory at each trial and several sub-trajectories are present in different trials. For instance, the path linking feeder 20 to 12 is observed in panels A, B, C and E.

We hypothesize that the random replay of snippets that allow the model to generate an efficient sequence as depicted in panel F is based on sequences from remote trials and modulated by a form of reward. This is compliant to the behaviors and measurements described in (Gupta et al. 2010).

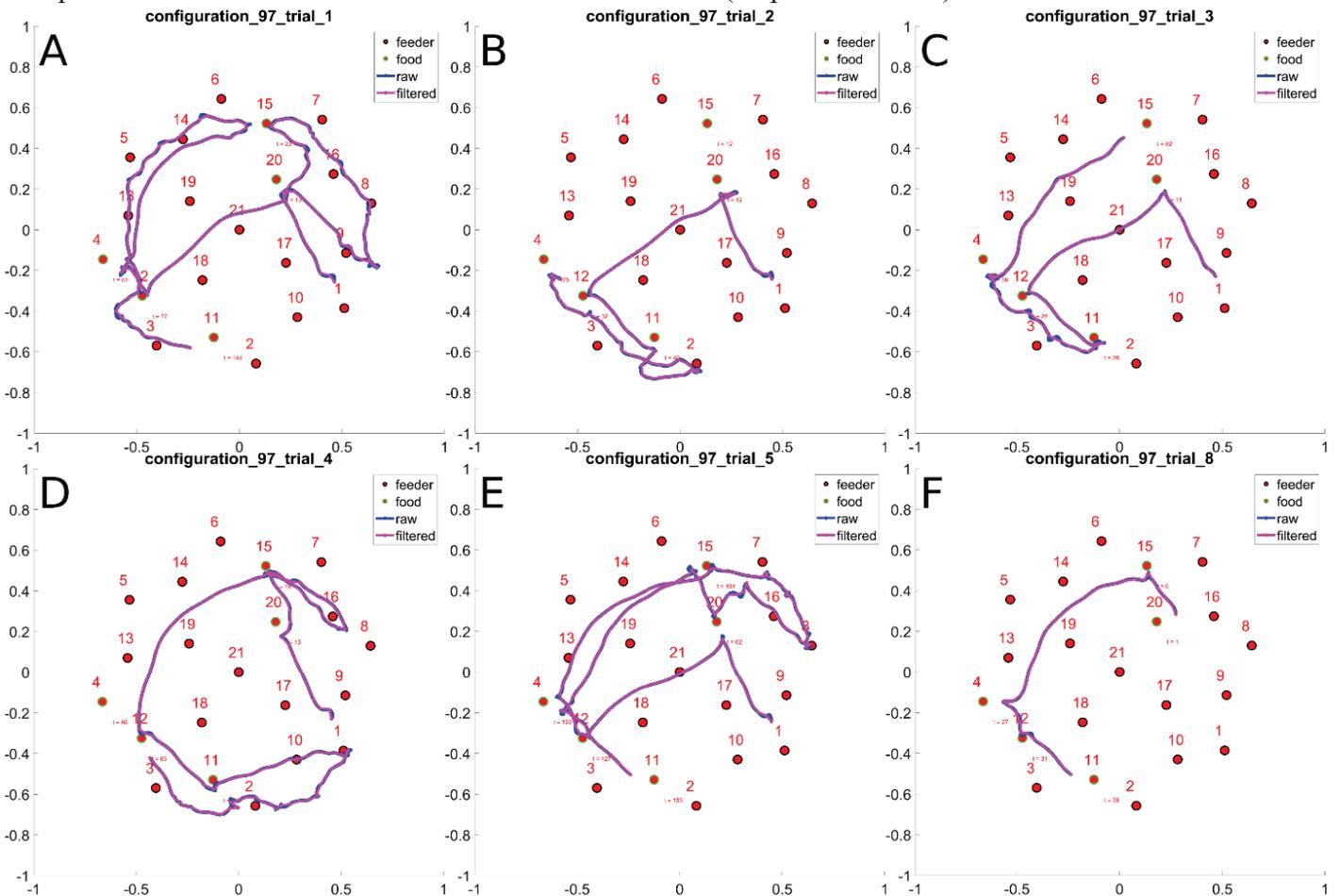


Figure 31: Trajectories performed by the rat over successive trials

The aforementioned modulation by a form of reward is illustrated in Figure 32. The BACDE sequence illustrated in Figure 5 panel D contains long and short paths linking the different feeders. This is related in Figure 32 by a different number of time indexes.

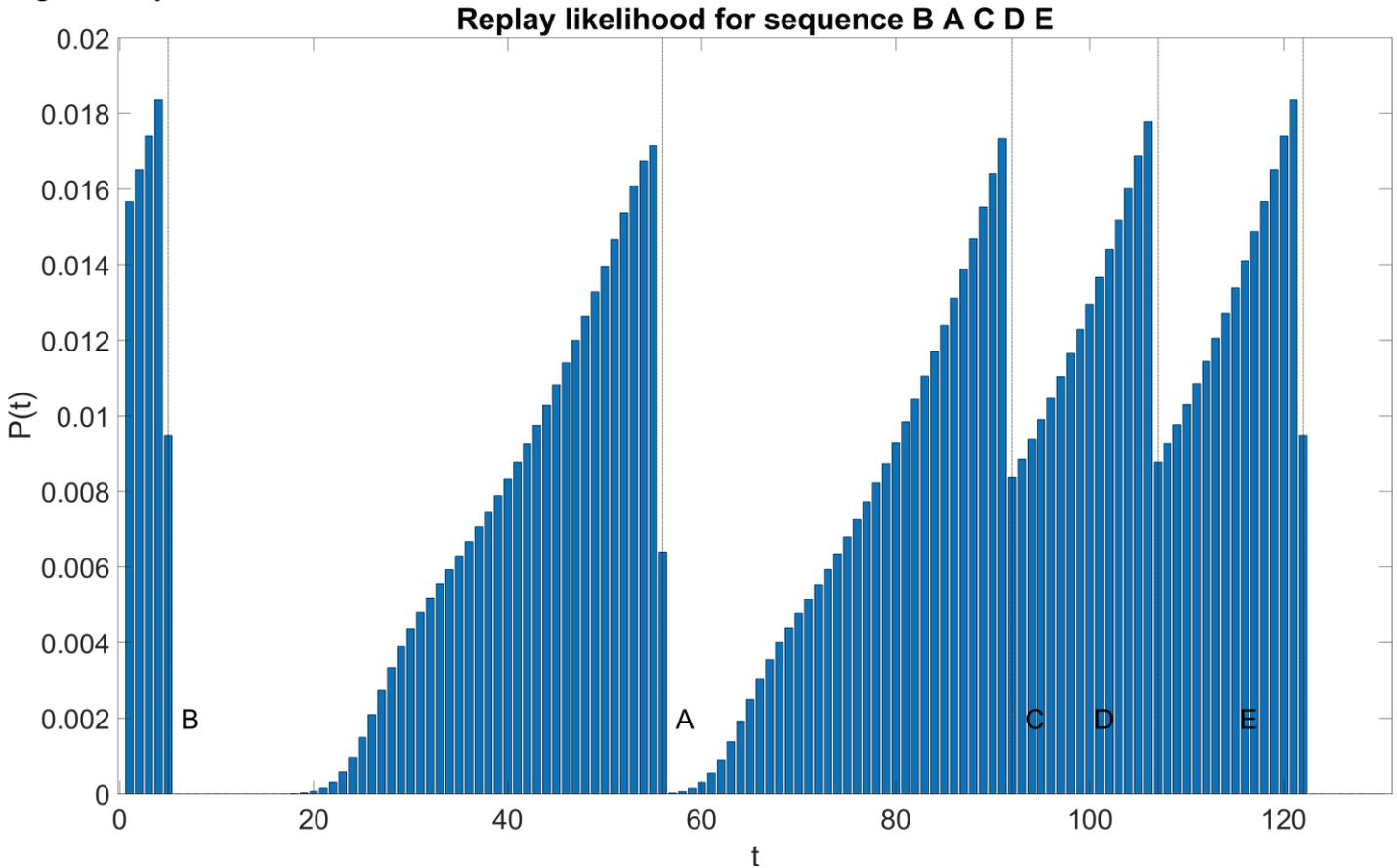


Figure 32: Snippet replay likelihood used in the snippet generation procedure.

The snippet replay likelihood reflects a prediction of the reward and is learnt by the replay model illustrated in Figure 6. Update equations and details about the replay model can be found in 2.4. It becomes possible to emphasize different parts of a trajectory related to rewards as depicted in Figure 33. If the reservoir model is able to concatenate the emphasized parts of different parts of a trajectory related to a reward and replayed at random, then it is possible to generate an efficient trajectory based on salient elements of recently experienced trajectory. This is the main result of the paper presented in section 2.4. From a more abstract point of view, the replay model proposes different associations between reward sites (A, B, C, D and E) and the consolidation model reconciles the overlapping representations, allowing to build a transitive relationship following a path between reward sites. This correspond to the point of (Preston and Eichenbaum 2013). Their understanding of the interplay between Hippocampus and Prefrontal cortex and their roles in the memory consolidation process entails the notion of schema, introduced to cognitive psychology by (Piaget 1967) and (Bartett 1932). The replay model proposes a solution to the credit assignment problem by propagating the reward information across time according to a $TD(\lambda)$ algorithm and the consolidation model allows to build a heuristic that is able to find efficient global paths in a graph by combining efficient sub-paths based on a reward expectation. This heuristic is in fact a neural implementation of a reinforcement learning (Sutton and Barto 1998) and it is a very simple and new manner to solve the credit assignment problem in recurrent neural networks. We will see that modulating snippet generation probability based on this reward value propagation will allow the implementation of a form of reinforcement learning that addresses the optimization requirement for solving the TSP problem.

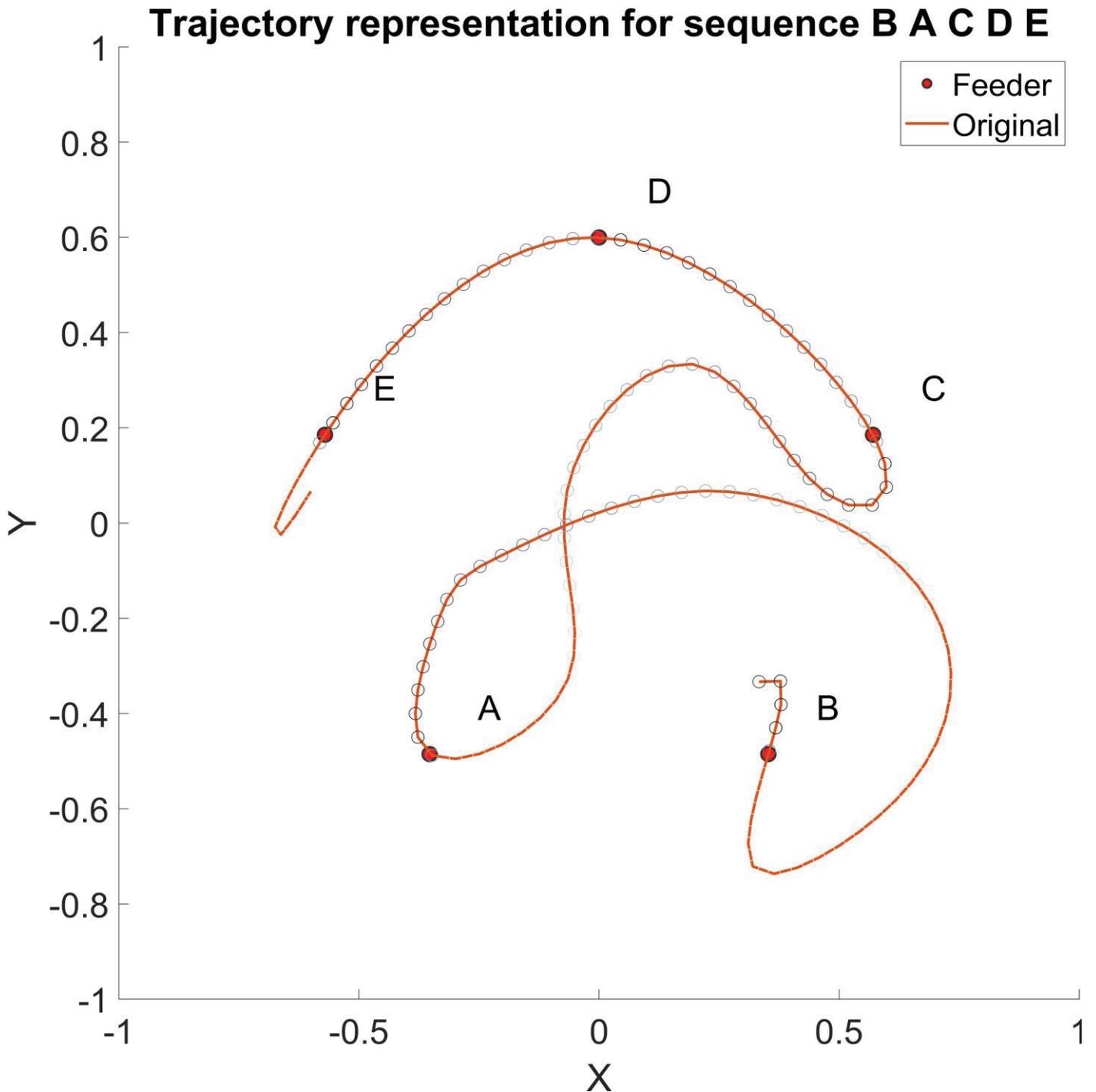


Figure 33: BACDE trajectory and the frequency of different parts being represented through random replay

2.2.4. Defining a sensory-motor loop

We propose to make a first attempt by selecting in the dataset³ of behavioral experiments configuration 40 where an efficient trajectory (feeders 14,12,18,17,1) is discovered during trial 5 and sub-trajectories of the efficient trajectory are visited during previous trials 2,3 and 4. Figure depicts these trajectories. During trial 2, feeders 14,18,12,14,7,1 are visited. Feeder 14 is consumed the first time it is visited and the second time, it is perceived as a non-rewarding feeder. The trajectory linking baited feeders 12 to 7 is particularly long and optimal sub-trajectory linking feeder 12 to 18 is performed in trial 2 in reverse order with a sub-trajectory linking feeder 18 to 12. This example illustrates the need to have a model that can extract sub-trajectories linking two reward sites with no particular direction and learn preferentially short sub-trajectories between rewards. During trial 3, feeders 14,18,12,17,1 are visited and it corresponds almost to the optimal trajectory,

³ Gracefully provided by Fellous & al, CENL Lab, Tucson, AZ, U.S.A

excepted that feeder 18 is visited before 12, instead of feeder 12 before feeder 18. During trial 4, feeders 17,1,14,12,18 which is a circular permutation of the optimal sequence. Sub-trajectories 17,1 and 14,12,18 belongs to the optimal sequence and this example suggests that a model able to consolidate randomly replayed parts of the optimal trajectories would be able to consolidate them if they are more salient as non-optimal sub-trajectories.

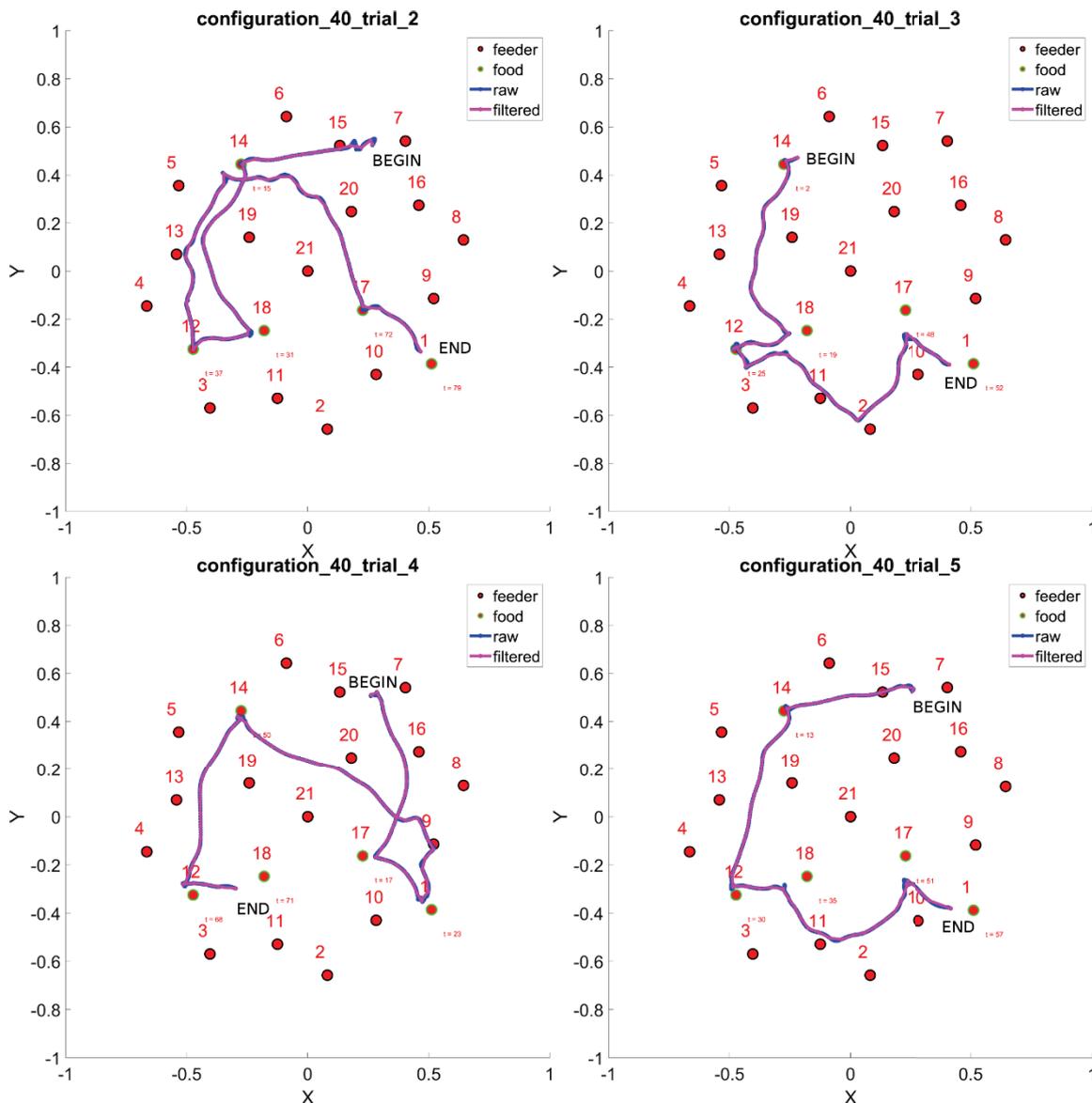


Figure 34 : Successive trajectories of the rat in configuration 40

The reservoir model is trained and evaluated several time with a random replay of place-cell activation subsequences based on different possible combinations of trial 2, 3 and 4. Each model is trained by exposing it to a random replay of snippets of 10 simulation cycles long for a duration of 1000 simulation cycles. The model is then evaluated by measuring the mean square prediction error on the readout layer with a Euclidian norm. This evaluation procedure is repeated 100 times and the mean square prediction error series are gathered for each trial combination in autonomous and non-autonomous generation mode. In autonomous mode, the predicted stimulus is reinjected as the next input, while in non-autonomous mode, the next input is forced to be the next stimulus as observed in the training set, regardless of the predicted stimulus. It allows one to measure an error along the whole trajectory. Figure 35 represents the prediction errors in non-autonomous mode for each trial combination.

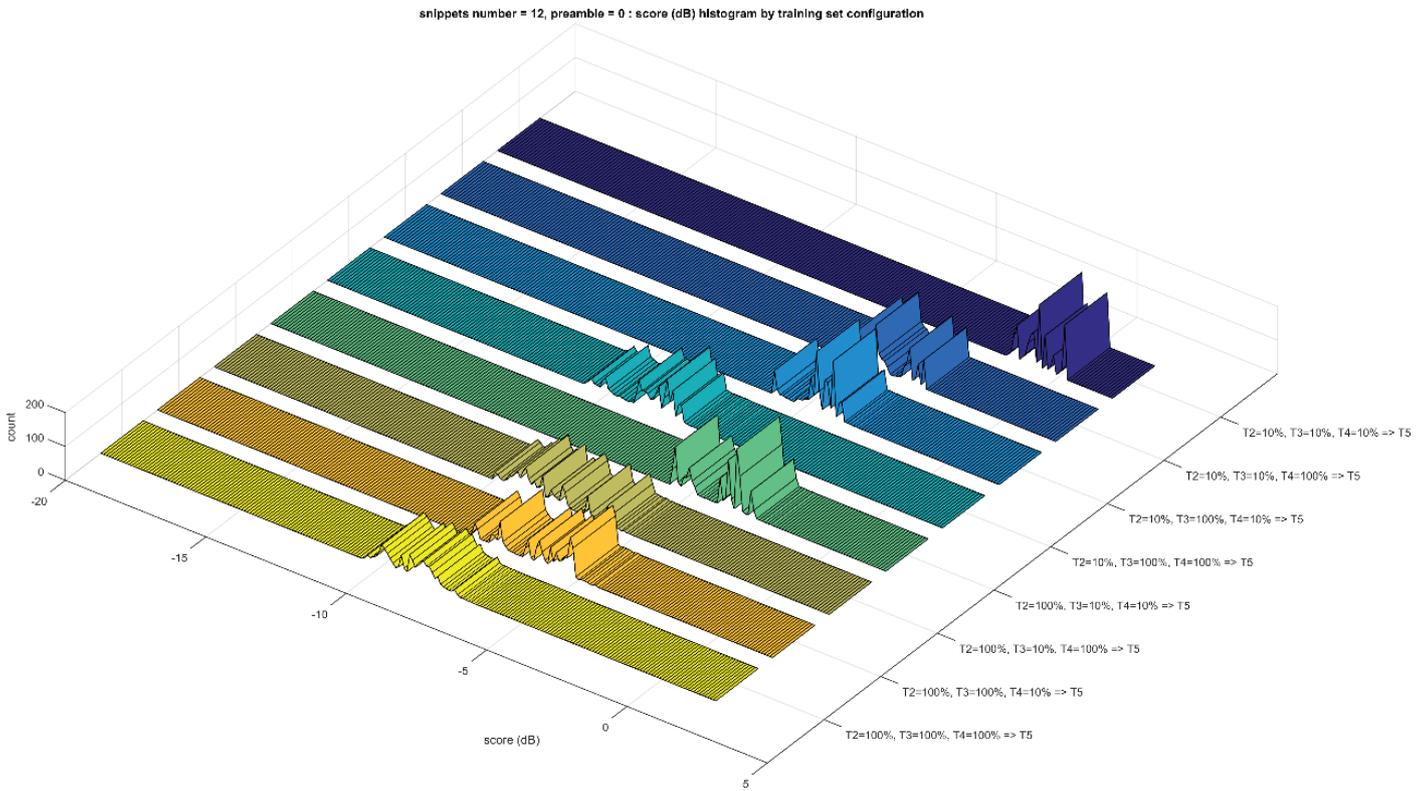


Figure 35: Mean square error histograms for different conditions evaluated in non-autonomous mode

A random replay of snippets based on 100 % of trial 2,3 and 4 leads to the lowest error. Other possible combination of sequences in the snippet generation suggest that some sequences are more informative than others and combining several trials for generating snippets have a positive effect on performances. It will be studied more in details in section 2.3. When evaluated in autonomous generation mode, results are not well differentiated (see Figure 36) and the expected place cell activation sequence is not generated at all. This is due to the recurrent nature of the generation process. If an error is observed during the early timesteps of the generation procedure, this erroneous place cell activation pattern is directly injected in the reservoir input neurons. Reservoir state is different, generates a different prediction that might be erroneous or totally unexpected or unrelated to a place cell coding of a location.

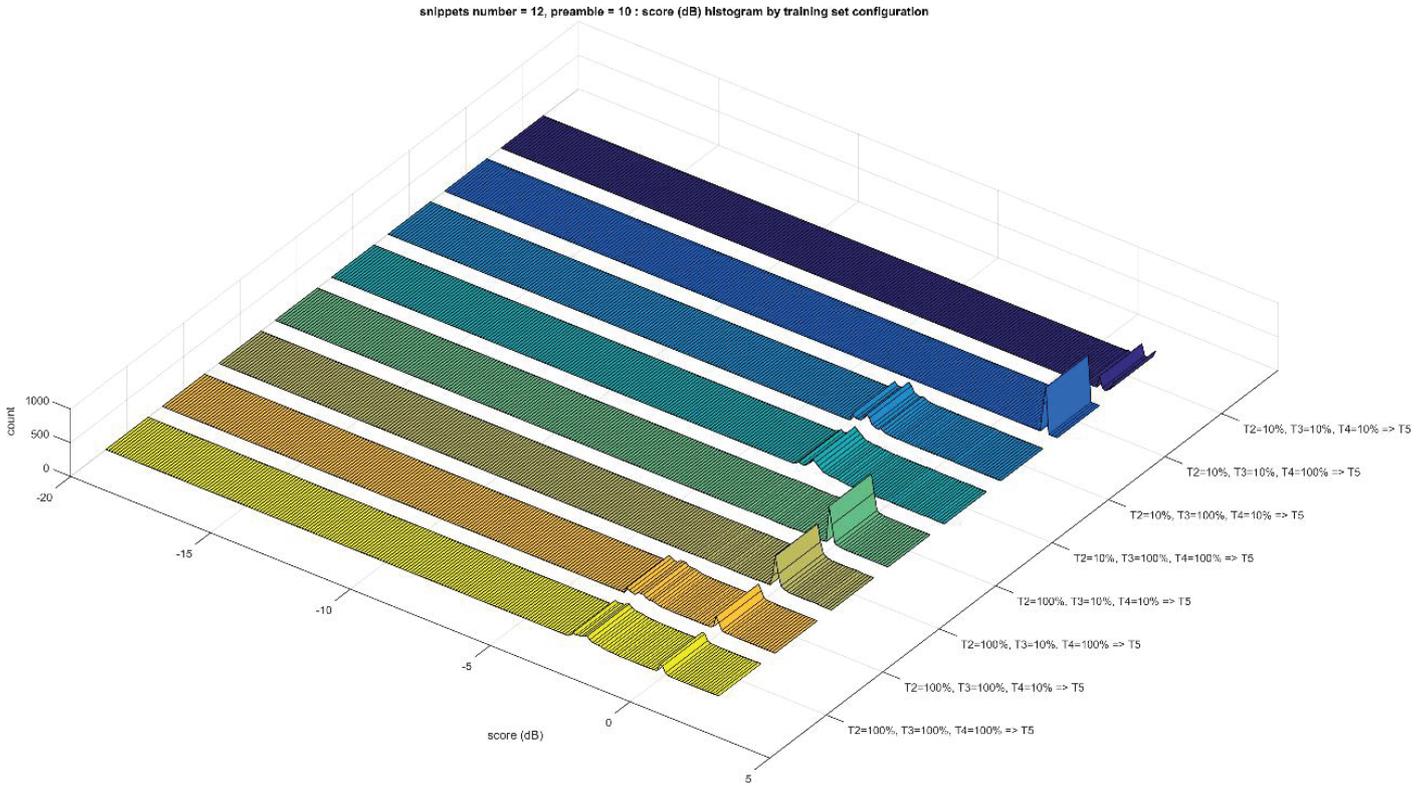


Figure 36: Mean square error histograms for different conditions evaluated in autonomous mode

In order to overcome this difficulty, we propose to implement a form of embodiment. The location encoded by a place-cell activation pattern is estimated by exploiting the spatial response each modelled place cell, the agent moved to this new location and then the place-cell model is applied to the new location of the agent in order to provide an interference free place cell activation pattern. It exploits the fact that an agent situated in a 2D environment can occupy only one location at a time. By selecting the most probable location encoded in a place cell activation prediction, the agent performs an action based on the last perception. This emulate a rudimentary sensory motor loop.

2.3. High performance computing simulator

A Temporal Recurrent Network (TRN) requires training typically 1000 neurons by exposing it repeatedly stimuli sequences. For each simulation cycle, the recurrent connections contribution to the membrane potential of the reservoir neurons is implemented by a matrix vector multiplication and thus requires 1000 000 floating operations with a dense implementation. It is also possible to use a sparse matrix implementation that requires less floating operations if the connectivity matrix is sparse, but the major drawback of this apparent time complexity simplification is that the benefit of using a local and fast paced memory is almost lost because of non-coalesced memory access that result in more frequent access to the remote and slower global memory. We choose to rely on a dense matrix implementation. Depending on the learning rule implemented and the learning rate, the magnitude and the stimuli sampling rate, a successful training requires at typically 10 000 simulation cycles. When instantiating a TRN, the synaptic weights are randomly drawn from a particular distribution with a particular random seed. An error measure on this particular instance cannot be significant with only one observation. In order to be consistent, a batch of TRN must be evaluated instead and this batch size must be empirically at least 10 and evaluating more than 100 TRN for building a single measure won't play a significant role for accuracy.

The training set is not necessarily formed as a time indexed multidimensional stimuli matrix, and might require an additional stochastic training set generation procedure that will drive the TRN during training. This stochastic procedure is in our case the emulation of hippocampus replay during awake SPW-R and it depends

on a particular random seed for a batch of TRN. It is necessary to evaluate at least a group of 10 TRN batch having each different random seeds for the snippet generation procedure initialization.

TRN produces predictions of the next stimulus input. For evaluating the whole sequence that the model can generate, one needs to inject the prediction as the new input and compute the next simulation cycle for updating the TRN states. Each prediction provided by a given TRN instance might reflect a conflict in the training set that results most of the time by two different sequences having the same prefix whose duration exceeds the mnemonic ability of the network. In the case of a stimulus representing hippocampus place-cell coding of a location, the next location prediction is thus not properly encoded and might contain more than a location. Preliminary study described in 2.1.8 states that autonomous generation is not always possible by direct injection of the predicted next stimulus as the new input for the next simulation cycle. We developed and described in section 2.3 a new place-cell location decoder based on a Gaussian kernel and using a finite element method. This decoder is very robust to noise because it exploits the redundancy and coherency of the spatial coding of a place-cell activation pattern over the whole arena. The finite element decoding method requires a grid that tessellates the 2D space defined by the workspace boundaries. A sufficiently low discretization error is required for avoiding a spatial aliasing effect that will necessarily have a negative impact on the generated trajectories. The size of the grid can become important very quickly and have a major impact on the computing load. It is possible to limit the number of grid elements being evaluated by considering a region of interest, centered on the agent current location and enclosing the locations the agent can reach, as defined by its transition model. It is thus possible to select the next location of the agent based on the location probability map computed with the place-cell decoding method, to move the agent to the most probable location and to apply the sensor model in order to provide to the TRN the stimulus that corresponds to the most probable location. Selecting the most probable location at each step is a greedy action selection strategy and does not necessarily lead to the generation of the optimal sequence. However, it is possible to give an account of the trajectories the model can generate by evaluating the model several times and injecting noise in the location probability map in order to allow multiple trajectories to be generated. Typically, 10 to 100 random walks are required for building a consistent 2D high-resolution histogram that will represent the frequencies where a particular location is visited and give an intuitive but not complete view of the set of trajectories the model can generate and their likelihood to be generated. It is also possible to enhance this view by extracting a vector field from the multiple displacements or to cluster trajectories in order to characterize the different classes of trajectories the model can generate.

A trained TRN can be seen as a generative model of trajectories that reflects the training set as represented by the recurrent neural network with a particular set of parameters. These parameters will condition the model's ability to learn a training set and generate a test set. The search space spanned by the Cartesian product of all parameters prohibits a systematic parameter search and one might follow echo state networks guidelines (M Lukoševičius 2012) in order to find a parameter set that allows a particular sequence learning problem to be solved. The authors states that finally, a global optimization algorithm can be used for finding the optimal parameter set for a given task. Tabu search (Glover and Martí 1986), Simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983) and genetic algorithms (Goldberg and Holland 1988) are suitable for this problem and we chose a simulated annealing variation (Chen et al. 2007) because of its implementation simplicity, the low number of parameters and its ability to take advantage of parallel simulations.

For all the reasons mentioned above, simulating multiple agents controlled by a HC-PFC-ST joint neural network model requires a tremendous amount of computing power. Simulating several neural networks exposes several levels of parallelism and it is possible to take advantage of a massively parallel computing infrastructure. Recently, it became possible to use graphics processing units (GPU) for general purpose computing. NVIDIA builds high end gaming graphic cards named Geforce that are perfectly suitable for training deep neural networks in a reasonable and in fact, they can reach the same level of performance of

professional graphic cards. Professional graphic cards (Quadro, Tesla) cost approximately 10 times the price of a gaming graphic card with the same architecture but provides more onboard memory that features an error correcting code (ECC) mechanism, more computing units, half precision arithmetic operations (16-bits floating point number) that doubles the computing throughput, faster double precision arithmetic operations (64-bits floating point number) and drivers certified for specific professional applications. Since artificial neural networks simulations do not require a high level of precision, a recent gaming graphic card is affordable and provides the same single precision (32-bits floating point numbers) computing throughput as approximately 30 recent central processing units (CPU) for the same price. In addition to computing throughput per price unit and numerical precision considerations, the availability of a high performance computing (HPC) infrastructure is another question of interest. Indeed, there exist computing farms that provide an access to computing devices. Some of those are not available for an extended time period and do not authorize the use of 3rd party libraries used in modern HPC approach. It is a serious problem when developing a fully asynchronous and distributed simulator. One need a development cluster with a reasonable computing power and a library and tools ecosystem that allow modern programming and debugging without having to add an extra cost for the time spent on these machines. Another drawback of some distributed computing solutions is that the simulations need to be split into several scripts, launched through a job scheduler on different remote shells. It does not facilitate the simulation process and in addition, scheduler job does not necessarily run immediately, the entire computing power is not available because it is shared with other users and a job might be suspended for hours or days.

We chose to build a hybrid CPU/GPU heterogeneous cluster by using a workstation equipped with a Tesla k40c, upgrading 2 existing workstations with 4 NVIDIA Geforce GTX 1080 and to build a supplementary high end workstation based on the specifications of the DIGITS DevBox by NVIDIA used in the deep learning field and hosting 4 more NVIDIA Geforce GTX 1080 boards. The theoretical computing throughput is 88 590 Gflops / second in single precision and this computing power is available constantly, for an unlimited amount of time. All GPU boards use an exhaust mechanism based on a blower, allowing an easy installation on different motherboards while maintaining a decent cooling performance for a reasonable price. NVIDIA provide a very efficient BLAS implementation, parallel random number generation algorithms and other libraries and tools that reduce significantly the need of developing low-level computing routines and save development time.

There exist deep neural network frameworks for a particular architecture that implement simple recurrent neural networks with a hyperbolic tangent activation function, while frameworks designed for other computing architecture do not provide a RNN with the same specification. State of the art libraries like Microsoft CNTK or Nvidia CuDNN provide different implementation of recurrent neural network but the training algorithm behind are not clearly specified, includes the training of the recurrent layer instead of the readout only required by our echo state approach or simply use a batch of different algorithm and retain the algorithm and parameter set that gave the best result. Deep learning is a new applied research domain and the most advanced libraries at this day were first released in 2015-2016, one year after the beginning of this thesis. We want to keep the model simple, provide clear explanations of their functioning based on several observations of variables internal or external to the model. It does not include a model of hippocampus replay and these of the shelf models are not necessarily online models. Practically speaking, the amount of code dedicated to a neural network model is about 5-10% of the total code volume of our implementation (approximately 35 000 lines of carefully designed and optimized C++ code) whose added value is more than being the fastest possible implementation of a simple RNN. We propose a flexible solution that allow one to take full advantage of the available computing power in a heterogeneous and distributed system. We chose to rely on the same accelerated linear algebra libraries for CPU and GPU named MKL and CUBLAS and to write custom kernels for implementing functions that do not already exist (for instance the Bayesian place-

cell decoding finite element method based on a Gaussian kernel). It is always possible to change the implementation of a model later and use another learning paradigm. Our objective is to test several hypothesis and to explain the internal functioning of a biologically plausible model. Exhibiting the best learning performances possible is not a primary objective but using efficiently the computing power provided by today's modern computing architectures for reducing significantly the time to result is a very valuable advantage.

We will present here the software design of the Temporal Recurrent Network solution. The detailed design part is omitted because it will have required at least 80 pages of pure software engineering. This is not the goal of a thesis manuscript and the following page aim at document the non-trivial and careful design and implementation of a fully asynchronous and distributed neural network simulator that exploits several degrees of parallelism and modern computing architectures while being easily interfaceable with many existing third party simulators. The same simulator can run on a simple laptop or on a modern high performance cluster of computers equipped with the best in slot CPUs and GPUs. It is based on C++ language because of its sophisticated features and its efficiency at execution.

The whole source code is available at <https://github.com/NicolasCAZIN/TRN>

2.3.1. Use case

The most synthetic and important view in a system design is the use case. Figure 37 summarizes the different services offered by the developed solution in an abstract manner:

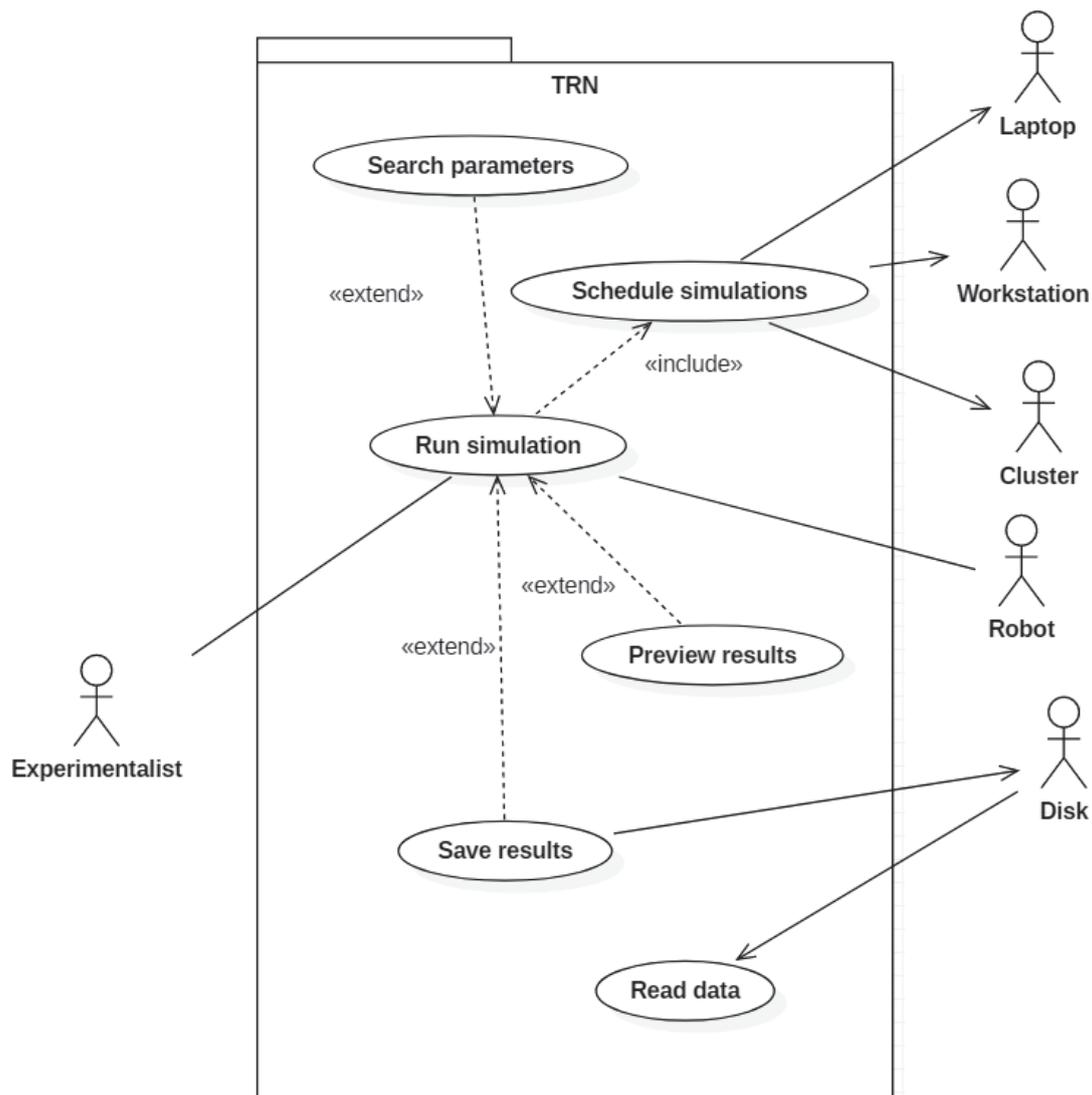


Figure 37: Use case diagram of the TRN solution

The main idea is to provide to an experimentalist an easy way to run simulations of neural network models on different architectures: a simple laptop, a robot, a workstation or a cluster. The underlying functionality for providing this ability to execute simulations on a heterogeneous architecture is to schedule simulations on simulators powered by different computing architecture. It also includes a load balancing mechanism not represented here. The ability to preview quickly results is important when designing a numerical experiment and storing the raw results in an easily usable format is essential for extracting valuable observations on an analysis phase. Reading data generated from 3rd party software like MATLAB is also required for working efficiently in a collaborative scientific work. The experimentalist is required to use a model that works with a given dataset and the model results depend on the parameters and initial conditions. The TRN solution provides a parameter search mechanisms based on simulated annealing for finding a global set of parameters that allows the model to simulate the experiment properly, or to asses that that the problem observed is deeper than a simple parameter configuration issue. A grid search mechanism is also provided for studying the effect of a limited set of parameters on several instances of the model. The elaboration of a model is iterative and requires many trial-error steps. The ultimate goal of the TRN system is to facilitate this tedious work.

2.3.2. Design

We choose to design the TRN solution as a library with different language bindings and a limited number of executables programs, extensible thanks to a plugin system. The architecture is based on the Model View Presenter (MVP) pattern and allows one to easily extend or use TRN.

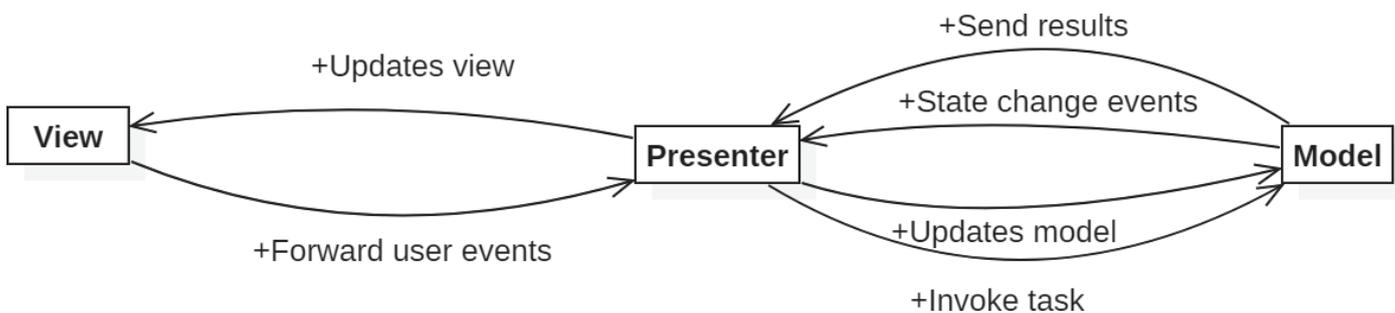


Figure 38: The Model View Presenter architecture used by TRN

The user sees the *View* updated by the *Presenter* and manipulates indirectly the *Model* by using the interface (see section 2.3.2.1) provided by *Presenter*, which also conveys events and data generated by the *Model*. In our case, the user can be a human user that previews the results through observables provided in real time by the model and rendered with a graphical library or it could be another simulator that interacts with TRN by using the same programming interface.

2.3.2.1. Application Programming Interface

The entry point of the TRN solution is an application programming interface acting as a Facade design pattern and provides to the programmer the tools required for using an extending TRN through a hierarchy of interfaces illustrated in Figure 39.

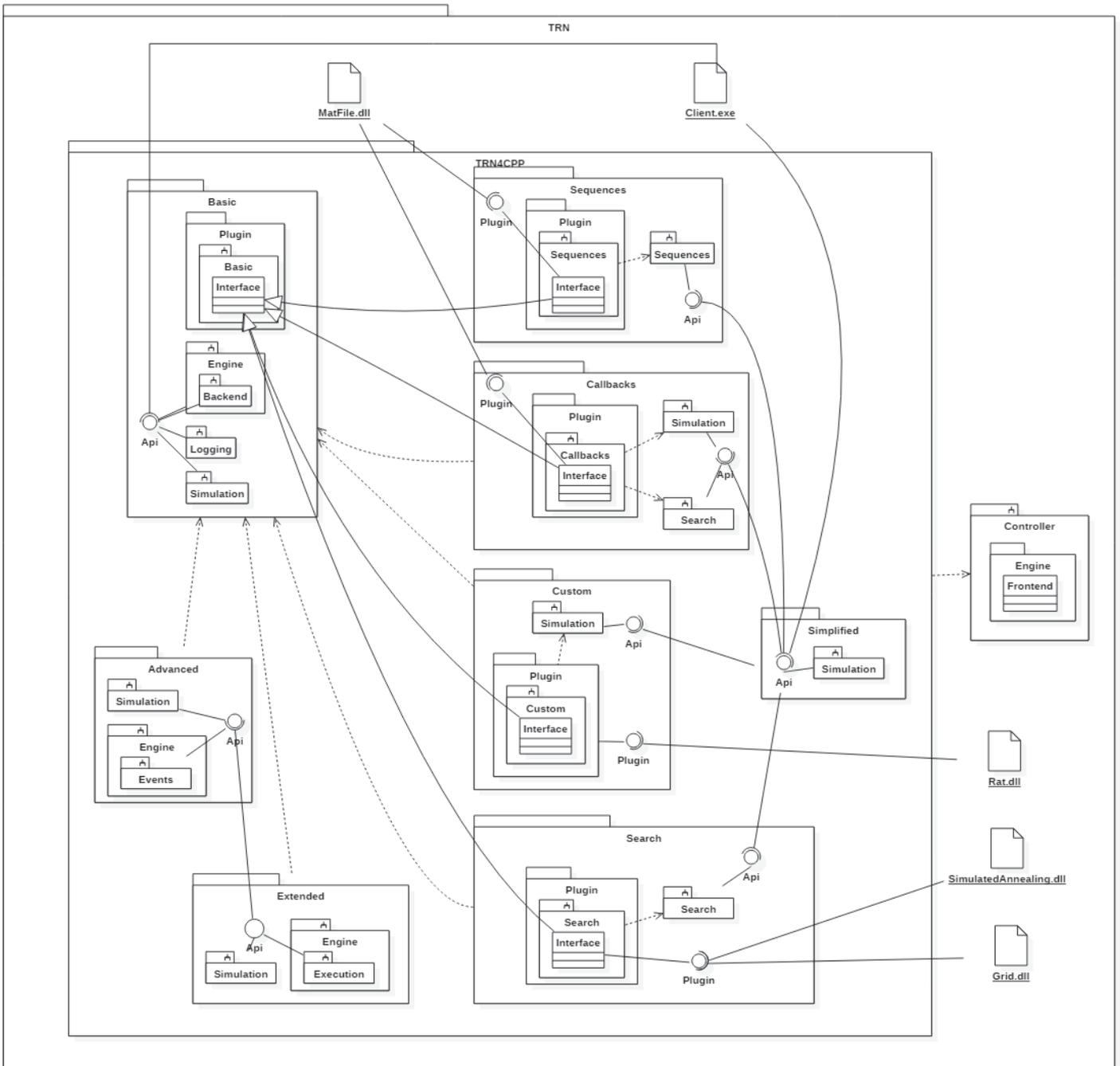


Figure 39: Different interfaces organized as different namespace

Several interfaces are available and they provide to the programmer different levels of complexity, allowing him to implement the tools that participate to the realization of use cases described in Figure 37. The interfaces hierarchy is articulated around asynchronous function calls and several interfaces allow one to exploit the system in a simplified manner while other interfaces allow the programmer to defined extension points, exploited in a transparent manner by the system. It could be described as follows:

- The *Basic* api allows the programmer to instantiate an engine based on a particular computing architecture handled the *Backend*. It is also possible through this interface to setup the logging subsystem and to access to common support functions allowing one to refer to a particular simulation when calling other interface functions. The base class for the plugin interface is defined here. A plugin is required to implement a standard initialization based on text arguments and uninitialization routine
- The *Sequence* api is used for declaring sequences in the system and the plugin interface provides methods that will use transparently the *Sequence* api and require the plugin programmer to implement

methods for accessing a sequence in a dataset according to a hierarchical pointed notation contained in a standard string.

- The *Search* api is used for defining global optimization algorithm supposed to find the optimal set of parameters for a particular experimental condition. The plugin interface provides to the programmer a standard manner of implementing a global search algorithm through a restricted set of methods.
- The *Advanced* api is based on the *Basic* interface and defines the whole set of functionalities accessible to the user.
- The *Callbacks* api defines the set of methods that conveys the results provided by a simulator through the arguments of these methods. The plugin interface provides a standard and transparent manner for a program for being called back when a result is available and an action is required. It is possible to attach to the TRN system several plugins implementing the *Callbacks* interface. They will be evaluated in parallel.
- The *Custom* api aggregates the Facade functions required for building custom model parts requiring the user to reply to the request of the simulator. This is used for example for defining a custom encoder model where the next location prediction will be used as a target location for a mobile robot. Once the mobile robot reached this target point, the custom encoder model will provide to the simulator its estimated position through a callback. The plugin interface allows one to extend the TRN system with custom plugins and it uses transparently the *Custom* api. Only one custom plugin at a time is allowed.
- The *Extended* api is based on the *Advanced* api allows one to define a simulation without necessarily requiring the use of *Custom* and *Callbacks* interfaces. It provides the methods for building and running a simulation by encapsulating calls to *Extended* the api and use when required the functors declared through *Custom* and *Callbacks* interfaces.
- The *Simplified* interface provides a unique method named *compute()* that allows the user to specify a scenario file written in XML, JSON, INI or INFO format. It is a complete simulation procedure that is able to use the full set of methods exposed by the *Extended Custom*, *Callbacks*, *Search* and *Sequence* apis. It is either possible to provide implementation of these interfaces through C++ functors or by specifying in the scenario file a plug-in implementing the corresponding interface. This is the easiest way to run multiple simulations in parallel and to store the generated results in a compressed and structured file format.

2.3.2.2. Model

The role of the *Model* in the MVP architecture is to define a set of objects representing an abstraction of a simulation independent of the implementations, performing the required computations on different computing architectures. We used a *Bridge* design pattern for allowing the definition of two orthogonal classes' hierarchies named the abstraction and the implementation.

The abstraction class hierarchy represents objects that contribute to the definition of a simulation. A *Reservoir* and its unique concrete subclass *WidrowHoff* model specify the methods and attributes relative to the simulation of the consolidation model described in Figure 7. Neurons states and synaptic weights aggregates are implemented by a *Matrix*, *Batch* or *Bundle* that are initialized by classes deriving from *Initializer*. A *Loop* class allows one to evaluate the *Reservoir* once trained. *Copy* is the simplest *Loop* possible and copies the readout activation pattern in the input layer of the *Reservoir*. The *Custom* loop allows one to implement its own *Loop* by providing a callback function and *SpatialFilter* represents a loop with an additional representation decoded from the readout activation pattern by a *Decoder* and encoded before reinjection by an *Encoder*. *Custom* represents a user specified encoder and *Model* is an isotropic Gaussian place-cell model. A *Decoder* represents a place-cell decoding method. *Linear* represents the linear decoding method based on the standard dot product; *Map* is the kernel dot product Bayesian method described in section 2.4 that uses an arbitrary firing rate map and *Model* is the decoding method that uses the same place-cell model as the *Model*

decoder. A *Scheduler* computes a *Scheduling*, which is concatenation of the time indexes of the sequences contained in a training set, represented by the class *Set*. The *Tiled* scheduler is the simplest one and consists in repeating several times the contiguous time indexes of an input sequence. *Snippets* represent the replay model described in Figure 7 and *Custom* allows the user to specify its own scheduler in a callback. A *Measurement* performs gather observations either on the readout layer or on the *Decoder* output.

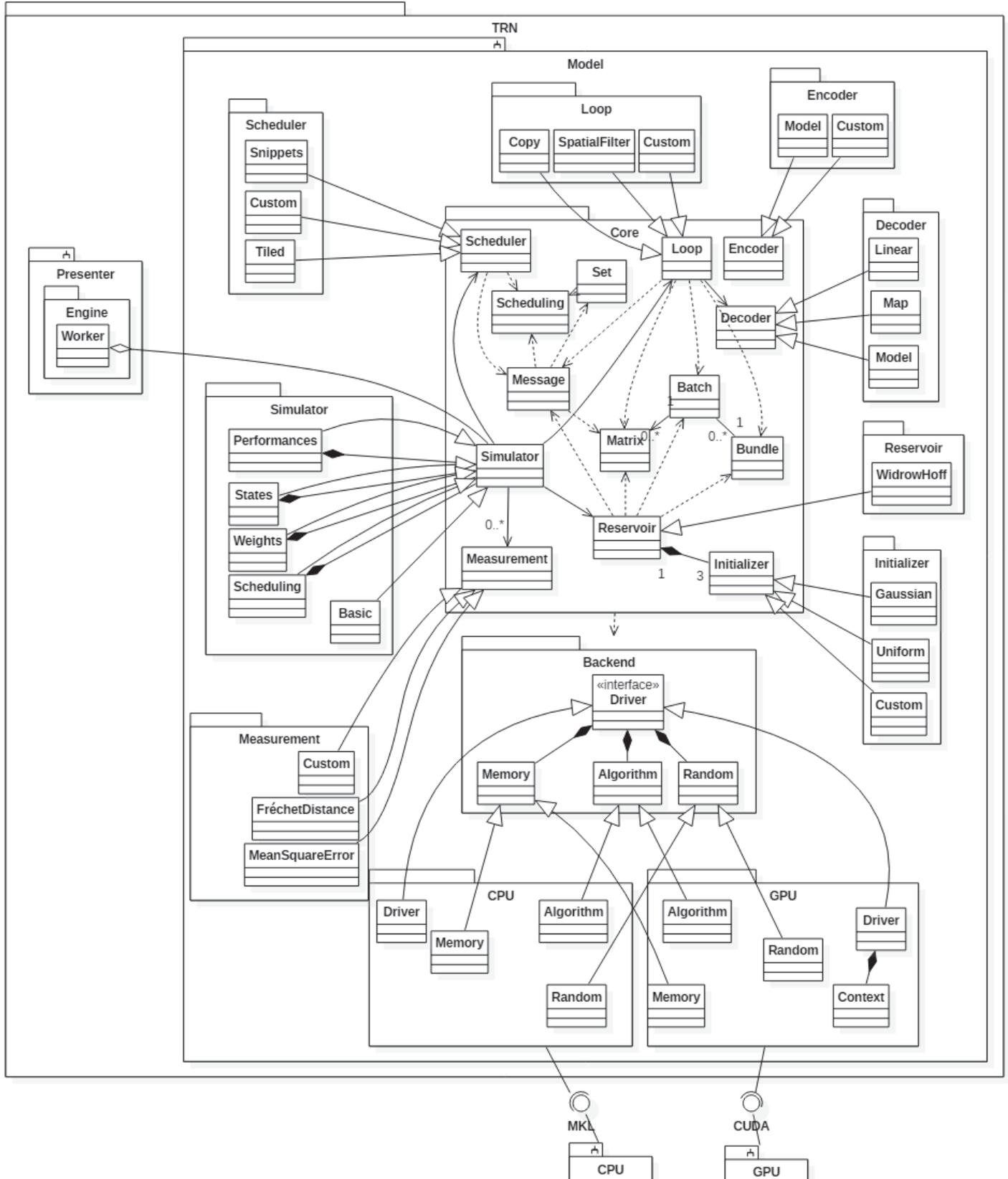


Figure 40: Class diagram of the overall model simulating multiple animats

A *Simulator* holds references to the classes aforementioned and represents one to several instances of the same simulation initialized with different random seeds. This allows one to simulate in parallel several animats on the same computing device. The *Message* class defines the internal communication interface shared by the abstraction classes implementing an *Observer* design pattern.

All the abstraction classes are implemented by several support classes named *Memory*, *Random* and *Algorithm* that are specific to a computing device. They are owned by a *Driver* that allows every class from the abstraction hierarchy to be a *Bridge* and admit either a *CPU* or a *GPU* implementation. Thus, a simulator could be instantiated and executed transparently on the computing device available on the target machine. It is possible to extend the supported architectures by writing a supplementary implementation of a *Driver*. For instance, it could be implementation targeting a modern FPGA or a more generic but less performant implementation based on OpenCL.

Several decorators of a *Simulator* are available and allow one to instrument transparently and optionally a simulation without affecting the performances when not required. The *Performances* decorator measures and report the simulation speed in number of cycles per second (Hz) and the achieved computing throughput (single precision Gflops/s), the *Scheduling* decorator reports the dynamic training set as the time indexes generated by a *Scheduler* during a training episode. *States* and *Weights* report the neurons and synaptic weights of the *Reservoir* object.

2.3.2.3. View

The view allows the user to observe data provided by the model and a view component must implement an interface able to decode a formatted stream of data. This stream of data can come from the calling parameter of a user implemented callback declared through dedicated parts of the Application programming interface, or can be a simple hypercube stored into NetCDF or MATLAB format for instance. A 3rd party software like ParaView or MATLAB might be used for examining the results and adjusting a scenario file that will describe how the controller must be used in order to simulate a particular experiment split in multiple conditions. The only TRN components related to the view part of the model are the plugins called *Monitor* and *Display* and they both implement the *Callback* interface.

```

C:\Users\cazin\OneDrive\Documents\Visual Studio 2015\Projects\TRN\vx64\Release\Client.exe
INFORMATION [2018-Jun-26 14:13:33.080457] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #5, trial #1, train #1, test #1, repeat #6, phase = PRIME, speed = 33122.421875 Hz, throughput = 123.103050 GFlops/s.
INFORMATION [2018-Jun-26 14:13:33.097974] Monitor@DEATHSTAR -> Callbacks::callback_measurement_position_raw : frontend #0, condition #1, batch #6, trial #1, train #1, test #1, repeat #5.
INFORMATION [2018-Jun-26 14:13:34.137542] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #6, trial #1, train #1, test #1, repeat #5, phase = GENERATE, speed = 8699.427734 Hz, throughput = 840.447083 GFlops/s.
INFORMATION [2018-Jun-26 14:13:34.153102] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #6, trial #1, train #1, test #1, repeat #6, phase = PRIME, speed = 32721.613281 Hz, throughput = 121.613403 GFlops/s.
INFORMATION [2018-Jun-26 14:13:34.168638] Monitor@DEATHSTAR -> Callbacks::callback_measurement_position_raw : frontend #0, condition #1, batch #8, trial #1, train #1, test #1, repeat #5.
INFORMATION [2018-Jun-26 14:13:35.234215] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #8, trial #1, train #1, test #1, repeat #5, phase = GENERATE, speed = 8078.153320 Hz, throughput = 780.426147 GFlops/s.
INFORMATION [2018-Jun-26 14:13:35.254743] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #8, trial #1, train #1, test #1, repeat #6, phase = PRIME, speed = 33658.984375 Hz, throughput = 125.097244 GFlops/s.
INFORMATION [2018-Jun-26 14:13:35.270276] Monitor@DEATHSTAR -> Callbacks::callback_measurement_position_raw : frontend #0, condition #1, batch #7, trial #1, train #1, test #1, repeat #5.
INFORMATION [2018-Jun-26 14:13:36.304457] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #7, trial #1, train #1, test #1, repeat #5, phase = GENERATE, speed = 8579.210938 Hz, throughput = 828.833069 GFlops/s.
INFORMATION [2018-Jun-26 14:13:36.322490] Monitor@DEATHSTAR -> Callbacks::callback_performances : frontend #0, condition #1, batch #7, trial #1, train #1, test #1, repeat #6, phase = PRIME, speed = 31403.033203 Hz, throughput = 116.712761 GFlops/s.
INFORMATION [2018-Jun-26 14:13:36.342011] Monitor@DEATHSTAR -> Callbacks::callback_measurement_position_raw : frontend #0, condition #1, batch #5, trial #1, train #1, test #1, repeat #6.

```

Figure 41: An example of the monitor plugin displaying performances information on the standard output.

Monitor provides a simple text summary of the different events generated by the model and *Display* renders trajectories, neurons rasters and sequence scheduling information. OpenCV is used for its windowing and rendering primitives.

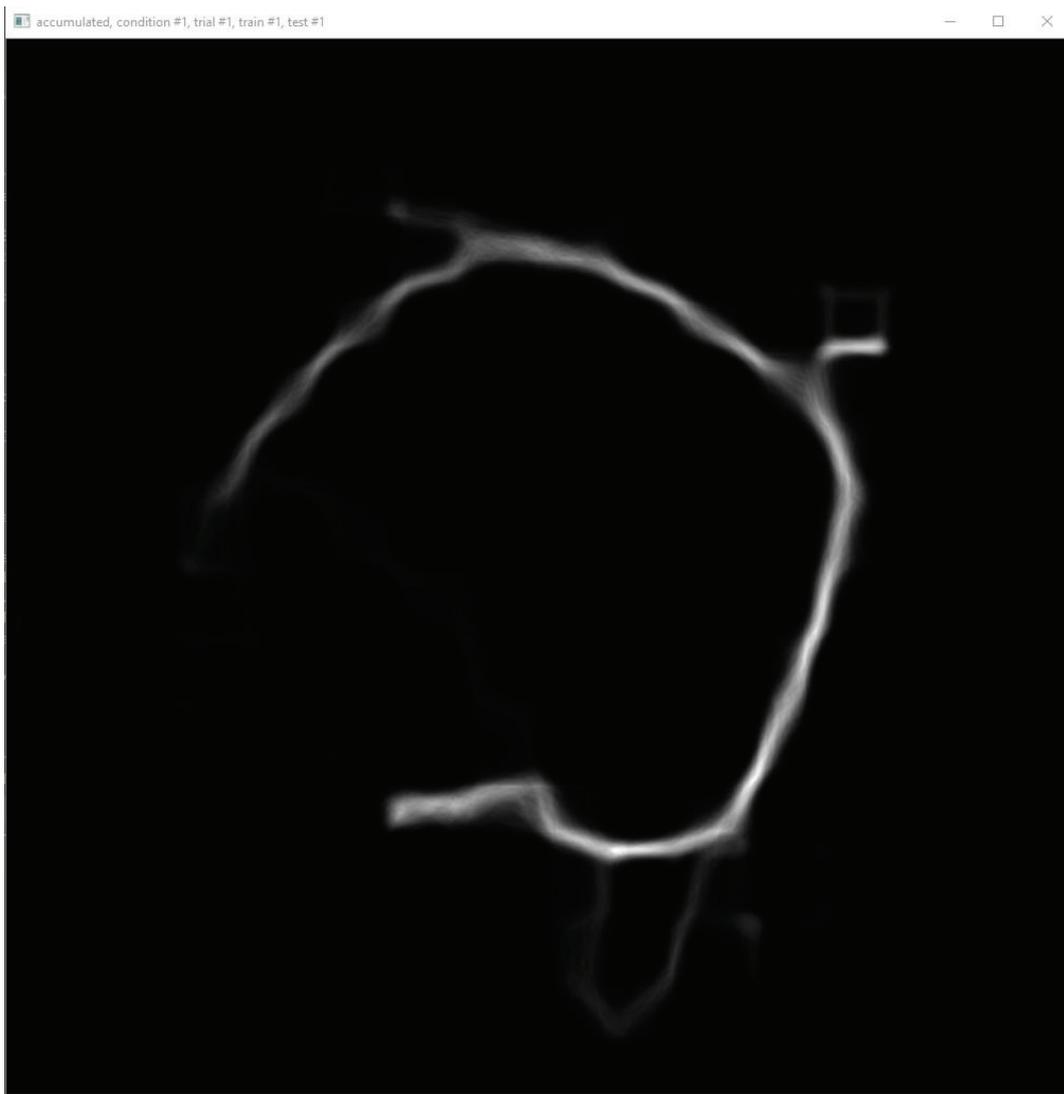


Figure 42: An example of the *Display* plugin rendering multiple trajectories computed by several simulators using 4 GPUs

The view part is not really developed in TRN and is left to third party software like MATLAB or ParaView specialized in scientific data visualization.

2.3.2.4. Presenter

The presenter's role is to mediate the user's events captured by the *View* and *Simulators* maintained by the *Model*. A *Worker* holds zero or several *Simulators* backed by the same *Driver* instance. It is particular *Node* whose role is to receive incoming *Messages* provided by the *Communicator* and to translate them into the appropriate *Simulator* method calls. A *Node* is a particular *Task* that maintains a *Cache* for received data and is implemented by a shared memory mechanism. This allows multiple *Node* running on the same computer to mitigate the communications by avoiding the *Broker* to send an already sent data chunk having the same checksum. A *Broker* is a particular task whose role is to send asynchronous messages to several *Nodes* represented by *Processors* whose life cycle and scheduling is managed by a *Manager*. A *Communicator* might use a *Compressor* for reducing the *Message* traffic between a *Broker* and its corresponding *Nodes*. It serializes *Messages* and establishes a communication channel between a *Broker* and one or more *Node*. *Messages* propagates through a local message queue with a *Local* communicator, a remote TCP/IP socket with a *Remote* communicator and over MPI channels with the *Distributed* communicator. When using a *Remote*

communicator, a *Proxy* is instantiated instead of a *Worker*. Its role is to behave like a *Worker*, but instead of using a *Simulator*, it will act as a regular *Broker* by implementing the *Node* methods with calls to *Dispatcher* methods. The *Dispatcher* owns another communicator that establishes a relationship with a *Worker*. This communicator polymorphism allows one to deploy TRN over a heterogeneous and distributed infrastructure.

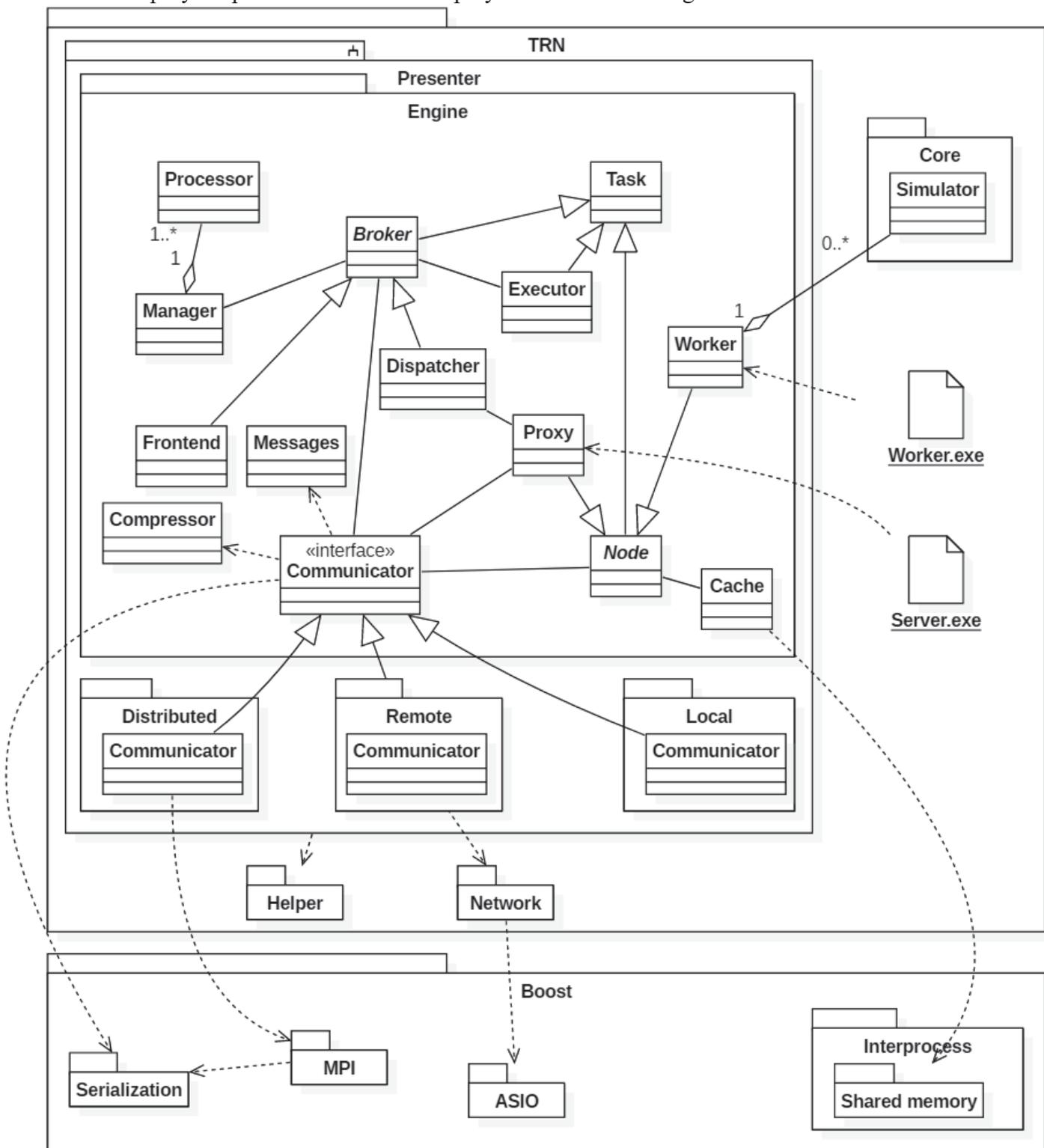


Figure 43: Class diagram of the Presenter layer and related artifacts

Two artifacts provide the TCP/IP and MPI connectivity and allow one to adapt the deployment on different and heterogeneous computer architectures mediated by a network:

- Worker.exe is an executable that instantiates a unique *Worker* and a *Distributed* communicator. It is used for declaring an MPI process that will exploit a particular computing device on a given host. It is also possible to oversubscribe the same computing device by specifying several times the same

computing device on the same host. It might be required for reaching the maximum performance possible in specific cases.

- **Server.exe** is an executable that is specifically used when the host executing the Client side software is a lightweight computer with almost no computing power and dedicated to monitoring or recording results. It could be deployed on a remote workstation equipped with several CPUs/GPUs or on a more modest computer that will play the role of the MPI cluster controller while providing a TCP/IP interface.

Several deployment examples are described in section 2.3.5.

2.3.3. Components

The MVP architecture described above result in several components and artifacts described in Figure 44:

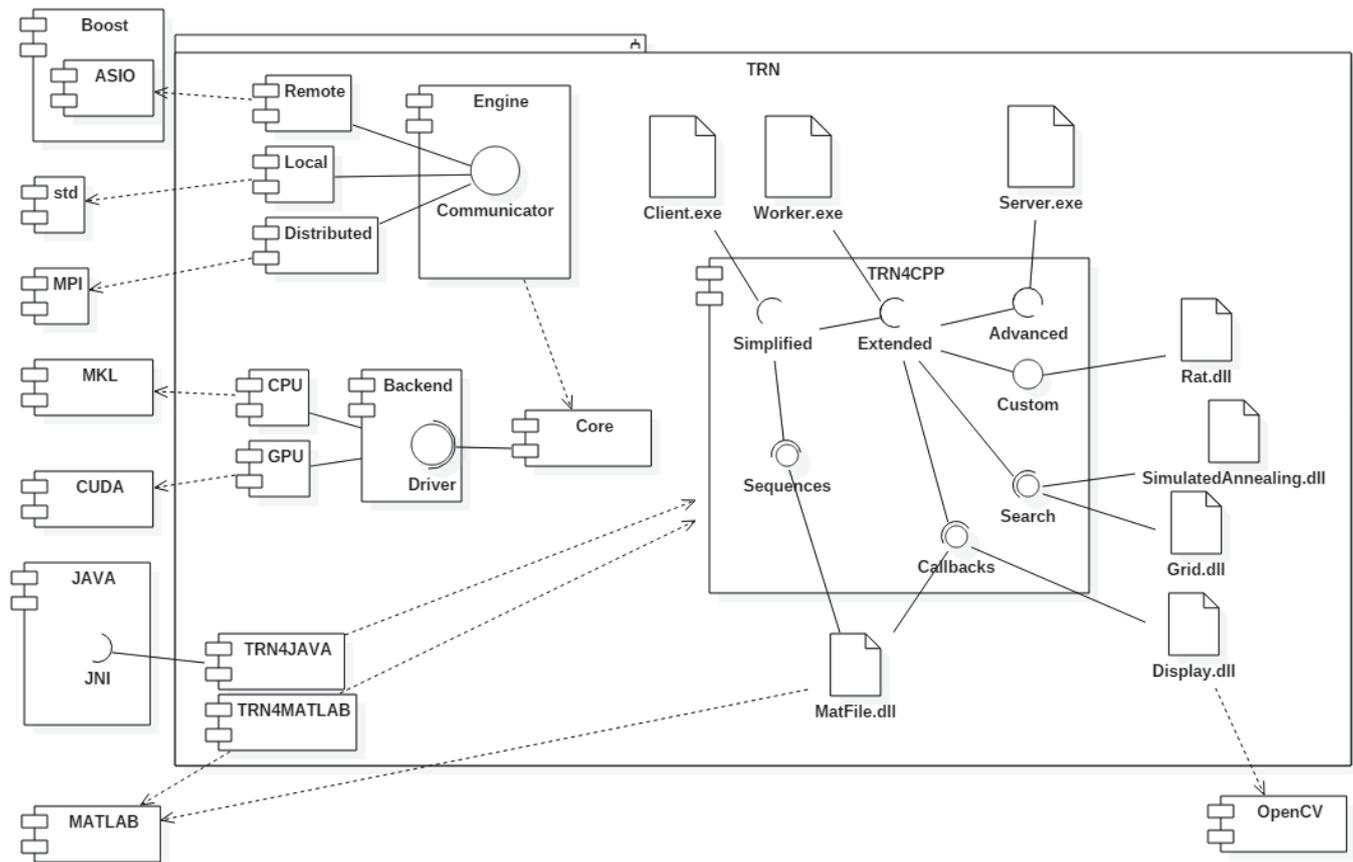


Figure 44: Component diagram of the TRN solution

The *Client.exe* program uses the *Simplified* interface and allows the user to compute simulations. Simulations could be executed on the same host by using a *Local* communicator and exploit transparently the computing power of a *CPU* through the *MKL* library or the *GPU* through the *CUDA* toolkit. It is also possible to offload the execution of simulators on a remote computed through a *Remote* communicator that will be used also by the *Server.exe* artifact for acting as a *Proxy* between the *Local* communicator of the remote host executing *Server.exe* and the *Remote* communicator of the local host executing *Client.exe*. *Worker.exe* is deployed and installed on computer nodes being a part of an MPI cluster. *Rat.dll* simulates a rat by implementing *Custom* callbacks related to the position and place-cell activation pattern, *SimulatedAnnealing.dll* and *Grid.dll* provide global parameter search algorithm by implementing the *Search* interface. *Display.dll* is intended to offer a fast graphical preview of results by using *OpenCV* for implementing the *Callbacks* interface. *MatFile.dll* implements the *Sequence* interface for providing stimuli sequences to TRN from a MATLAB .mat file and implements the *Callbacks* interface for recording result in a structured .mat file.

2.3.4. Language bindings

TRN4CPP is a C++ interface and provide all functionalities to a C++ program. *TRN4MATLAB* is a component that allows a MATLAB script to use TRN through the *Simplified*, *Sequences* and *Basic* API (see Figure 45)

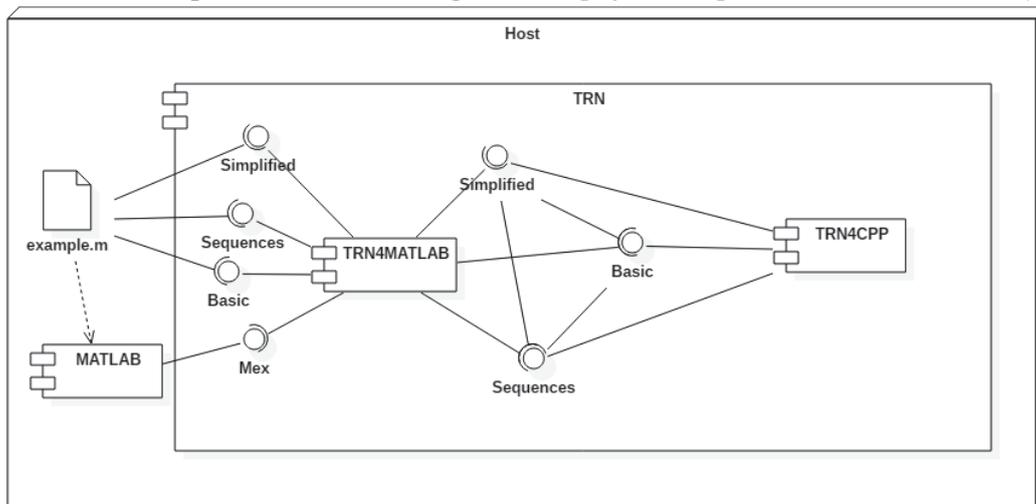


Figure 45: The TRN4MATLAB component provide to a MATLAB script a limited set of TRN4CPP interfaces

TRN4MATLAB is based on the *MEX* interface provided by *MATLAB*. Other interfaces involving asynchronous callbacks are not implementable because *MATLAB* does not support multitasking. *TRN4JAVA* is the Java binding of most of the *TRN4CPP* interfaces. It relies on the *JNI* interface of a Java virtual machine and allows a third party simulator to use TRN and to interact with it through Java interfaces provided by *TRN4JAVA*. Figure 46 illustrates the TRN4CPP interface Java bindings.

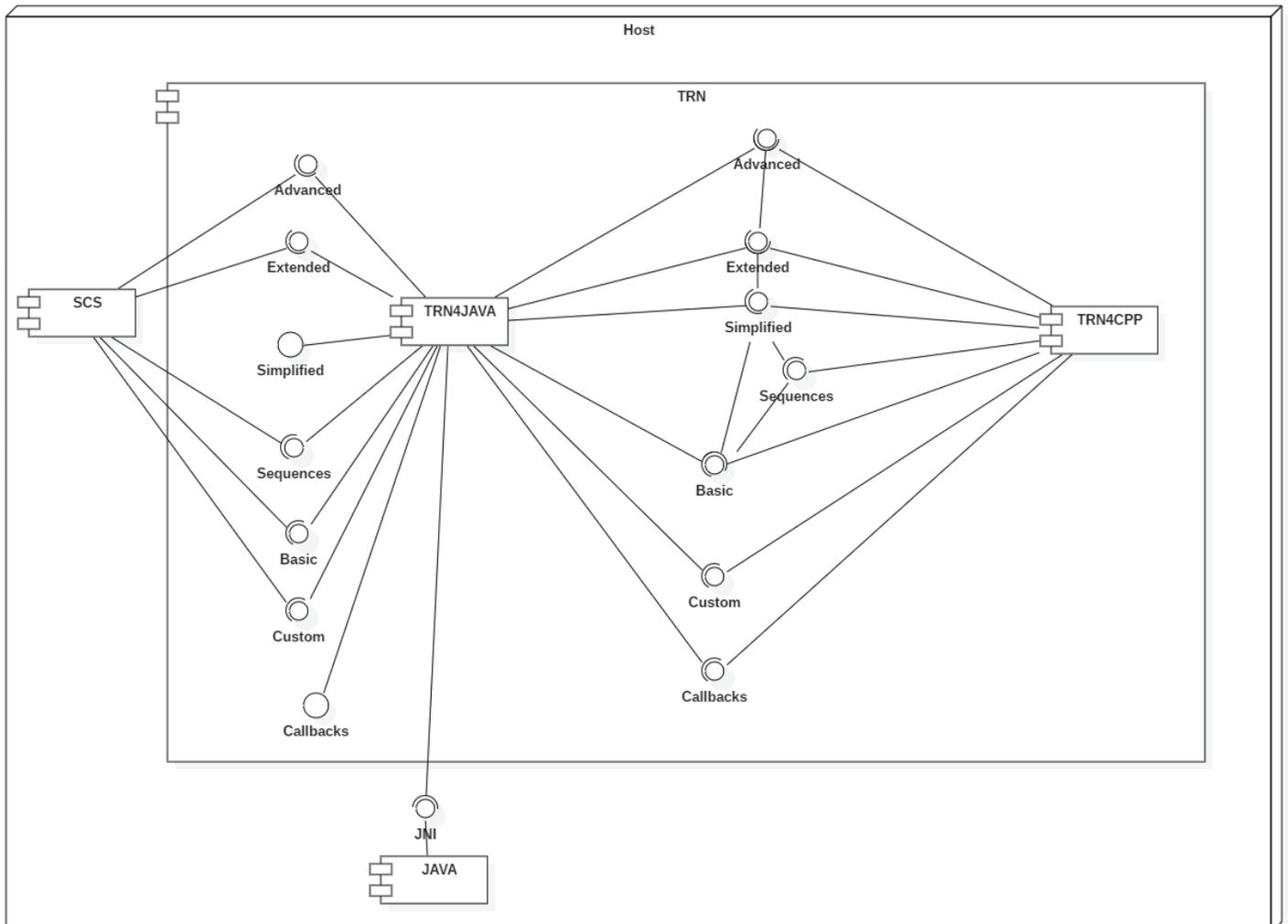


Figure 46: TRN4JAVA allows one to interface a third party simulator (SCS, a robotic simulator from USF Tampa)

2.3.5. Deployment

The components and artifacts provided by TRN allow the user to launch simulations in various situations while taking advantage of computing accelerators like modern GPUs. The simplest deployment possible is on a laptop computer, equipped with a modest CPU and or GPU. Figure 47 illustrates the relationship between components, interfaces and artifacts.

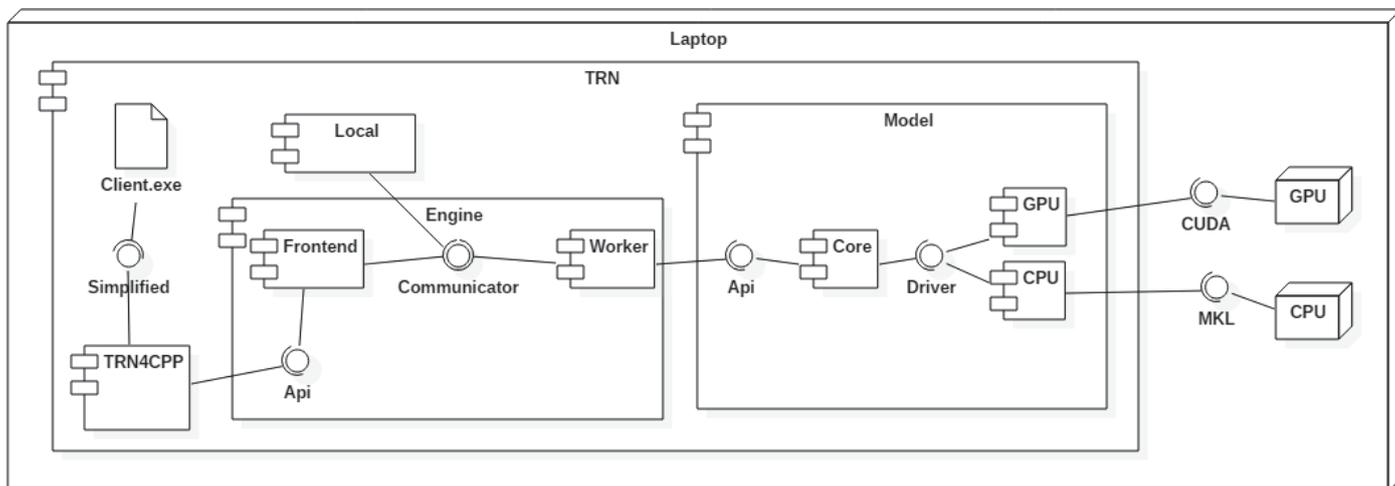


Figure 47: Deployment on a standalone computer

The *Client.exe* program is invoked on a computer and is configured for using a local communicator, allowing the user to launch multiple simulations on the CPU and GPU at the same time. It is not recommended because a significant part of the CPU time will be dedicated to managing the GPU and it will have the effect of slowing down the GPU simulation. If one or more GPU are used, it is recommended not to use the CPU backend. This deployment also accounts for a standalone workstation equipped with one or more GPUs.

Another use case allows a lightweight or embedded computer to offload the simulations to a remote workstation. This is illustrated in Figure 48. A *Remote* communicator is used on the laptop instead of the *Local* communicator in Figure 47.

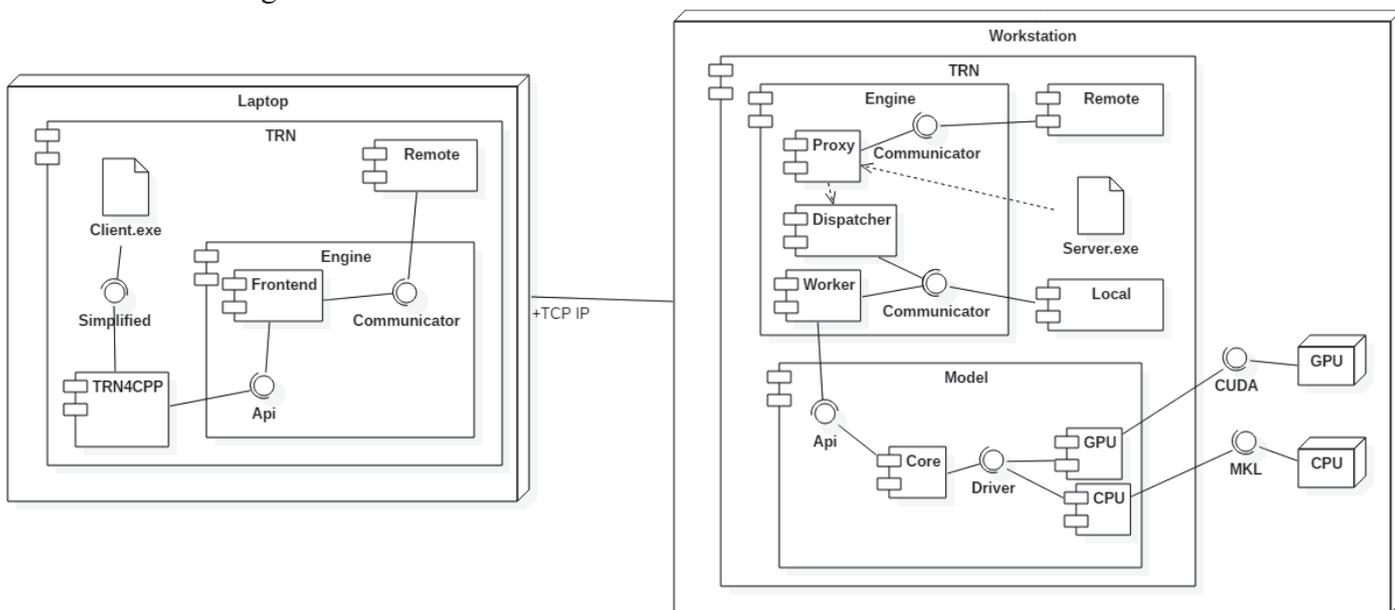


Figure 48: Deployment of a remote Backend for offloading simulations

A robot can execute the *Client.exe*, provide a position estimate and the corresponding place-cell activation and offload the model learning tasks to a remote workstation. It is possible for different *Client.exe* instances to use the same remote *Server.exe*. It means that a workstation can handle the computational load of multiple robot. It is also possible to execute a MATLAB script on a modest computer and to offload the computational

load on a remote workstation equipped with modern computing devices and exploited with a *Local* communicator. Another more sophisticated and deployment would involve one or more workstation running *Worker.exe* in order to define an MPI based cluster. *Server.exe* and *Worker.exe* must be ran with the *mpiexec.exe* or *mpirun.exe* tool provided with the MPI bundle of a particular vendor. It is also possible to use a *Distributed* communicator on the laptop computer by including *Client.exe* in the arguments of the execution MPI tools. If the client software is not *Client.exe* by another client based on the same API, it must be invoked as an MPI process as well. It is not always possible to invoke the client as an MPI process. In this case, a TCP/IP connection must be used for separating the *Frontend* part from the *Backend* part.

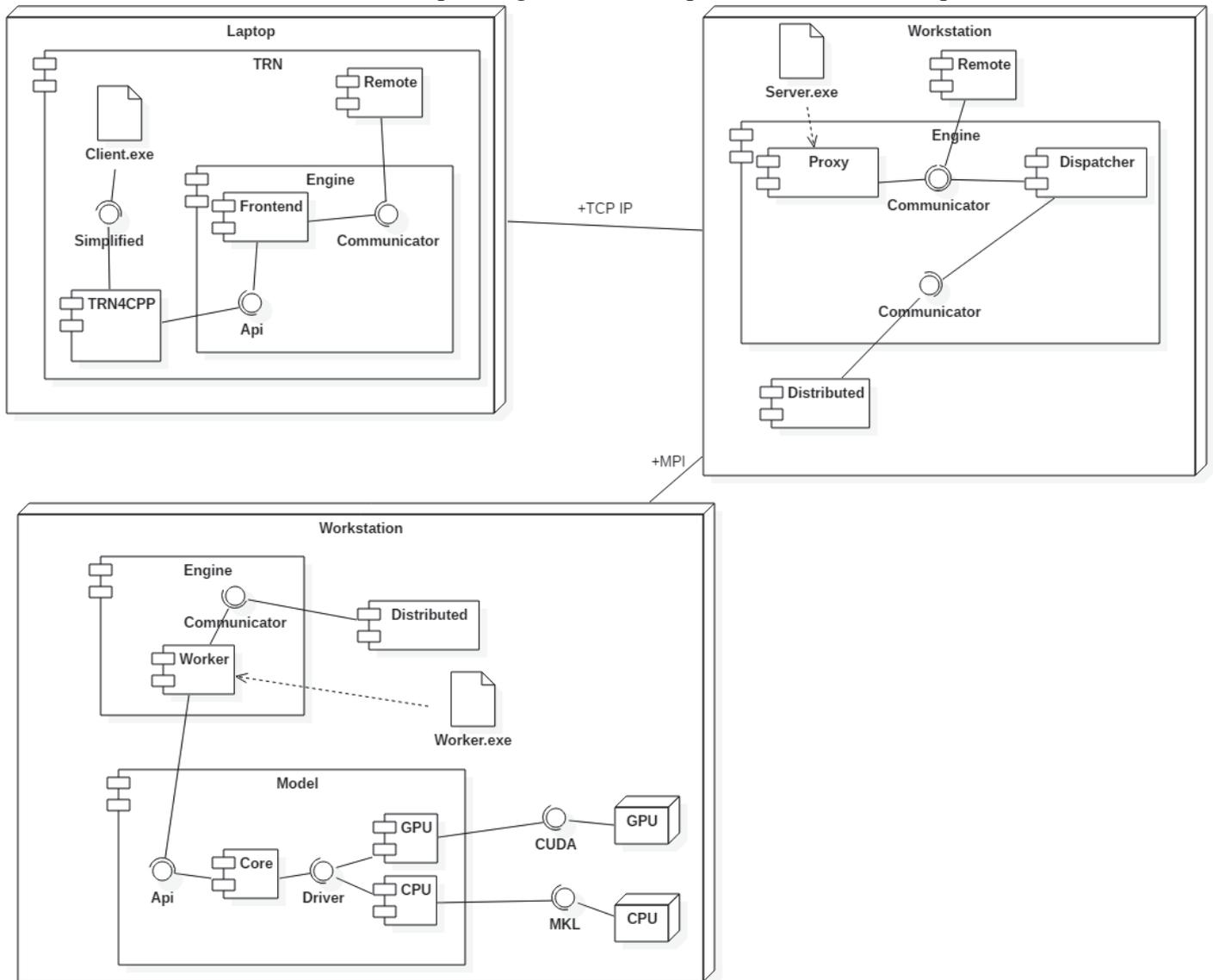


Figure 49: Cluster deployment of TRN with a TCP/IP Server

It is then possible to simulate an important number of scenarios in a limited amount of time by using all the computing resources available over a local area network (LAN). The cluster used for producing all the results described in 2.4 and 2.5 is specified by the deployment diagram in Figure 50. A lightweight laptop executes *Client.exe* is responsible for reading the simulations described in *scenario.xml* and to compute them. Results are stored in a MATLAB mat file and *Client.exe* communicates through a TCP/IP link with *Server.exe* executed on the cluster controller as an MPI process. Computing nodes are equipped with one or more computing device and execute several instances of *Worker.exe* as MPI processes. We propose to evaluate the performances of this cluster by running several simulations in parallel by setting the dimensioning factors evoked in 2.3 and to use the *Performances* decorator for measuring the simulation speed (number of simulation cycles per second) and the single precision computation throughput (number of single precision floating point

operations per second) during the TRAIN, PRIME and GENERATE phase. These 3 phases does not involve the same algorithms and need to be measured separately.

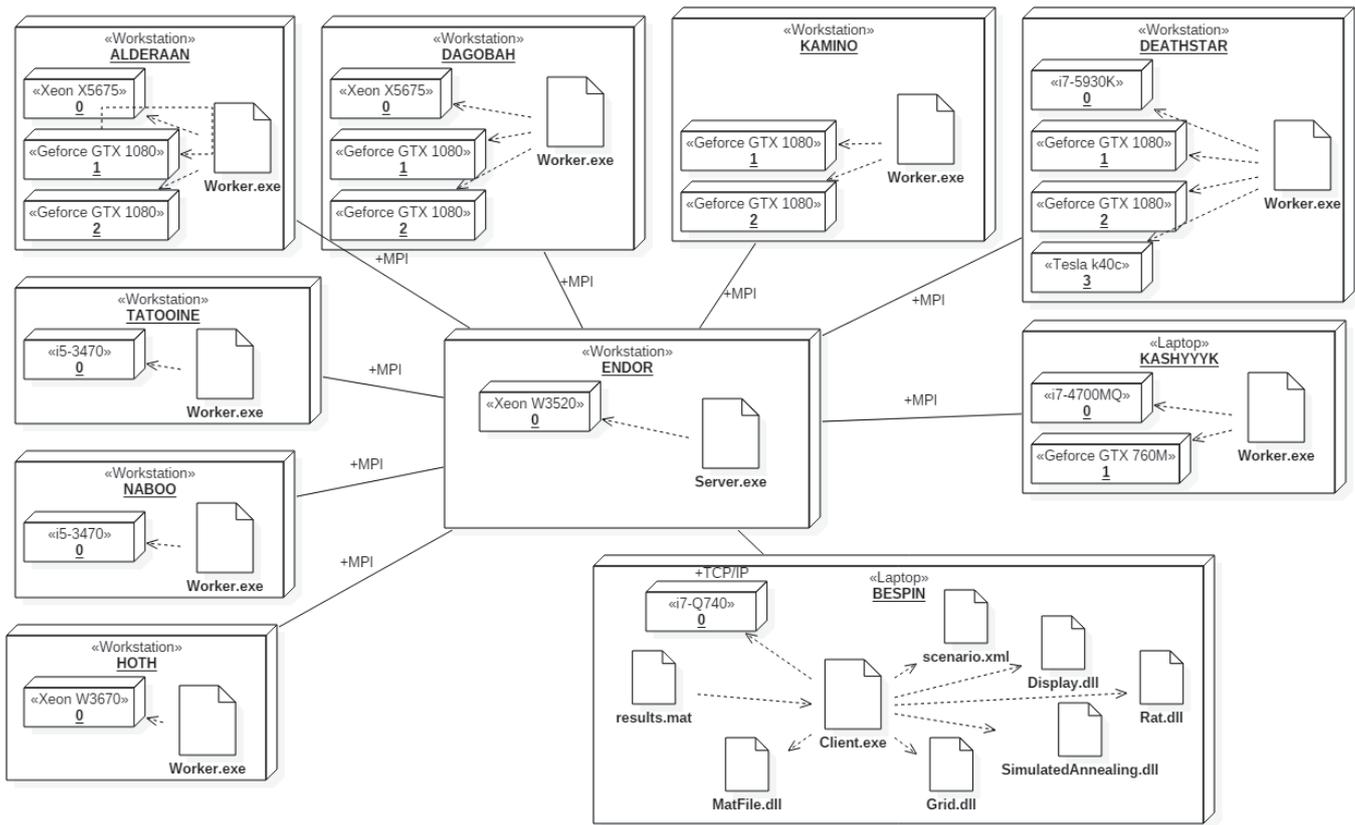


Figure 50: Deployment of the heterogeneous Cluster used in this thesis

The reference simulator used for producing the results described in section 2.1 was based on an optimized CPU version using SSE4.1 instructions and the 12 cores provided by the Xeon X5675. The theoretical throughput of this CPU is 294 Gflops/s. It was not possible to measure the effective throughput but the simulation speed was measured at approximately 2000Hz. This will be the reference of our benchmark. The new simulator implemented use the notion of mini-batch during the training phase. This result in the usage of several matrix-matrix multiplication instead of multiple matrix vector product computed serially. The matrix vector product is bandwidth limited and it is never possible to reach the theoretical computing throughput because too many global memory accesses (slow) are performed regarding the number of arithmetic operations. The CPU version implemented by the CPU driver of TRN is based on MKL and achieves the following performances:

- Train: 2955 Hz @ 9,43 Gflops (1,47x speedup)
- Prime: 2184 Hz @ 8,12 Gflops
- Generate:40Hz @ 192 Gflops

The moderate performance increase of the training phase is due to the use of mini-batch. When using a GTX 1080 with the total drawing power limited to 115%, the maximum throughput an algorithm can reach is 10240 Gflops because of the increased GPU clock. This clock might decrease at one point because the heat produced is too important. This is the maximum peak performance. The same scenario executed on a unique GTX1080 overclocked gives the following performances:

- Train: 52847 Hz @ 168,84 Gflops (26,42x speedup)
- Prime: 34367 Hz @ 127,72 GFlops

- Generate: 1156 Hz @ 9746 GFlops

One can notice that the speedup in training is about 26 times quicker than an optimized CPU version and the theoretical throughput is almost reached during the generation phase. This is explained by the fact that an optimized kernel has been developed and the number memory access was minimized regarding to the number of arithmetic operations required. When running the same experiment in a cluster configuration by using all the modern GPU available (8x GTX 1080 and 1x Tesla k40c overclocked, theoretical throughput 88 590 GFlops), the following performances are observed:

- Train: 454875 Hz @ 1446.5 Gflops (227,43x speedup)
- Prime: 301005 Hz @ 1081.62 Gflops
- Generate: 9993 Hz @ 84236 Gflops (95% efficiency)

The cluster is about 227x faster than the original implementation and it is possible to reach the simulation speed of 454 KHz in training mode. The computing throughput is still limited by the matrix-vector multiplication used in the evaluation of neurons states and it is not possible to compute it faster. The generation mode displays a computing throughput of 84236 Gflops, which correspond to 95% of efficiency. It would be difficult to go further in terms of optimization. Even the optimized matrix-matrix multiplication provided by GPU vendors does not reach the theoretical throughput of their devices. The achieved speedup makes possible new investigations requiring orders of magnitude more computing power. The TRN solution is capable of exploiting the computing power provided by a modest development cluster by using all degrees of parallelism available. It is possible to simulate numerical experiments faster by embedding the TRN solution in containers and executing it in a wider cluster but our low latency, moderate computing power cluster having a fixed cost is sufficient for the remaining experiment.

2.4. Prefrontal cortex reservoir network learns to create novel efficient navigation sequences by concatenating place-cell snippets replayed with spatial credit assignment in hippocampus

PLOS Computational Biology

Prefrontal cortex creates novel navigation sequences from hippocampal place-cell replay - evidence from simulation

--Manuscript Draft--

Manuscript Number:	PCOMPBIOL-D-18-01285
Full Title:	Prefrontal cortex creates novel navigation sequences from hippocampal place-cell replay - evidence from simulation
Short Title:	Prefrontal cortex synthesizes efficient sequences
Article Type:	Research Article
Keywords:	rat; navigation; prefrontal cortex; hippocampus; traveling salesman problem; reservoir computing
Corresponding Author:	Peter Ford Dominey INSERM Bron, FRANCE
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	INSERM
Corresponding Author's Secondary Institution:	
First Author:	Nicolas Cazin
First Author Secondary Information:	
Order of Authors:	Nicolas Cazin Martin Llofriu Alonso Pable Scleidorovich Chiodi Tatiana Pelc Bruce Harland Alfredo Weitzenfeld Jean-Marc Fellous Peter Ford Dominey
Order of Authors Secondary Information:	
Abstract:	As rats learn to search for multiple sources of food or water in a complex environment, they generate increasingly efficient trajectories between reward sites, across multiple trials. This is referred to as the Traveling Salesrat Problem (TSP) (de Jong et al (2011)). Such spatial navigation involves the replay of hippocampal place-cells during awake states, generating small sequences of spatially related place-cell activity that we call "snippets". These snippets occur primarily during sharp-wave-ripple (SWR) events. 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 use a biologically motivated computational framework known as 'reservoir computing' (Enel et al 2016), in which large pools of prewired neural elements process information dynamically through reverberations. This PFC model consolidate snippets into larger spatial sequences that may be later recalled by subsets of the original sequences. We developed a model of snippet generation that is modulated by reward, in the forward and reverse directions. This implements a form of spatial credit assignment for reinforcement learning. Our simulation experiments provide neurophysiological explanations for two pertinent observations related to navigation. Reward modulation allows the system to reject non-optimal segments of experienced trajectories, and reverse replay allows the system to "learn" trajectories that is has not physically experienced, both of which significantly

Powered by Editorial Manager® and Prodxion Manager® from Aries Systems Corporation

	contribute to the TSP behavior.
Suggested Reviewers:	<p>Mike Hasselmo Boston University hasselmo@bu.edu Studies integrated behavior using a combination of neurophysiological and behavioral experiments along with computational modeling</p> <p>Jim Knierim Johns Hopkins University jknierim@jhu.edu His research focuses on the neurophysiology of memory in the hippocampal formation</p> <p>Kamran Diba University of Michigan kdiba@umich.edu Background includes large-scale neuronal recordings from the cortex of freely behaving rats, and applying this to the study of sequential activity patterns in the hippocampus, with a computational approach</p> <p>Kechen Zhang Johns Hopkins University Department of Biomedical Engineering kzhang4@jhmi.edu Computational neuroscientist including focusing on recurrent networks, sequence learning, hippocampus.</p>
Opposed Reviewers:	<p>McNaughton Bruce University of California Irvine brucemcn@uci.edu Unquestionable expertise, but potentially too oriented to experimental vs computational aspects for this paper</p> <p>György Buzsáki New York University Gygyorgy.Buzsaki@nyumc.org Unquestionable expertise, but potentially too oriented to experimental vs computational aspects for this paper</p>
Additional Information:	
Question	Response
<p>Financial Disclosure</p> <p>Please describe all sources of funding that have supported your work. This information is required for submission and will be published with your article, should it be accepted. A complete funding statement should do the following:</p> <ol style="list-style-type: none"> 1. Include grant numbers and the URLs of any funder's website. Use the full name, not acronyms, of funding institutions, and use initials to identify authors who received the funding. 2. Describe the role of any sponsors or funders in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. If the funders had no role in any of the above, include this 	<p>This research was funded by the NSF Under grant 1429929 https://www.nsf.gov/awardsearch/showAward?AWD_ID=1429929, and by the French ANR under grant ANR-14-NEUC-0005-01. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.</p>

Powered by Editorial Manager® and Prodxion Manager® from Aries Systems Corporation

<p>sentence at the end of your statement: <i>"The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript."</i></p> <p>However, if the study was unfunded, please provide a statement that clearly indicates this, for example: <i>"The author(s) received no specific funding for this work."</i></p> <p>* typeset</p>	
<p>Competing Interests</p> <p>You are responsible for recognizing and disclosing on behalf of all authors any competing interest that could be perceived to bias their work, acknowledging all financial support and any other relevant financial or non-financial competing interests.</p> <p>Do any authors of this manuscript have competing interests (as described in the PLOS Policy on Declaration and Evaluation of Competing Interests)?</p> <p>If yes, please provide details about any and all competing interests in the box below. Your response should begin with this statement: <i>I have read the journal's policy and the authors of this manuscript have the following competing interests:</i></p> <p>If no authors have any competing interests to declare, please enter this statement in the box: <i>"The authors have declared that no competing interests exist."</i></p>	<p>The authors have declared that no competing interests exist.</p>

* typeset

Data Availability

Yes - all data are fully available without restriction

PLOS journals require authors to make all data underlying the findings described in their manuscript fully available, without restriction and from the time of publication, with only rare exceptions to address legal and ethical concerns (see the [PLOS Data Policy](#) and [FAQ](#) for further details). When submitting a manuscript, authors must provide a Data Availability Statement that describes where the data underlying their manuscript can be found.

Your answers to the following constitute your statement about data availability and will be included with the article in the event of publication. **Please note that simply stating 'data available on request from the author' is not acceptable. If, however, your data are only available upon request from the author(s), you must answer "No" to the first question below, and explain your exceptional situation in the text box provided.**

Do the authors confirm that all data underlying the findings described in their manuscript are fully available without restriction?

Please describe where your data may be found, writing in full sentences. **Your answers should be entered into the box below and will be published in the form you provide them, if your manuscript is accepted.** If you are copying our sample text below, please ensure you replace any instances of **XXX** with the appropriate details.

All relevant data are within the paper and its Supporting Information files.

- If your data are all contained within the paper and/or Supporting Information files, please state this in your answer below. For example, "All relevant data are within the paper and its Supporting

<p>Information files.”</p> <ul style="list-style-type: none"> • If your data are held or will be held in a public repository, include URLs, accession numbers or DOIs. For example, “All XXX files are available from the XXX database (accession number(s) XXX, XXX).” If this information will only be available after acceptance, please indicate this by ticking the box below. • If neither of these applies but you are able to provide details of access elsewhere, with or without limitations, please do so in the box below. For example: <p>“Data are available from the XXX Institutional Data Access / Ethics Committee for researchers who meet the criteria for access to confidential data.”</p> <p>“Data are from the XXX study whose authors may be contacted at XXX.”</p> <p>* typeset</p>	
<p>Additional data availability information:</p>	

Unité de recherche U1208
Stem Cell and Brain Research Institute



Peter Ford Dominey
Director Robot Cognition Laboratory
Stem Cell and Brain Research Institute
INSERM U1208, 18 ave Doyen Jean Lepine
69675 Bron, Cedex, France

Lyon, July 23, 2018

Dear Editor,

Please find in this submission our paper “Prefrontal cortex creates novel navigation sequences from hippocampal place-cell replay – evidence from simulation” to be considered for publication in PLoS Computational Biology.

We believe the paper is well suited for the readership of the journal. The paper addresses an open issue in rodent prefrontal cortex and hippocampal interaction: how hippocampal replay is integrated by prefrontal cortex in order to generate novel and efficient navigation behavior. In addressing this problem, we propose a new model that introduces a spatial reward bias in replay, which implements a novel and efficient form of reinforcement learning, and allows the simulated PFC to synthesize novel efficient trajectories based on limited experience.

The study should inspire neuroscientists both in the reinterpretation of their data in this new framework, and in the development of new experiments. It should also likewise inspire computational neuroscientists as it opens up a new computing framework for reinforcement learning in the context of reservoir computing.

Sincerely,

A handwritten signature in blue ink, appearing to read "P. Dominey", is centered below the text "Sincerely,".

Peter Ford Dominey, CNRS Research Director

République Française

INSERM – Unité de Recherche U1208
18 Avenue du Doyen Lépine
69675 Bron Cedex, France
Tel : (33)4 72 91 34 75 Fax : (33)4 72 91 34 61
Email : Chandara.Nay@inserm.fr

1 **Title Page**

2 **Title:** Prefrontal cortex creates novel navigation sequences from hippocampal place-cell
3 replay – evidence from simulation

4

5 **Abbreviated Title:** Prefrontal cortex synthesizes efficient sequences

6

7 **Authors:** Nicolas Cazin¹, Martin Llofriu Alonso², Pablo Scleidorovich Chiodi², Tatiana Pelc³,
8 Bruce Harland³, Alfredo Weitzenfeld², Jean-Marc Fellous³, Peter Ford Dominey¹

9 **Affiliations:**

10 1. Human and Robot Cognitive Systems Group, INSERM, France

11 2. College of Engineering, University of South Florida, USA

12 3. Department of Psychology, University of Arizona, USA

13

14 **Corresponding Author:** Peter Ford Dominey

15 **Number of pages:** 70

16 **Number of figures (17), tables (4)**

17 **Conflict of Interest:** No conflict of interest

18 **Acknowledgements:** Work supported by NSF-ANR CRCNS (#1429929, Spaquence)

19

1 **Title Page**

2 **Title:** Prefrontal cortex creates novel navigation sequences from hippocampal place-cell
3 replay – evidence from simulation

4

5 **Abbreviated Title:** Prefrontal cortex synthesizes efficient sequences

6

7 **Authors:** Nicolas Cazin¹, Martin Llofriu Alonso², Pablo Scleidorovich Chiodi², Tatiana Pelc³,
8 Bruce Harland³, Alfredo Weitzenfeld², Jean-Marc Fellous³, Peter Ford Dominey¹

9 **Affiliations:**

10 1. Human and Robot Cognitive Systems Group, INSERM, France

11 2. College of Engineering, University of South Florida, USA

12 3. Department of Psychology, University of Arizona, USA

13

14 **Corresponding Author:** Peter Ford Dominey

15 **Number of pages:** 70

16 **Number of figures (17), tables (4)**

17 **Conflict of Interest:** No conflict of interest

18 **Acknowledgements:** Work supported by NSF-ANR CRCNS (#1429929, Spaquence)

19

20 **Abstract (242 words)**

21 As rats learn to search for multiple sources of food or water in a complex environment, they generate
22 increasingly efficient trajectories between reward sites, across multiple trials. This is referred to as the
23 Traveling Salesrat Problem (TSP) ([de Jong et al \(2011\)](#)). Such spatial navigation involves the replay of
24 hippocampal place-cells during awake states, generating small sequences of spatially related place-cell
25 activity that we call “snippets”. These snippets occur primarily during sharp-wave-ripple (SWR)
26 events. Here we focus on the role of replay during the awake state, as the animal is learning across
27 multiple trials. We hypothesize that these snippets can be used by the PFC to achieve multi-goal
28 spatial sequence learning. We use a biologically motivated computational framework known as
29 ‘reservoir computing’ ([Enel et al 2016](#)), in which large pools of prewired neural elements process
30 information dynamically through reverberations. This PFC model consolidate snippets into larger
31 spatial sequences that may be later recalled by subsets of the original sequences. We developed a
32 model of snippet generation that is modulated by reward, in the forward and reverse directions. This
33 implements a form of spatial credit assignment for reinforcement learning. Our simulation
34 experiments provide neurophysiological explanations for two pertinent observations related to
35 navigation. Reward modulation allows the system to reject non-optimal segments of experienced
36 trajectories, and reverse replay allows the system to “learn” trajectories that is has not physically
37 experienced, both of which significantly contribute to the TSP behavior.

38 **Author Summary (120 words)**

39 As rats search for multiple sources of food in a complex environment, they generate increasingly
40 efficient trajectories between reward sites, across multiple trials, referred to as the Traveling Salesrat
41 Problem (TSP). This likely involves the coordinated replay of place-cell into “snippets” between
42 successive trials. We hypothesize that “snippets” can be used by the prefrontal cortex (PFC) to
43 implement a form of reward-modulated reinforcement learning. Our simulation experiments provide
44 neurophysiological explanations for two pertinent observations related to navigation. Reward

20 **Abstract (242 words)**

21 As rats learn to search for multiple sources of food or water in a complex environment, they generate
22 increasingly efficient trajectories between reward sites, across multiple trials. This is referred to as the
23 Traveling Salesrat Problem (TSP) ([de Jong et al \(2011\)](#)). Such spatial navigation involves the replay of
24 hippocampal place-cells during awake states, generating small sequences of spatially related place-cell
25 activity that we call “snippets”. These snippets occur primarily during sharp-wave-ripple (SWR)
26 events. Here we focus on the role of replay during the awake state, as the animal is learning across
27 multiple trials. We hypothesize that these snippets can be used by the PFC to achieve multi-goal
28 spatial sequence learning. We use a biologically motivated computational framework known as
29 ‘reservoir computing’ ([Enel et al 2016](#)), in which large pools of prewired neural elements process
30 information dynamically through reverberations. This PFC model consolidate snippets into larger
31 spatial sequences that may be later recalled by subsets of the original sequences. We developed a
32 model of snippet generation that is modulated by reward, in the forward and reverse directions. This
33 implements a form of spatial credit assignment for reinforcement learning. Our simulation
34 experiments provide neurophysiological explanations for two pertinent observations related to
35 navigation. Reward modulation allows the system to reject non-optimal segments of experienced
36 trajectories, and reverse replay allows the system to “learn” trajectories that is has not physically
37 experienced, both of which significantly contribute to the TSP behavior.

38 **Author Summary (120 words)**

39 As rats search for multiple sources of food in a complex environment, they generate increasingly
40 efficient trajectories between reward sites, across multiple trials, referred to as the Traveling Salesrat
41 Problem (TSP). This likely involves the coordinated replay of place-cell into “snippets” between
42 successive trials. We hypothesize that “snippets” can be used by the prefrontal cortex (PFC) to
43 implement a form of reward-modulated reinforcement learning. Our simulation experiments provide
44 neurophysiological explanations for two pertinent observations related to navigation. Reward

20 **Abstract (242 words)**

21 As rats learn to search for multiple sources of food or water in a complex environment, they generate
22 increasingly efficient trajectories between reward sites, across multiple trials. This is referred to as the
23 Traveling Salesrat Problem (TSP) ([de Jong et al \(2011\)](#)). Such spatial navigation involves the replay of
24 hippocampal place-cells during awake states, generating small sequences of spatially related place-cell
25 activity that we call “snippets”. These snippets occur primarily during sharp-wave-ripple (SWR)
26 events. Here we focus on the role of replay during the awake state, as the animal is learning across
27 multiple trials. We hypothesize that these snippets can be used by the PFC to achieve multi-goal
28 spatial sequence learning. We use a biologically motivated computational framework known as
29 ‘reservoir computing’ ([Enel et al 2016](#)), in which large pools of prewired neural elements process
30 information dynamically through reverberations. This PFC model consolidate snippets into larger
31 spatial sequences that may be later recalled by subsets of the original sequences. We developed a
32 model of snippet generation that is modulated by reward, in the forward and reverse directions. This
33 implements a form of spatial credit assignment for reinforcement learning. Our simulation
34 experiments provide neurophysiological explanations for two pertinent observations related to
35 navigation. Reward modulation allows the system to reject non-optimal segments of experienced
36 trajectories, and reverse replay allows the system to “learn” trajectories that is has not physically
37 experienced, both of which significantly contribute to the TSP behavior.

38 **Author Summary (120 words)**

39 As rats search for multiple sources of food in a complex environment, they generate increasingly
40 efficient trajectories between reward sites, across multiple trials, referred to as the Traveling Salesrat
41 Problem (TSP). This likely involves the coordinated replay of place-cell into “snippets” between
42 successive trials. We hypothesize that “snippets” can be used by the prefrontal cortex (PFC) to
43 implement a form of reward-modulated reinforcement learning. Our simulation experiments provide
44 neurophysiological explanations for two pertinent observations related to navigation. Reward

45 modulation allows the system to reject non-optimal segments of experienced trajectories, and reverse
46 replay allows the system to “learn” trajectories that it has not physically experienced, both of which
47 significantly contribute to the TSP behavior.

48 **Introduction (637/650)**

49 Spatial navigation in the rat involves the replay of place-cell subsequences (snippets) during awake
50 and sleep states in the hippocampus during sharp-wave-ripple (SWR) events (Carr et al 2011,
51 Davidson et al 2009, Euston et al 2007, Kudrimoti et al 1999). We focus on the role of replay during
52 the awake state (Karlsson & Frank 2009), as the animal generates increasingly efficient trajectories
53 between reward sites, across multiple trials. This trend toward near-optimal solutions is reminiscent of
54 the classic Traveling Salesperson Problem (TSP) (de Jong et al 2011). While it appears likely that
55 replay contributes to this learning behavior, the underlying neurophysiological mechanisms remain to
56 be understood. We hypothesize (a) that snippet replay allows recurrent dynamics in prefrontal cortex
57 (PFC) to consolidate snippet representations into novel efficient sequences, by rejecting other
58 sequences that are less robustly coded in the input, and (b) that a form of reward-modulated replay in
59 hippocampus implements a simple and efficient form of reinforcement learning to achieve this (Singer
60 & Frank 2009).

61 INSERT Figure 1 HERE

62 An example of the behavior in question is illustrated in Figure 1. Panels A-C illustrate navigation
63 trajectories that contain subsequences of the optimal path (in red), as well as non-optimal
64 subsequences (in blue). Panel A illustrates the assembled optimal path in red. We show that a
65 biologically inspired recurrent network model of prefrontal cortex (Enel et al 2016) is able to integrate
66 snippets from examples of non-optimal trajectories and to synthesize an optimal path.

67 For sequence learning, recurrent networks provide inherent sensitivity to serial and temporal structure.
68 Modification of recurrent connections requires different methods of unwinding the recurrent
69 connections in time, which limits the full dynamics of the recurrent system over extended time. To
70 avoid this temporal cut-off and the space and time complexity required in the calculation of credit

71 assignment to recurrent connections we used the framework of reservoir computing in which input and
72 recurrent connections are fixed, and learning-related plasticity occur outside the reservoir network
73 (Dominey 1995). The readout connections from the recurrent network learn the statistical structure of
74 the data that the system is trained on, which places requirements on the mechanism that trains the
75 model. We test the hypothesis that the structure of snippet replay from the hippocampus will provide
76 the PFC with constraints that can be integrated in order to contribute to solving the TSP problem.

77 Two principal physical and neurophysiological properties of navigation and replay are exploited by
78 the model and contribute to the system's ability to converge onto an acceptable solution to the TSP.
79 First, during navigation between baited food wells in the TSP task, non-optimal trajectories (by
80 definition) cover more distance between rewards than near-optimal ones. Second, during the replay of
81 recently activated places cells, the trajectories are encoded in forward and reverse directions (Diba &
82 Buzsaki 2007, Foster & Wilson 2006). Exploiting these observations, we test the hypotheses that:

- 83 1. Non-optimal trajectories will be less represented in SWR replay because of their increased
84 distance to reward, allowing the PFC to eliminate non-optimal subsequences in constructing
85 the final efficient trajectory.
- 86 2. Reverse SWR replay will allow the model to exploit the information provided by a given
87 sequence in forward and backward directions, whereas the actual trajectory run by the rat has
88 one direction only.

89 In testing these hypotheses, we will illustrate how the system can meet the following challenges:

- 90 1. Learn a global place-cell activation sequence from an unordered set of snippets
- 91 2. Efficiently decode a two dimensional location from a noisy place-cell activation pattern that is
92 output from the PFC reservoir.
- 93 3. Consolidate multiple non-optimal sequences into a trajectory that efficiently links rewarded
94 locations, thus converging to a good solution to the TSP problem.

95 4. Learn a trajectory in forward direction and generate it in forward and backward direction,
96 including concatenating parts of both forward and reverse replayed snippets in order to
97 generate novel trajectories as demonstrated in (Gupta et al 2010).

98 The objective is to provide a coherent explanation of how critical aspects of replay – notably its
99 modulation by reward, and the forward and reverse aspects, can be exploited by a cortical sequence
100 learning system in order to display novel and efficient navigation trajectory generation.

101 **Material & methods**

102 The model developed in this research provides a possible explanation of mechanisms that allow PFC
103 and hippocampus to interact to perform path optimization. This implies functional connectivity
104 between these two structures. In a recent review of hippocampal–prefrontal interactions in memory-
105 guided behavior Shin and Jadhav (2016) outlined a diverse set of direct and indirect connections that
106 allow bi-directional interaction between these structures. Principal direct connections to PFC originate
107 in the ventral and intermediate CA1 regions of the hippocampus (Cenquizca & Swanson 2007,
108 Harland et al 2018). direct connections between hippocampus and PFC pass via the medial temporal
109 lobe (subiculum, entorhinal cortex, peri- and post-rhinal cortex) (Delatour & Witter 2002), and the
110 nucleus reuniens (Vertes et al 2007). Indeed memory replay is observed to be coordinated across
111 hippocampus and multiple cortical areas including V1 (Ji & Wilson 2007). These studies allow us to
112 consider that there are anatomical pathways that justify the modeling of bi-directional interaction
113 between PFC and hippocampus (McClelland et al 1995).

114 Our approach is to start with a set of navigation trajectories (observed from rat behavior, or generated
115 automatically) that represent the recent experience from the simulated rat. Snippets are extracted from
116 this experience, and used to train the output connections of the PFC reservoir. This requires the
117 specification of a model of place-cell activation in order to generate SWR snippets. Based on this
118 training, the sequence generation performance is evaluated to test the hypotheses specified. The
119 evaluation requires a method for comparing sequences generated with expected sequences that is
120 based on the Fréchet distance.

121

122 **Navigation Trajectories**

123 A trajectory is a sequence of N contiguous two-dimensional coordinates sampled from time t_1 to time
124 t_N noted $L(t_1 \rightarrow t_N)$ that corresponds to the rat's traversal of the baited feeders. The Euclidian
125 distance between two successive points defines the spatial resolution. We will use points per meter
126 ($points * m^{-1}$) as the spatial resolution unit and the spatial resolution of trajectories depicted in is 20
127 $points * m^{-1}$ along the trajectory.

128 Experiments were performed using navigation trajectories, including those displayed in Figure 1,
129 based on data recorded from rats as they ran the TSP task ([de Jong et al 2011](#)). Experiments take place
130 in a circular arena having a radius of 151cm. Twenty-one feeders' locations are distributed according
131 to a spiral shape. The feeders' locations and labels are always the same. In a typical configuration,
132 only 5 feeders are baited with a cup containing a 20mg food pellet while the others are empty.
133 Navigation trajectories are sampled from 100 different configurations of baited feeders that were
134 explored by different rats over 10 trials each. Before each trial, the rats were positioned at the same
135 starting point and the trajectory of the animal is recorded by an overhead camera at about 20-30 frames
136 per seconds. Figure 2 panel A depicts the trajectory performed by a rat during trial 4 for configuration
137 84 (17,10,3,14,15).

138 INSERT Figure 2 HERE

139 Video tracking sometimes contain noisy or missing samples and thus we interpolate data with splines.
140 Trajectories are first condensed by eliminating contiguous samples sharing the same coordinates (i.e.
141 static parts of the trajectories are pruned). Then the remaining points of the trajectories are resampled
142 evenly in order to match the target spatial resolution of 20 $points * m^{-1}$, corresponding to 5cm
143 between each position of the trajectory. This can be visualized in Figure 2B.

144 In the acquired dataset of 100 configurations of rat behavior, 64 configurations resulted in the rat
145 finding the optimal paths. In 2/3 (42/64) of these configurations the animal found the optimal
146 trajectory without having previously experienced the component subsequences. In one third (22/64) of

147 the optimal configurations the animal traversed parts of the efficient path in early trials before finally
 148 displaying the efficient trajectory. These data are thus suitable for testing our hypothesis, and are used
 149 in our experiments. For these data we created an idealized trajectories that link optimally the baited
 150 feeders as illustrated in Figure 2, panel B. This is required for establishing a performance criterion to
 151 assess the model.

152 **Place-cells**

153 The rat is modeled as an animat that navigates in a closed space of 2x2 meters (Wilson 1991). The
 154 animat can move freely in all direction within a limited range ($\pm 110^\circ$ left and right of straight ahead),
 155 and encodes locations using hippocampus place-cells activity alone. A given $s = (x, y)$ location is
 156 associated with a place-cell activation pattern by a set of 2D Gaussian place-fields:

$$f_k(s) = e^{-\frac{\|s - c_k\|^2}{w_k}} \tag{1}$$

157
 158 Where:

- 159 • k is the index of the place-cell
- 160 • $f_k(s)$ is the mean firing rate of the k^{th} place-cell
- 161 • c_k is the (x, y) coordinate of the k^{th} place-cell
- 162 • $w_k = \frac{r_k^2}{-\log(\theta)}$ is a constant that will constrain the highest activations of the place-cell to be
 163 mostly contained in a circle of radius r_k , centered in c_k
- 164 • r_k is the radius of the k^{th} place-field
- 165 • θ is the radius threshold which controls the spatial selectivity of the place-cell

166 Parameter w_k is a manner of defining the variance of the 2D Gaussian surface with a distance to center
 167 related parameter r_k .

168 In this study, we will consider a uniform grid of 16x16 Gaussian place-fields of equal size (mimicking
 169 dorsal hippocampus). In Figure 3 the spatial position and extent of the place fields of several place-cells
 170 is represented in panel A by red circles. The degree of red transparency represents the mean firing rate.

171

INSERT Figure 3 HERE

172 A mean firing rate close to one will result in a bright circle if the location s is close to the place-field
173 center c_k of the place-cell k . For a more distant place-field center c_l of place-cell l , the mean firing
174 rate will be less important and the red circle representing this mean firing rate will be dimmer.

175 Thus at each time step, the place-cell coding that corresponds to a particular point in a trajectory is
176 defined as the projection of this $L(t_n)$ point through K radial basis functions (i.e. Gaussian place-fields
177 spatial response)

$$X_{in}(t_n) = \{f_k(L(t_n))\}_{k \in 1 \dots K}$$

178

(2)

179 Each coordinate of the input vector $X_{in}(t_n)$ represents the mean firing rate of hippocampus place-cells
180 and its value lies between 0 and 1. Figure 3 represents in panel B the ABCED trajectory $L(t_1 \rightarrow t_N)$ and
181 the corresponding place-cell mean firing rate raster $X_{in}(t_1 \rightarrow t_N)$ is depicted in panel C

182 **Hippocampus replay**

183 The hippocampus replay observed during SWR complexes in active rest phase is modeled by
184 generating a training set made of condensed (time compressed) subsequences of place-cell activation
185 patterns that are then replayed at random. The distribution that is sampled for drawing a random place-
186 cell activation pattern might be uniform or modulated by new or rewarding experience as described in
187 ([Carr et al 2011](#)) : In particular, we model a random replay based on reward. In ([Ambrose et al 2016](#))
188 the authors show that during SWR sequences place-cell activation occur in reverse order at the end of
189 a run. We will focus on reward modulation and reverse replay.

190 We define a snippet as the concatenation of a pattern of successive place-cell activation:

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

191

(3)

192 Where:

- 193 • s is the number of place-cell activations.

194 In our model of hippocampus replay, we define a time budget noted T that corresponds to the duration
195 of a replay episode (experimentally, typically 70-100 ms). A replay episode E is a set of snippets of
196 length s :

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

197 (4)

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

200 In Figure 3, Panel B represents a particular trajectory through feeders A, B, C, E and D. The depicted
201 snippet is a subsequence of 5 contiguous locations belonging to the ABCED sequence. The B and E
202 feeders are baited and marked as rewarding (R_1 and R_2). Panel B shows the spatial extent of a given
203 snippet chosen in sequence ABCED and panel C shows the place-cell activation pattern of the
204 ABCED trajectory and the corresponding snippet location in the raster.

205 The snippet replay model favors snippets that are on efficient paths linking rewarded sites (e.g. paths
206 linking feeders A, B and C in panel B Figure 3), and not those that are on inefficient paths (as in paths
207 linking C, D and E in the same panel). This is achieved by propagating reward value backwards from
208 rewarded locations, and calculating the probability of replay as a function of proximity to a reward.
209 Panel D of Figure 3 illustrates the resulting probability distributions for snippet selection along the
210 complete path. Panel E represents the spatial extent of snippet replay likelihood. Note that the paths
211 linking A, B and C have the highest probabilities for snippet replay.

212 Hippocampus place-cell replay can occur in forward or backward direction as suggested in (Foster &
213 Wilson 2006). We model the reverse replay as follows: For a given trajectory k of N_k samples, there
214 are $N_k - s$ possible snippets that can be replayed but only a limited number of snippets will be
215 selected to fit the time budget T . A snippet $S(n)$ has a likelihood of being replayed if it is related to a
216 reward prediction. A generative model of snippet replay likelihood is first learnt by propagating a time
217 delayed reward information according to the replay direction and the snippet duration. The timespan of
218 a snippet acts as a propagation vector during the estimation phase of the snippet replay likelihood.

219 The reward prediction $V(t_1 \rightarrow t_{N_k})$ is learnt by initializing it to small positive random values and
 220 then iteratively refined by applying the procedure (8) K times:

221 1. Draw a random contiguous time index subset $\tau \equiv T(n, s, r; \beta_{learn})$ according to the reverse rate
 222 β_{learn} :

223 a. Select a time-step t_n such that $n \in \{1 \dots N_k\}$ according to the replay likelihood defined by:

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

224 (5)

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

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

227 (6)

228 2. Update the reward estimate V over increasing indices of τ by computing the update equation:

$$V(\tau_k) = \alpha(R(\tau_{k-1}) + \gamma V(\tau_{k-1})) + (1 - \alpha)V(\tau_k)$$

229 (7)

230 Where:

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

234 (8)

235 It is a convex combination of the current estimate of the reward information $V(\tau_k)$ at the next time
 236 step and the instantaneous reward information $R(\tau_{k-1}) + \gamma V(\tau_{k-1})$ based on the previously observed
 237 reward signal $R(\tau_{k-1})$ and delayed previous reward estimate $\gamma V(\tau_{k-1})$. Equation 7 implements a
 238 form of temporal difference learning. It is sufficient to define a coarse reward signal as:

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

239 The snippet generation procedure is simply the repetition of the steps a and b of procedure (8) with
 240 $\beta_{generate}$ used instead of β_{learn} until the sum of time subsequences durations overflows the fixed
 241 budget duration. These snippets will serve as inputs to the reservoir model of PFC described next.

242 **Reservoir model of PFC for snippet consolidation**

243 We model the prefrontal cortex as a recurrent reservoir network. Reservoir computing refers to a class
 244 of recurrent network models with fixed recurrent connections. The reservoir units are driven by
 245 external inputs and the network dynamics provides a high dimensional representation of the inputs
 246 from which the desired outputs can then be read out by a trained linear combination of the reservoir
 247 unit activities. The principle has been co-developed in distinct contexts as the temporal recurrent
 248 network ([Dominey 1995](#)), the liquid state machine ([Maass et al 2002](#)), and the echo state network
 249 ([Jaeger 2001](#)). The version that we use to model the frontal cortex employs leaky integrator neurons in
 250 the recurrent network. At each time-step, the network is updated according to the following schema:

251 INSERT Figure 4 HERE

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

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

255 (9)

256 Where:

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

258 Synaptic weights are randomly selected at the beginning of the simulation. Various probability density
 259 functions (PDF) could be sampled and one condition on W_{ffwd} is its bijectivity: Every stimulus
 260 $X_{in}(t_n)$ must have a distinct image through W_{ffwd} and each U_{ffwd} must correspond to a unique

261 $X_{in}(t_n)$. Practically speaking (Lukosevicius 2012), sampling $U[-1, 1]$ a uniform distribution is
 262 sufficient. A positive synaptic weight in a connectivity matrix models an excitatory connection and a
 263 negative weight models an inhibitory connection between two neurons (that could be implemented via
 264 an intervening inhibitory interneuron). A synaptic weight equal to zero models no connection between
 265 two given neurons. A larger absolute value of the synaptic weights represents a reinforced correlation
 266 between firing patterns of those two neurons.

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

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

270 (10)

271 Where:

- 272 • W_{rec} is a N by N square connectivity matrix.

273 Synaptic weights are drawn from a $U[-1, 1]$ uniform distribution, scaled by a $S(N; K) = K \frac{1}{\sqrt{N}}$ factor.

274 The same sign convention as in equation (9) applies for the recurrent connectivity matrix.

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

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

277 (11)

278 The membrane potential of the reservoir's neurons P_{res} then is computed by solving the following
 279 ordinary derivative equation (ODE):

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

280 (12)

281 Where:

- 282 • τ is the neuron's time constant. It models the resistive and capacitive properties of the
283 neuron's membrane.

284 In this article, we will consider a contiguous assembly of neurons that share the same time constant.
285 The inverse of the time constant is called the leak rate and is noted h . By choosing the Euler's forward
286 method for solving equation(12), the membrane potential is computed recursively by the equation:

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

287 (13)

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

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

$$X_{res}(t_n) = \sigma_{res}(P_{res}(t_n); \theta_{res})$$

296 (14)

297 Where:

- 298 • σ_{res} is the non-linear activation function of the reservoir neurons
- 299 • θ_{res} is a bias that will act as a threshold for the neuron's activation function.

300 We choose a $\sigma_{res} \equiv \tanh$ hyperbolic tangent activation function with a zero bias. Negative firing rate
301 values represent the inhibitory/excitatory connection type in conjunction with the sign of the synaptic
302 weight. Only the product of the mean firing rate of the afferent neuron by its associated synaptic
303 weight is viewed by the leaky integrator neuron.

304 At this point, the reservoir's states carry information about its present and past activation values within
 305 a limited time span. One single reservoir neuron does not contain the whole information but its
 306 activation fluctuations carry partial information about the serial and temporal structure of the stimulus
 307 sequence. This property corresponds to mixed selectivity (Rigotti et al 2013).

308 The concatenation of b reservoir states from time t_{n-b+1} to time t_n is noted $X_{res}(t_{n-b+1} \rightarrow t_n)$. If we
 309 attempt to represent the mean firing rate activation pattern of neurons within this time range, we can
 310 observe that the dynamics are rich but it is difficult to read useful information from a raster: Figure 5
 311 shows in panel B the raster of reservoir neurons when exposed to the ABCDE place-cell activation
 312 sequence depicted in Panel A. The corresponding trajectory is showed in Figure 1, panel A.

313 Instead, we propose to compute the Singular Value Decomposition (SVD) of the matrix
 314 $X_{res}(t_{n-b+1} \rightarrow t_n)$. $X_{res}(t_{n-b+1} \rightarrow t_n)$ is viewed as a matrix containing b observations of N
 315 variables (number of neurons in reservoir). The SVD factorizes $X_{res}(t_{n-b+1} \rightarrow t_n)$ into three matrices
 316 named U , Σ , V such that:

$$X_{res}(t_{n-b+1} \rightarrow t_n) = U * \Sigma * V^*$$

(15)

317 We will call the matrix $U * \Sigma$ the "equivalent reservoir". Figure 5.C shows a representation of the neural
 318 network model including the equivalent reservoir.
 319

INSERT Figure 5 HERE

320
 321 Matrices U and V^* might be viewed intuitively as rotation matrices and Σ as a scaling matrix. Since U
 322 and V^* are orthonormal basis, we can see $U * \Sigma = X_{res}(t_{n-b+1} \rightarrow t_n) * V$ as a raster of equivalent
 323 neurons. Each equivalent neuron is a neuron whose activation value is a linear combination of
 324 reservoir neurons. The connectivity matrix that projects X_{res} reservoir neurons into equivalent
 325 reservoir neurons X_{eq} is matrix V .

$$X_{eq}(t_{n-b+1} \rightarrow t_n) = X_{res}(t_{n-b+1} \rightarrow t_n) * V$$

(16)

327 In Figure 5.D, we observe the matrix $X_{eq}(t_{n-b+1} \rightarrow t_n)$ as a raster of equivalent neurons. Most of the
 328 information carried by reservoir neurons can be described with a significantly lower number of
 329 dimensions.

330 This will allow us to interpret reservoir neurons activation patterns as a simple trajectory in a two or
 331 three -dimensional space as depicted in Figure 5.E

332 The activation values of the first three neurons of the equivalent reservoir over time are represented as
 333 a three-dimensional trajectory. Time steps of the trajectory that match with a feeder position is
 334 represented with the feeder associated label (A, B, C, D or E).

335 Moreover, the W'_{ro} synaptic weights that connect the equivalent reservoir to the readout layer are
 336 exactly the correlation coefficient between equivalent reservoir and readout neurons.

$$W_{ro} = V * corr(X_{des}, X_{eq})$$

337 Intuitively, the readout layer selects the linear combinations of neurons correlated with the expected
 338 activation pattern.

339 **Learning in Modifiable PFC Connections to Readout**

340 Based on the rich activity patterns in the reservoir, it is possible to decode the reservoir's state in a
 341 supervised manner in order to produce the desired output as a function of the input sequence. The
 342 expected output is only required to be an activation pattern that is temporally congruent with the input
 343 stimulus. This decoding is provided by the readout layer and the matrix of modifiable synaptic weights
 344 linking the reservoir to the readout layer, noted W_{ro} and represented by dash lines in Figure 4.

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

$$X_{ro}(t_n) = \sigma_{ro}(W_{ro} * X_{res}(t_n); \theta_{ro})$$

346 (17)

347 Where:

- 348 • σ_{ro} is the non-linear activation function of the readout neurons

349 • θ_{ro} is a bias that will act as a threshold for the neuron's activation function

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

351 Notice that the update algorithm described above is a very particular procedure inherited from
352 feedforward neural networks. We chose to use it because it is computationally efficient and
353 deterministic.

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

$$W_{ro}(t_n) = W_{ro}(t_{n-b+1}) + \alpha * X_{res}(t_{n-b+1} \rightarrow t_n) * (X_{ro}(t_{n-b+1} \rightarrow t_n) - X_{des}(t_{n-b+1} \rightarrow t_n)) * (1 - X_{ro}(t_{n-b+1} \rightarrow t_n)^2)$$

357 (18)

358 Where:

359 • α is a small positive constant called the learning rate

360 • $t_{n-b+1} \rightarrow t_n$ is the concatenation of b time steps from t_{n-b+1} to t_n

361 When $b = 1$, equation (18) computes a stochastic gradient descent. The case when $b > 1$ is called a
362 mini-batch gradient descent and allows one to estimate the synaptic weight gradient base on b
363 successive observations of predicted and desired activation values. A mini batch gradient allows one
364 to compute efficiently and robustly the synaptic weight gradient. Empirically, $b = 32$ gives satisfying
365 results.

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

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

367 (19)

368 This readout is considered to take place in the striatum, as part of a cortico-striatal learning system.

369 This is consistent with data indicating that while hippocampus codes future paths, the striatum codes
370 actual location (van der Meer et al 2010).

371 **Training**

372 After each trial, the model is trained using a dataset that is generated online by the snippet replay
373 mechanism described above in the paragraph on Hippocampus replay. The readout synaptic weights
374 are also learned online by using the learning rule described in the Learning in Modifiable PFC
375 Connections to Readout section. The model does not receive any form of feedback from the
376 environment and it learns place-cell activation sequences based only on random replay of snippets.

377 Between each sequence of the training set (snippets in our case), the states of the reservoir and readout
378 are set to a small random uniform value centered on zero. This models the fact that a time between the
379 replays of two snippets is sufficiently long for inducing states in the neural network that are not
380 correlated with the previous stimulus. This is required for having the same effect as simulating a
381 longer time after each snippet but without having to pay the computational cost associated to this extra
382 simulation time.

383 **Embodied Simulation of Sensory-Motor loop via the spatial filter**

384 Once the model is trained, we need to evaluate its performance and the trajectories it can generate. The
385 model is primed with the first p steps of the place-cell activation sequence the model is supposed to
386 produce. This sequence is called the target sequence. Then the model's ability to generate a place-cell
387 activation sequence is evaluated by injecting the output prediction of the next place-cell activation
388 pattern as the input at the next step. In this iterative procedure, the system should autonomously
389 reproduce the trained sequence pattern of place-cell activations.

390 Predicted place-cell activation values might be noisy, and the reinjection of even small amounts of
391 noise in this autonomous generation procedure can lead to divergence. We thus employ a procedure
392 that determines the location coded by the place-cell activation vector output, and reconstructs a proper
393 place-cell activation vector coding this location. We call this denoising procedure the spatial filter as
394 referred to in Figure 4.

395 We model the rat action as 'reaching the most probable nearby location'. Since only the prediction of
396 the next place-cell activation pattern η is available, we need to estimate the most probable point

397 $s^*(t_{n+1}) = (x^*(t_{n+1}), y^*(t_{n+1}))$. From a Bayesian point of view, we need to determine the most
 398 probable next location $s(t_{n+1})$, given the current location $s(t_n)$ and the predicted place-cell activation
 399 pattern $\eta(t_n)$. We can state our problem as:

$$s^*(t_{n+1}) = \operatorname{argmax}_{s(t_{n+1})} P(s(t_{n+1}) | \eta(t_n), s(t_n)) + u$$

400 (20)

401 Where:

- 402 • u is a noise function sampling a uniform distribution $\mathcal{U}(0, m)$

403 u is useful at least in degenerate cases when a zero place-cell activation prediction generates an invalid
 404 location coding. It is also used for biasing the generation procedure and to explore other branches of
 405 the possible trajectories the model can generate as described in section Evaluating Behavior with
 406 Random walk.

407 The animat is then moved to this new location s^* and a new noise/interference free place-cell
 408 activation pattern is generated by the place-field model described in section Sensor model. We call this
 409 place-cell prediction/de-noising method the spatial filter and it emulates a sensory-motor loop for the
 410 navigating rat in this study. Figure 4 depicts this sensory motor loop. An embodied animat able to
 411 choose the most appropriate action according to a sequence of stimuli should be able to perform the
 412 same experiments. We will study trajectory consolidation with the spatial filter.

413 By using Bayes rule, we can write:

$$P(s(t_{n+1}) | \eta(t_n), s(t_{n-1})) = \frac{P(\eta(t_n) | s(t_{n+1})) P(s(t_{n+1}) | s(t_n))}{P(\eta(t_{n+1}) | s(t_n))}$$

414 (21)

415 Where:

- 416 • $P(\eta(t_n) | s(t_{n+1}))$ is called the sensor model and represents the probability to fire the pattern η
 417 at time t_n , given the point s at time t_{n+1}

- 418 • $P(s(t_{n+1})|s(t_n))$ is called the transition model and represents the probability for the agent to
 419 be at location $s(t_{n+1})$ given the current location $s(t_n)$
- 420 • $P(\eta(t_n)|s(t_n))$ is a normalization factor that does not depend on $s(t_{n+1})$ and could be omitted
 421 for a maximization problem as stated in equation (20)

422 The sensor and transition model need to be defined in order to evaluate the most probable point given
 423 the predicted place-cell activation vector and the current position.

424 **Sensor model**

425 If we consider the place-cell activations $\eta = (\eta_k)_{k \in 1 \dots K}$ at time t as a conditionally independent
 426 consequence and the site $s = (x, y)$ as the cause that generated η , the sensor model can be written as:

$$P(\eta|s) = \prod_{k=1}^K P(\eta_k|s)$$

427 (22)

428 We propose to estimate the posterior probability $P(\eta|s)$ by using a similarity function based on a one-
 429 dimensional Gaussian function for mapping any difference between mean firing rates $f_k(s)$ and η_k
 430 into a real number between 0 and 1. The Gaussian based function is described by the equation:

$$g(x, x'; \sigma) = e^{-\frac{(x-x')^2}{2\sigma^2}}$$

431 (23)

432 Where:

- 433 • σ is the standard deviation of the Gaussian function

434 Thus when a mean firing rate $f_k(s)$ is close to η_k , the corresponding response $g(f_k(s), \eta_k; \sigma)$ is close
 435 to 1. A very different mean firing rate will be related with a response close to zero. Parameter σ
 436 controls the selectivity of the kernel. Small σ values will define a narrow Gaussian curve and result in
 437 the rejection of points separated from a relatively close distance from each other. We define the
 438 posterior probability as:

$$P(\eta_k|s) = \frac{g(f_k(s), \eta_k; \sigma)}{\sum_{s' \in S} g(f_k(s'), \eta_k; \sigma)}$$

439 (24)

440 Then it is possible to rewrite equation (22) as:

$$P(\eta_k|s) = \prod_{k=1}^K \frac{e^{-\frac{(f_k(s) - \eta_k)^2}{2\sigma^2}}}{\sum_{s' \in S} e^{-\frac{(f_k(s') - \eta_k)^2}{2\sigma^2}}}$$

441 (25)

442 Since we are using a product of exponential functions, we can simplify equation (22) :

$$P(\eta|s) = e^{-\frac{\|f(s) - \eta\|^2}{2\sigma^2}} \Gamma(\eta)$$

443 (26)

444 Where:

445 • $\Gamma(\eta) = \prod_{k=1}^K \frac{1}{\sum_{s' \in S} e^{-\frac{(f_k(s') - \eta_k)^2}{2\sigma^2}}}$ is a normalization function that only depends on η

446 $\Gamma(\eta)$ might be simplified by considering that $\sum_{s' \in S} e^{-\frac{(f_k(s') - \eta_k)^2}{2\sigma^2}}$ is an approximation of the one
447 dimensional Gaussian integral. The Gaussian integral in K dimensions is given by:

$$\int_{\mathbb{R}^K} e^{-\alpha \|x\|^2} dx = \left(\frac{\pi}{\alpha}\right)^{\frac{K}{2}}$$

448 (27)

449 Where:

- 450 • $x = f_k(s) - \eta_k$
- 451 • $\alpha = \frac{1}{2\sigma^2}$

452 Thus, the one-dimensional Gaussian integral approximation evaluates to a constant

453 $\sum_{s' \in S} e^{-\frac{(f_k(s') - \eta_k)^2}{2\sigma^2}} \approx (2\pi\sigma^2)^{\frac{1}{2}}$ and we have:

$$\Gamma \approx (2\pi\sigma^2)^{-\frac{K}{2}}$$

454 (28)

455 In equation (26), $\Gamma(\eta)$ could be also defined as:

$$\Gamma(\eta) = \frac{1}{\sum_{s' \in S} e^{-\frac{\|f(s') - \eta\|^2}{2\sigma^2}}}$$

456 (29)

457 According to equation (27), the K-dimensional Gaussian integral approximation evaluates to

458 $\sum_{s' \in S} e^{-\frac{\|f(s') - \eta\|^2}{2\sigma^2}} = (2\pi\sigma^2)^{\frac{K}{2}}$ and we still have $\Gamma \approx (2\pi\sigma^2)^{-\frac{K}{2}}$. It is also possible to evaluate $\Gamma(\eta)$ with

459 Taylor series expansions but the major drawback of this solution would be its sensitivity to noise

460 because of the use of high order polynomials. Since we are only going to evaluate all the points of the

461 S 2D space and find the maximum of equation (26), the normalization constant Γ could be omitted.

462 The computation equation (26) is based on a similarity function G which measures the similarity

463 between place-cell activation patterns:

$$G(x, x') = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$$

464 (30)

465

466 It is a K dimensional Gaussian kernel similar to those used in support vector machines ([Cortes &](#)

467 [Vapnik 1995](#)). Intuitively, it is a dot product in feature space. In Figure 6, panel A shows an example of

468 the values of equation (30) computed for $\sigma = 10$ and a finite element map of 1000x1000 elements.

469 The most probable locations emerge as the most salient areas.

470 INSERT Figure 6 HERE

471 Note that the same selection principle could be extended to any place-cell activation spatial response.

472 A two dimensional grid description of a place-cell activation function built on observations gathered

473 during a behavioral experiment with a rat is a good example.

474 **Transition model**

475 The agent can move only on its adjacent locations. This is a constraint of physical continuity that
 476 restricts the areas of the map that must be evaluated for selecting the next location $s(t_{n+1})$. This is
 477 modeled by a circle of radius R centered on the agent's current position $s(t_n)$. The agent is allowed to
 478 move within a circular sector defined by Radius R and angle α defined by:

$$V = \left\{ s(t_n) \in S \mid \|\vec{c}(t_n)\|^2 < R^2 \text{ and } \frac{\langle \vec{c}(t_n), \vec{d}(t_{n-1}) \rangle}{\|\vec{c}(t_n)\| \|\vec{d}(t_{n-1})\|} < \cos\left(\frac{\alpha}{2}\right) \right\}$$

479 (31)

480 Where:

- 481 • $\vec{c}(t_n) = s(t_n) - s(t_{n-1})$ is a vector that represent the 'heading' of the agent, based on its current
 482 position $s(t_n)$ and previous position $s(t_{n-1})$
- 483 • $\vec{d}(t_n) = s(t_{n+1}) - s(t_n)$ is a vector that represent the future heading of the agent, based on the
 484 future position $s(t_{n+1})$ being evaluated and its current position $s(t_n)$

485 If no direction is available (i.e. the agent is no longer moving) or if α is set to zero, only the
 486 $\|\vec{c}(t_n)\|^2 < R^2$ condition is evaluated. R is chosen in order to enclose the maximum range of motion of
 487 the agent. Restricting the search space for the most probable point by using an appropriate radius R
 488 bounds and significantly reduces the time complexity of the implementation. We then define the
 489 transition model as:

$$P(s(t_{n+1})|s(t_n)) = \begin{cases} \frac{1}{|V|} & \text{if } s(t_{n+1}) \in V \\ 0 & \text{otherwise} \end{cases}$$

490 (32)

491 Where $|V|$ represents the area of the V set

492 Once again, we can omit the normalization constant because of the maximization problem we are
 493 trying to solve and the term $\frac{1}{|V|}$ in (32) can be replaced by 1

494 **A unified formulation**

495 In (Zhang et al 1998) several reconstructions methods are evaluated and they can be viewed as the
 496 maximization of a function $\psi(s, \eta)$ defined by the following general equation for K place-cells:

$$\psi(s, \eta) = \sum_{k=1}^K \phi_k(s) * \eta_k + A(s)$$

497 (33)

498 Where η_k is the mean firing rate of place-cell k , $\phi_k(s)$ is a basis function associated with place-cell k
 499 and $A(s)$ is a bias term that is independent of place-cells and reminiscent of a regularization term.

500 The sum over the K place-cells is in fact a dot product and equation (33) can be rewritten as:

$$\Psi(\mathbf{s}, \boldsymbol{\eta}) = \mathbf{K}(\boldsymbol{\phi}(\mathbf{s}), \boldsymbol{\eta}) + \mathbf{A}(\mathbf{s})$$

501 (34)

502 Where $K(x, x')$ is a function that evaluates the similarity between x and x'

503 In (Zhang et al 1998), the similarity function $K(x, x')$ is the usual dot product $\langle x, x' \rangle$.

504 Reconstructing a two dimensional location s^* based on place-cell activation pattern is equivalent to
 505 maximizing the similarity function $K(\phi(s), \eta(t))$ under the constraint represented by $A(s)$. The
 506 reconstruction problem is stated as:

$$s^*(t) = \operatorname{argmax}_{s \in S} (\psi(s, \eta))$$

507 (35)

508 We propose to formulate our reconstruction kernel based/Bayesian method by using this unified
 509 formulation and identifying terms in equation (34) where $\phi(s) \equiv f(s)$ is the firing rate map (i.e. the
 510 spatial response of the place-cells), $K(x, x') \equiv G(x, x')$ is the Gaussian kernel defined in equation
 511 (30), $A(s) \equiv u$ is the noise function mentioned in equation (20), $S \equiv V$ is the set of points lying in a
 512 circle sector centered on the agent as defined in equation (31)

513 **Parameterization of the Spatial Filter**

514 The spatial filter is required for evaluating the model and it is necessary to evaluate its accuracy. We
 515 use a finite element method for simplicity and speed. The two dimensional space that encloses the

516 arena (4m^2) is thus discretized into square tiles of equal size. Depending on the spatial resolution of
517 this uniform grid, trajectories discretized through this grid will exhibit an aliasing effect if the grid size
518 is too coarse. We estimated the absolute discretization error by measuring repeatedly the distance
519 between a random point drawn uniformly in $[-1,1] \times [-1,1]$ and its discretized coordinates.

520 A grid of 1024×1024 provides an acceptable discretization error of less than 1.5 millimeter

521 Place-cell prediction might be noisy and it is useful to evaluate the performances of the spatial filter in
522 noisy conditions. For each possible grid size, an important number of random points in $[-1, 1] \times [-1, 1]$
523 is drawn from a uniform distribution. Then for each point, a uniform noise is added to the mean firing
524 rate patterns of place-cells activations that correspond to random points. The noise magnitude varies
525 from 0 to 1 (equivalent to 100% mean firing rate) and produce different values of signal to noise ratio.

526 The reconstruction error is measured for different values of the σ parameter of the spatial filter
527 (described in section Sensor model) and for different values of signal to noise ratio.

528 Figure 6, panel B shows the reconstruction mean square root error of different place-cell location
529 decoding methods. We can observe that the linear model is quite poor at decoding an approximation of
530 a hippocampus place-cell activation. Even with a 0.03 noise magnitude (SNR = 13.64dB, 3% error in
531 place-cell pattern prediction) added to the mean firing rate of 256 place-cells, the linear decoder
532 commits errors about 10 cm. Our kernel-based decoder provides reconstruction errors about the same
533 order of magnitude as the discretization error when no noise is added. Reconstruction error is still
534 under 1cm with a 10% noise (SNR = -0.33dB). Panel C summarizes the performance of our decoder
535 for different combinations of σ and noise magnitude.

536 It appears that $\sigma = 10$ is the best parameter regardless of the grid size. The spatial filter will now be
537 configured with a 1024×1024 grid and $\sigma = 10$.

538 **Evaluating Behavior with Random walk**

539 Once the model has been trained, it is then primed with place-cell activation inputs corresponding to
540 the first few steps of the trajectory. The readout from the PFC reservoir generates the next place-cell

541 activation pattern in the trajectory, which is then reinjected into the reservoir via the spatial filter, in a
542 closed loop process. This loop evaluation procedure is called autonomous generation. In order to
543 evaluate what the model really learned in a particular experimental condition, several instances of the
544 same model are evaluated multiple times. We call this a random walk procedure. The batch of
545 generated trajectories (usually 1000) are accumulated in a stencil buffer which acts as a two
546 dimensional histogram showing the most frequently generated trajectories. The arena is drawn with its
547 feeders and a vector field is computed from trajectories in order to show the main direction of these
548 trajectories. Trajectories are superimposed and summed, resulting in a two-dimensional histogram
549 representing the space occupied by trajectories. Figure 7 shows an example of random walk trajectories.

550 INSERT Figure 7 HERE

551 In cases where small errors are reinjected into the closed loop system they can be amplified, causing
552 the trajectory to diverge. It is possible to overcome this difficulty by injecting the expected position at
553 each time step instead of the predicted position. The error/distance measurement will quantify the
554 diverging prediction, while allowing the trajectory generation to continue. This method is called the
555 non-autonomous generation and it evaluates only the ability of a model to predict the next place-cell
556 activation pattern, given a sequence of place-cell activation. If this procedure succeeds, then the model
557 has a chance to succeed as well with the autonomous generation procedure but it is not guaranteed.

558 **Comparing produced and ideal sequences using Discrete Fréchet distance**

559 The joint PFC-HIPP model can be evaluated by comparing an expected place-cell firing pattern with
560 its prediction by the readout layer. At each time step, an error metric is computed and then averaged
561 over the duration of the expected neurons firing rate sequence. The simplest measure is the mean
562 square error. This is the error that the learning rule described in equation (18) minimizes.

563 Although the model output is place-cell coding, what is of interest is the corresponding spatial
564 trajectory. A useful measurement in the context of comparing the predicted and actual navigation
565 sequences is the discrete Fréchet distance. It is a measure of similarity between two curves that takes
566 into account the location and ordering of the points along the curve. We use the discrete Fréchet

567 distance applied to polygonal curves as described in (Eiter & Mannila 1994). In (Wylie 2013) the
568 Discrete Fréchet distance F between two curves A and B is defined by:

$$F(A, B) = \min_{\alpha, \beta} \max_{t \in [1, m+n]} [d(A(\alpha(t)), B(\beta(t)))]$$

569 (36)

570 Where $d(.,.)$ is the Euclidean distance, m is the number of steps of the curve A , n is the number of
571 steps of the curve B , and α, β are reparametrizations of the curves A and B .

572 Intuitively, the Fréchet distance is the maximum distance between matched points of curves A and B .

573 Figure 8 illustrates the discrete Fréchet distance.

574 INSERT Figure 8 HERE

575 The closest points of curves A and B are matched by finding the sequence of steps along A and B (i.e.
576 coupling) that must be followed to achieve the minimum coupling distance. The free space diagram
577 depicted in panel A represents the propagation of the reachability information between two points of A
578 and B .

579 A point in the coupling sequence associates two steps of sequence A and B whose Euclidian distance
580 minimize the coupling distance.

581 In equation (36), the $max(.,.)$ function is used in order to measure the coupling distance between A
582 and B . A particular case happens when A and B are similar, excepted at unique step t . The distance
583 between curves at this step will be reported as the Fréchet distance and we prefer to avoid masking
584 interesting results by using a variant of the Fréchet distance. Instead of $max(.,.)$, we use the $sum(.,.)$
585 function (i.e. operator $+$). The implied coupling distance gives the minimum of the total distance of an
586 order preserving correspondence between points of P and Q (Eiter & Mannila 1994).

587 Results

588 For robustness purposes, results are based on a population of neural networks rather than a single
589 instance. The population size is usually 1000 for evaluating a condition and the metrics described

590 above are aggregated by computing their mean $\mu(\cdot)$ and standard-deviation $\sigma(\cdot)$. For convenience, we
591 define a custom score function associated to a batch of coherent measurements as:

$$score(X) = \mu(X) + \sigma(X)$$

592 (37)

593 Results having a low mean and standard deviation will be reported as low score whilst other possible
594 configurations will result in a higher score. We choose this method rather than Z-score, which
595 penalizes low standard deviations.

596 **Sequence learning**

597 The model performance depends on several parameters. In (Lukosevicius 2012), a guideline for
598 manual parameters tuning is described. Tuning parameters like input scaling requires experience and
599 intuitive insight, and trial and error is suggested. Trial and error is also suggested for tuning the leak
600 rate parameter and the spectral radius of the recurrent weights matrix should be tuned according to the
601 mnemonic abilities required by the task. We prefer to rely on automated parameter selection and we use a
602 clustered simulated annealing algorithm as described in (Ram et al 1996). A solution is thus a
603 particular set of parameters for the model. The possible values of those parameters are:

- 604 • W_{in} scaling factor having values from 0 to 10 with 0.001 increments and noted *in_scale*
- 605 • W_{rec} scaling factor having values from 0 to 10 with 0.001 increments and noted *rec_scale*
- 606 • *leak rate* having values from 0 to 1 with 0.001 increments and noted *leak_rate*

607 In section Reservoir model of PFC for snippet consolidation, we defined $S(N;K) = K \frac{1}{\sqrt{N}}$ as the
608 scaling factor of W_{rec} with N being the number of neurons in the reservoir because the spectral radius
609 $\rho(W_{rec})$ is approximately the same for different values of N . In the parameter search, we are looking
610 for the optimal value of *rec_scale* such that $S(N;rec_scale) = rec_scale \frac{1}{\sqrt{N}}$

611 This parameter set is evaluated on $B = 10$ different model instances, and Fréchet distance
612 measurements are reduced into a one-dimensional cost function (described in equation 37) that is
613 associated to the parameter set.

614 A population of $C = 10$ solutions is evaluated during $I = 1000$ iterations. The initial temperature is
615 $T_0 = 1$ and the final temperature $T_I = 1 * 10^{-20}$. The cooling schedule is exponential multiplicative
616 ($T_k = T_0 * \alpha^k$) and $\alpha = (\frac{T_I}{T_0})^{\frac{1}{I}}$. Every $K = 100$ iterations, the population is renewed with mutations of
617 the best solution evaluated so far. The number of reservoirs neurons is set to $N = 1024$. The noise
618 injected in the location probability map (see paragraph Embodied Simulation of Sensory-Motor loop
619 via the spatial filter) is set to $m = 0$ in order to ensure that the next location selected during the
620 sequence generation is the most probable location. Each model instance is trained with a slow learning
621 rate of $\alpha = \frac{0.1}{N} = 9.7656e - 05$ and by exposing only 100 times to the same sequence. Trajectories
622 are preprocessed as described in section Navigation Trajectories and a spatial resolution of 20
623 *dots * m⁻¹* is used.

624 As a preliminary experiment, we propose to search for a parameter set that allows the model to learn
625 and generate a complex navigation sequence. A sequence is qualified as complex sequences because it
626 contains loop, a feature that is particularly challenging for a sequence learning model. Figure 7 panel A
627 illustrates a complex trajectory featuring 2 loops. It corresponds to a navigation task requiring good
628 mnemonic abilities and from a simulation point of view, sequences with long loops require a context, or
629 state history, long enough to disambiguate repeating subsequences. The goal is to find a parameter set
630 that allows learning and generating complex sequences even in degraded conditions. Table 1
631 summarizes the results of 10 parallel simulated annealing procedures.

632 **INSERT Table 1 HERE**

633 The best solution has been evaluated with a score of 1.1879078. It means that the sum of coupling
634 distances of the generated sequences is low for every model instance evaluated. Figure 7 panel B shows
635 the trajectories generated by the 10000 instances of model setup with the best parameters. They
636 correspond visually to the expected trajectory despite the fact that noise has been added for an easier
637 visualization. We propose to estimate the position prediction error as an average coupling distance
638 between generated and target Trajectory by computing for the 10000 model instances the discrete
639 Fréchet distance score (sum of coupling distances) and to divide it by the number of samples in the

640 target trajectory. It results in a set of averages of distances to the target trajectory. After exposing the
641 model only 100 times to the target sequence, it is able to generate it with a prediction error of 1.16cm
642 ± 2.4 mm. A longer exposure of 1000 times to the target sequence lowers the prediction error to
643 1.13cm ± 2 mm and. There is no significant gain at exposing the model 10 times longer to the training
644 set.

645 The discretization error induced by the use of a finite element method for location reconstruction with
646 our place-cell decoding bayesian method (see section Embodied Simulation of Sensory-Motor loop via
647 the spatial filter) is 1.5mm. It is the lower boundary of the prediction error it is possible to reach. The
648 reconstruction error of our Bayesian method with a noise free signal is 0.32mm ± 0.22 mm

649 This demonstrates that the overall model with only 1024 reservoir neurons is suitable for trajectory
650 learning and the best parameter set is $in_scale = 6.379720$, $rec_scale = 6.553708$, $leak_rate =$
651 0.494997 . The optimized $rec_scale = 6.553708$ value results in a spectral radius $\rho(W_{rec}) \approx$
652 3.8712 and $in_scale = 6.379720$ ensures a significant influence of the input on the reservoir
653 dynamics. The inverse of the model time constant, $leak_rate = 0.494997$, is adapted to the dynamics
654 of the complex training sequence sampled at a spatial resolution of $20\ dots * m^{-1}$.

655 By keeping the same parameters and trajectory and then varying the $leak_rate$ from 0 to 1 and the
656 spatial resolution of the trajectory, it is possible to show that there exists for each spatial resolution a
657 tight range of leak rates that allows good performance. Figure 9 illustrates this property of recurrent
658 neural networks.

659 INSERT Figure 9 HERE

660 The score of each ($leak\ rate, spatial\ resolution$) pair is depicted on a heat map with a logarithmic
661 scale. The x-axis represents the spatial resolution in $dots * m^{-1}$ and the y axis the leak rate
662 $leakrate = 0$ results in a systematic failure and this is expected. Reservoirs neurons are always silent
663 and their states cannot be associated to an expected readout value. Depending on the spatial resolution
664 used for sampling the trajectories, there is an adapted leak rate range that yields good performance.

665 **Consolidation from snippet replay**

666 The model is able to learn and generate navigation sequences from place-cell activation patterns. The
667 next step is to demonstrate that a sequence can be learned by the same model when it is trained on
668 randomly presented snippets, instead of the continuous sequence.

669 INSERT Figure 10 HERE

670 The model parameters are kept the same and snippets are replayed in a random order. In this
671 experiment, no reward is used, and thus each snippet has equal chance of being replayed. The only
672 free parameter is the snippet size. We collect the state-trajectories of reservoir neurons when exposed
673 to snippets. The internal state of the reservoir is driven by the external inputs, and by recurrent internal
674 dynamics. The reservoir adopts a dynamical state-trajectory when presented with an input sequence.
675 Such a trajectory is visualized in Figure 10D. This is a 2D (low dimensional) visualization, via PCA,
676 of the high dimensional state transitions realized by the 1000 neurons reservoir as the input sequence
677 corresponding to ABCDE is presented. Panels A-C illustrate the trajectories that the reservoir state
678 traverses as it is exposed to an increasing number of randomly selected snippets generated for the
679 same ABCDE sequence. We observe that as snippets are presented, the corresponding reservoir state-
680 trajectories start roughly from the same point because of the random initial state of the reservoir before
681 each snippet is replayed. Then the trajectories evolve and partially overlap with the state-trajectory
682 produced by the complete sequence. Replaying snippets at random thus has no negative impact
683 because the reservoir states overlap when snippet trajectories overlap. Snippets quickly drive the
684 reservoir state from an initial random activation (corresponding to the grey area at the center of each
685 panel) onto their corresponding locations in the reservoir activation state-trajectory. Thus, a snippet
686 that encodes the sequence near element B will drive the reservoir into the same state of activation that
687 it did when it appeared in the un-fragmented initial sequence (as in panel D).

688 The stochastic gradient descent can occur because the activation values of X_{res} are reliable and
689 everything happens as if the model had been exposed to the whole sequence. Given this visualization
690 of the reservoir state induced by snippet replay, we now demonstrate that sequence learning is possible

691 through snippet replay by running several simulations using data from rat traversals of different
692 configurations from the behavioral dataset.

693 INSERT Figure 11 HERE

694 Figure 11 illustrates the effects of snippet size on sequence reconstruction performance. In general,
695 longer snippets lead to better learning, though it is of interest to note the variability between
696 performance on these 10 sequences of varying length and complexity from the rat behavior dataset.
697 Different snippet sizes are possible and they are defined by the number of samples corresponding to a
698 percentage $p \in]0, 1]$ of the total number of samples T_k of a k trajectory. Several snippets sizes are
699 evaluated with the optimal parameters found in Sequence learning. Figure 11 displays the Fréchet
700 distance scores when the model learns ten sequences, based on random replay with different snippets
701 sizes. The time budget parameter is fixed to $T = 184 * 100$, 100 times the size of the trajectory
702 having the highest number of samples.

703 The maximum snippet size corresponds to the length of the shortest sequence itself. It is the particular
704 case where the number of possible snippets is one. We are more interested by smaller snippet sizes
705 that only represent partially the place-cells activations of the entire sequence because they model
706 hippocampus place-cell replay during SWR.

707 The snippet size limits the size of the place-cell activation history (context) maintained by the
708 reservoir. If the snippet size is shorter than the size of the context required for generating a trajectory,
709 then the trajectory generated by the model will be different from the expected trajectory. This is not
710 necessarily a bad property of snippet replay since it allows the model to generate novel trajectories that
711 it can represent. We will continue to illustrate this property by examining other generated sequences
712 extracted from the training set.

713 Fréchet distance scores of generated trajectories are clustered into two classes of trajectories by a
714 standard K-means algorithm: the ‘best’ class aggregates trajectories having the smallest possible
715 Fréchet distance score, and the ‘others’ class aggregates trajectories having the remaining scores.

716 Table 2 shows for the trajectory found in configuration 34 and a limited range of snippets size, the
717 population percentage of each class and its associated Fréchet distance score.

718 INSERT Table 2 HERE

719 For *snippet size* = 9, 91.2% of the generated trajectories have a Fréchet distance score of 0.037858.
720 This is the smallest possible snippet size that allows acceptable generation performances for the
721 trajectory found in Configuration 34, trial 2. *snippet size* = 20 is also an interesting point because
722 even the ‘others’ class score has the same order of magnitude as the ‘best’ class score.

723 Sequences are globally generated with the best accuracy the model can reach for *snippet size* = 10.
724 Snippet replay entails a performance drop compared to standard sequence learning paradigms but it is
725 not relevant for navigation tasks. We observe that random snippet replay allows the model to align
726 random subsequences of a trajectory and consolidate them; to break the long range dependencies by
727 using a short snippet size and to generate any trajectory the model can represent. Now we will see the
728 advantages of this capability.

729 **Longer paths are rejected**

730 The previous experiment showed that it is possible to consolidate single navigation sequences from a
731 uniform random replay of snippets. Now we will examine how using reward proximity to modulate
732 snippet replay probability distributions (as described in the hippocampal replay description) can lead
733 to the rejection of longer, inefficient paths between rewarded targets.

734 1000 copies of the model run 10 times. Each are exposed to the reward modulated replay of two
735 sequences ABC and ABD having a common prefix AB as illustrated in Figure 12. At the end of the
736 common prefix AB, the trajectory forks and takes two different directions. This situation is ambiguous
737 because the model has no prior knowledge about which location is the next one. The model is exposed
738 to a random replay of sequences ABC and ABE. The random replay is not uniform and takes into
739 account the reward associated with a baited feeder when food was consumed. Snippets close to a
740 reward have more chance to be replayed and thus to be consolidated into a trajectory.

741

INSERT Figure 12 HERE

742 Panel A in Figure 12 illustrates the distribution of snippets selected from the two sequences, ABC in
743 pink and ABD in blue. At the crucial point of choice at location B, we the distribution of snippets for
744 sequence ABC largely outnumbers those for sequence ABD. This is due to the propagation of rewards
745 respectively from points C and D. Per design, rewards propagated from a more proximal location will
746 have a greater influence on snippet generation. Panel C shows the 2D histogram of autonomously
747 generated sequences when the model is primed with the initial sequence prefix starting at point A. We
748 observe a complete preference for the shorter sequence ABC illustrated in panel E.

749 The snippet generation model described above takes into account the location of rewards, and the
750 magnitude of rewards. In panel B, we show the distribution of snippets allocated to paths ABC and
751 ABD when a 10x stronger reward is presented at location D. This strong reward dominates the snippet
752 generation and produces a distribution that strongly favors the trajectory towards location D, despite
753 its farther distance. Panel F confirms this tendency. This suggests an interesting interaction between
754 distance and reward magnitude. For both conditions, distances to the expected sequence have been
755 measured for every trajectory generated (10 000 for ABC and 10 000 for ABD). Then a Kruskal
756 Wallis test confirms with a result of zero ($p\text{-value} = 0$) for both cases that trajectories generated
757 autonomously correspond to the expected trajectories

758 **Novel efficient sequence creation**

759 Based on the previously demonstrated dynamic properties, we determined that when rewards of equal
760 magnitudes are used, the model would favor shorter trajectories between rewards. We will now test the
761 model's ability to exploit this capability, in order to generate a novel and efficient trajectory from
762 trajectories that contain a sub-path of the efficient trajectory. That is, we determine whether the model
763 can assemble the efficient subsequences together, and reject the longer inefficient subsequences in
764 order to generate a globally efficient trajectory. Figure 1 illustrates the desired trajectory that should be
765 created without direct experience (Panel A), and panels B-D illustrate the three trajectories that each
766 contain part of the optimal trajectory, and which will be used to train the model (red).

767

INSERT Figure 13 HERE

768 The same A, B, C, D, E feeders are used, the efficient sequence is ABCDE, the model parameters are
769 the same and the random replay is based on the following trajectories: ABCED that contains the ABC
770 part of the ABCDE target sequence, BACDE that contains the CDE part of the ABCDE target
771 sequence, and EBCDA that contains the BCD part of the ABCDE target sequence.

772 Figure 13 A illustrates how the hippocampal replay model generates distributions of snippets that
773 significantly favor the representation of the efficient subsequences of each of the three training
774 sequences. This is revealed as the three successive peaks of snippet distributions on the time histogram
775 for the blue (ABCED) sequence, favoring its initial part ABC, the yellow (EBCDA) sequence,
776 favoring its middle part BCD, and the pink (BACDE) sequence, favoring its final part CDE. When
777 observing each of the three color-coded snippet distributions corresponding to each of the three
778 sequences we see that each sequence is favored (with high replay density) precisely where it is most
779 efficient. Thus, based on this distribution of snippets for training, the reservoir should be able to
780 extract the efficient sequence.

781 This is shown in panel B, which illustrates the autonomously generated sequences for 1000 instances
782 of the model executed 10 times each. The spatial histogram reveals that the model is able to extract
783 and concatenate the efficient subsequences to create the optimal path, though it was never seen in its
784 entirety in the input. Panel C illustrate the significant differences between distances to different
785 sequences used in this experiment. A Kruskal-Wallis test confirms these significant differences with a
786 very small p-value = 5.9605e-08.

787

Reverse replay

788 In (Carr et al 2011), hippocampus replay during SWR is characterized by the activation order of the
789 place-cells which occurs in backward and forward direction. We hypothesize that reverse replay
790 allows the rat to consolidate in both directions a trajectory explored in one direction only. If we
791 consider the TSP problem as a graph optimization problem, the effect of reverse replay could be
792 intuitively understood as maintaining the possible trajectories graph as undirected, rather than directed

793 with a snippet replay in forward direction only. This means that an actual trajectory, and its
794 unexplored reverse version, can equally contribute to new behavior, and fewer actual trajectories are
795 required for gathering information for solving the TSP problem.

796 In the current experiment, we examine the effects of forward and reverse replay during the sequence
797 learning process. At the same time, we investigate the effects of forward and reverse reward
798 propagation during the creation of the hippocampal replay model. We first validate that a random
799 replay containing reversed snippets allows the model to generate sequences in both directions. One
800 hundred instances of the model are exposed to a random replay of snippets having size 10 during a
801 reverse replay stage that is evaluated over the following parameters ranges: Reverse rate varying from
802 0 to 1 with increments of 0.1.

803 We examine the train-test direction pairs forward-forward, forward-reverse, reverse-forward and
804 reverse-reverse as illustrated in Figure 14, as we vary the proportion of reverse replay. For each
805 corresponding condition (Cartesian product of hippocampal reward propagation reverse rate, and
806 replay direction reverse rate), the model is successively reset to its initial state and trained.

807 INSERT Figure 14 HERE

808 As shown in Figure 14, the model is able to generate the ABCDE sequence in forward and backward
809 direction and best performances are achieved when about half of the snippets of ABCDE are entirely
810 reversely replayed. The same conclusion applies for sequence EDCBA. At this point, the effect of
811 reverse replay is to have virtually a forward random replay of a given trajectory in forward direction
812 (as experienced by the agent) and the same trajectory experienced in the reverse direction.

813 We now investigate how reverse replay can be exploited in a recombination task where some
814 sequences are experienced in the forward direction, and others in the reverse direction, with respect to
815 the order of the sequence to be generated. We use the same setup as described above for novel
816 sequence generation, but we invert the direction of sequence EBCDA in the training set. Without
817 EBCDA, the model is not exposed to sub trajectories linking feeders B to C and C to D and the
818 recombination cannot occur. We then introduce a partial reverse replay, which allows snippets to be

819 played in forward and reverse order. This allows the reservoir to access segments BC and CD (even
820 though they are not present in the forward version of the experienced trajectory.

821 INSERT Figure 15 HERE

822 Figure 15 illustrates the histogram of sequence performance for 10000 runs of the model (1000 models
823 run 10 times each) on this novel sequence generation task with and without 50% reverse replay. We
824 observe a significant shift towards reduced errors (i.e. towards the left) in the presence of reverse
825 replay (panels A and B).

826 We then examine a more realistic situation by reproducing an observation of spontaneous creation of
827 “shortcuts” described in (Gupta et al 2010). The model is exposed to a random replay of snippets
828 extracted from two trajectories having different direction (clockwise and counter clockwise). Figure 16
829 shows the left lap and right lap trajectories. The system thus experiences different parts of the maze in
830 different directions. We examine whether the use of reverse replay can allow the system to generate
831 novel shortcuts.

832 INSERT Figure 16 HERE

833 The left and right trajectories used for training are illustrated in Figure 16A and B. The model is
834 trained with snippets from these sequences using different random replay rates. The model is
835 evaluated in non-autonomous mode with 4 sequences representing the 4 possible types of shortcut.
836 Figure 16 C shows that the best performance is achieved with an equal proportion of forward and
837 reverse replay. Figure 16D shows trajectories generated in non-autonomous mode with parameters
838 $\alpha = 220^\circ$, $R = 10\text{cm}$, Reverse Replay rate 0%, with a prime starting at the central start point. After
839 turning left, the system continues on the trajectory. Then, crucially, at the bottom of the maze, the
840 system crosses into the right hand maze, and begins to display large errors, as this part of the maze has
841 never been experienced in this direction. In Figure 16E, this same segment is now experienced with
842 low error, even though this movement has never actually been experienced by the system. This is
843 because in this case the system has been trained with Reverse Replay rate 50%. Thus, in the right hand

844 part of the maze, it is as if the system had experienced this already in the counter-clockwise direction,
845 though in reality this has never occurred, but is simulated by the reverse replay.

846 The trajectory is evaluated in non-autonomous mode and the position of the agent necessarily follows
847 the target trajectory. In this case, the expected trajectory describes a counterclockwise path and the
848 right part of the trajectory describes a clockwise path. Results are not significantly different with a
849 replay rate 25% and 75%, where the best performance is observed, and all the other conditions are
850 significantly different. Table 3 Shows the significant p-values for all possible comparisons between
851 two different replay reverse rate.

852 **INSERT Table 3 HERE**

853 This phenomenon was obtained for the 4 possible shortcuts.

854 **Effects of Consolidation and Reverse Replay**

855 We demonstrated the ability of the system to consolidate experience and generate new efficient
856 sequences, and we also showed the added value of reverse replay. The model demonstrates the ability
857 to accumulate and consolidate paths over multiple trials, and to exploit reverse replay. Here we
858 examine these effects on the more extensive and variable dataset extracted from rat behavior ([de Jong](#)
859 [et al 2011](#)). We show the positive effects of replay on trajectories from rats trying to optimize spatial
860 navigation in the TSP task.

861 We consider a trajectory as a graph and the vertices are the feeders and the edges the paths that link
862 the feeders. By analyzing the trajectories, it is possible to split the remaining dataset in 9 different
863 groups characterized by two criteria: The number of informative trials required to observe all the edges
864 contained in the target trajectory, and the direction of the edges contained in the trials taken into
865 account

866 **INSERT Table 4 HERE**

867 In, each column contains configurations in which the rat performed trajectories containing edges of the
868 target trajectory in a particular direction, and in which the rat converged on a near-optimal trajectory at

869 least once. Each configuration consists of 8 to 13 trials where each trial might contain a trajectory
870 having a shared edge with the target trajectory. Such a trial is called an informative trial. Therefore, a
871 given configuration can be classified into different clusters. For example, configuration 97 has three
872 trials that contain the edges of the optimal sequence in the forward direction and three trials containing
873 the edges of the optimal sequence in the forward and backward directions. Configurations 80 and 58
874 contain three trials in which trajectories have edges in the backward direction only.

875 We study the different clusters by training the model with random replay of different combinations of
876 informative trials. The combination representing an empty set of informative trials is not evaluated.
877 For each combination of informative trials, the random replay is evaluated with 0%, 25%, 50%, 75%
878 and 100% of reverse replay rate in order to show the joint effect of random replay and combination of
879 informative trials. The model is then evaluated in non-autonomous mode with the target sequences
880 that consist in a set of trajectories linking the baited feeders in the correct order. An idealized sequence
881 is added to the target sequence set because trajectories generated by the rat might contain edges that do
882 not relate the shortest distance between two vertices. Agent's moves are restricted to a circle having a
883 10 cm radius.

884  INSERT Figure 17 HERE

885 Figure 17 illustrates the combined effects of successive integration of experience and its contribution
886 to reducing error, and of the presence of different mixtures of forward and reverse replay. The
887 ANOVA revealed that there is a significant effect for consolidation ($F(4, 58500) = 333.96, p \leq$
888 0.0001), as performance increases with exposure to more previous experience (Panel A). There is also
889 a significant effect for reverse replay rate ($F(2, 58500) = 2921.54, p < 0.0001$), illustrated in Panel B.
890 There was a significant interaction between consolidation and replay direction ($F(8, 58500) = 2.81, p <$
891 0.01).

892 **Discussion**

893 In this study, we tested the hypothesis that hippocampus awake replay observed during sharp wave
894 ripple events plays a role in memory consolidation by exposing the prefrontal cortex between

895 successive trials to short subsequences of place-cell activation patterns. The behavior of interest is a
896 form of spatial navigation trajectory optimization in a task, mimicking the well-known traveling
897 salesperson problem. It is a NP-Hard problem and finding an exact solution would require significant
898 time and computing resources. Nevertheless, it has been observed that a rat was able to quickly solve
899 simplified versions of this problem ([Bureš et al 1992](#), [de Jong et al 2011](#)).

900 In this behavior, rats are observed to converge quickly to a near-optimal path linking 5 baited food
901 wells in a 151cm radius open arena. During their successive approximation to the optimal path, the
902 rats often traversed segments of the optimal trajectory, as well as non-optimal segments. Observing
903 this behavior, we conjectured the existence of neural mechanisms that would allow most of the
904 optimal segments to be reinforced and most of the non-optical segments to be rejected, thus leading to
905 the production of the overall near-optimal trajectory. The overall mechanism we propose can be
906 decomposed into two distinct neural systems. The first is a replay mechanism that favors the
907 representation of snippets that occurred on these optimal segments, and that in contrast will give
908 reduced representation to snippets that correspond to non-optimal trajectory segments. Interestingly,
909 this characterization of replay is broadly consistent with the effects of reward on replay observed in
910 behaving animals ([Ambrose et al 2016](#)).

911 The second neural system required to achieve this integrative performance is a sequence learning
912 system that can integrate multiple subsequences (i.e. snippets) into a consolidated representation,
913 taking into consideration the probability distributions of replay so as to favor more frequently replayed
914 snippets. Here we considered a well-characterized model of sequence learning based on recurrent
915 connections in prefrontal cortex.

916 Replay mechanism: Replay is modeled using a procedure that randomly selects a subset of place-cells
917 coding part of a sequence, and outputs this snippet while taking into account the proximity of this
918 snippet to a future reward. Each time a reward is encountered, it is taken into consideration in
919 generating the snippet, and reward value is propagated backwards along the sequence, thus
920 implementing a form of spatio-temporal credit assignment. This can be viewed in the figures
921 illustrating the snippet probability densities. The replay mechanism also implements a second feature

922 observed in animal data, which is a tendency to replay snippets in reverse order. These two features of
923 the replay model correspond to what is observed in the rat neurophysiology, and they also make
924 fundamental contributions to the model's ability to converge on an efficient navigation path. This
925 extends previous demonstrations of the value of replay to include reward-modulated optimization
926 (Johnson & Redish 2005).

927 Reservoir network: The reservoir computing framework models the prefrontal cortex and its recurrent
928 connections. A reservoir network is able to learn stimuli sequence predictions by maintaining a form
929 of neuronal state history, resulting from multiple reverberations of neural activity, and the time
930 constant of these neurons related to the resistive and capacitive properties of a neuron's membrane. It
931 is a form of fading memory and when a reservoir network is exposed to a random replay of snippets,
932 its memory storing abilities are truncated by the duration of the snippet. The effect of snippet replay is
933 to break long-range time dependencies within an input sequence. The neurons state history is
934 necessarily shortened. After several time-steps, the reservoir neuron states are similar to those that
935 would have been observed by exposing the neural network to the entire sequence until it matches the
936 snippet. This coherency in neurons state coding and the truncation of the neurons states history allows
937 the reservoir network to naturally align and stitch together subsequences of place-cell activation in the
938 neurons state space. It is thus possible to learn a whole sequence from a random replay of snippets by
939 representing only short term dependencies between successive place-cell activation patterns. Long-
940 range time dependencies will not be represented nor learnt by the reservoir network. These results
941 yield the prediction that if the whole sequence is decomposed into snippets that do not feature long-
942 range time dependencies, the complete sequence could be learnt and generated, otherwise it will result
943 in a simplified sequence.

944 Effects of reward: The instantaneous reward information acquired during a past experience is used for
945 recursively updating the snippet replay likelihood in the hippocampus model. The resulting time
946 distribution of snippets features multiple modes defined around the moments that rewards are
947 obtained. This creates a reward gradient and allows sensory-motor associations to be learned by the
948 prefrontal cortex and striatum model. This is a novel form of reinforcement learning, and the main

949 effect of reward combined with hippocampus replay is to reinforce existing efficient paths between
950 rewards. A secondary effect of rewards could be observed when rewards are sufficiently close for
951 allowing a mutual contribution to the snippet replay likelihood surrounding the moments associated
952 with reward delivery. Thus, we predict that a cluster of reward sites will have the effect of propagating
953 the reward information farther than a single reward.

954 Effects of reverse replay: The reverse replay mechanism has a dual effect. First, a snippet replayed in
955 reverse order will play the role of a short-term support for the reward propagation in reverse direction
956 in the snippet replay likelihood online learning. Place-cell activation sequences leading to a nearby
957 reward are represented more frequently and earlier than other paths, and allow less efficient sub-paths
958 to be rejected. It results in a rewarding path selection mechanism by implementing a form of saliency
959 based on reward information. This is a form of spatio-temporal credit assignment that allows to take
960 advantage of the reservoir network ability to combine multiple snippets into a whole sequence. We
961 showed that it is possible to consolidate multiple sequences featuring parts of the same underlying
962 optimal sequence into one efficient sequence and to generate it autonomously. Second, when the
963 snippet replay likelihood is learned, a non-zero reverse replay rate allows the prefrontal cortex to be
964 exposed to sequences of place-cell activations in both forward and reverse direction. This results in
965 sequence learning in both directions while having experienced a place-cell activation sequence in one
966 direction only. These results can be tested experimentally by recording place cells activities in SWR
967 during the task.

968 Conclusions and limitations: The joint PFC-HP-ST model we studied here is able to mimic the rat's
969 ability to find good approximations to the traveling salesperson problem by taking advantage of recent
970 rewarding experiences for updating a trajectory generative model using hippocampus awake replay.
971 We showed that reverse replay allows the agent to reduce the TSP task complexity by considering an
972 undirected graph where feeders are vertices and trajectories are the edges instead of a directed graph.
973 In this case, autonomous sequence generation is no longer possible but the information available in
974 each prediction of the prefrontal cortex contains the expected locations. This allows the building of a

975 navigation policy taking into account the salient actions suggested by the prefrontal cortex predictions,

976 which are learned from hippocampus replay.

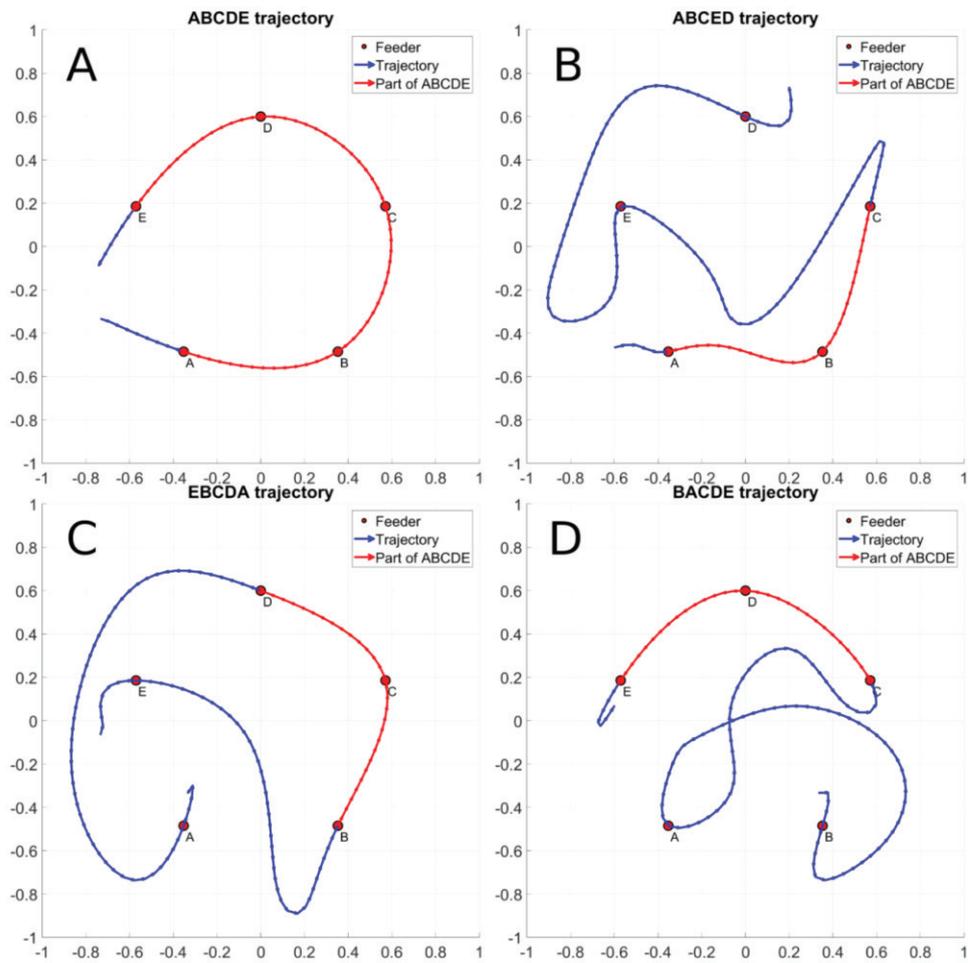
977

978 **References**

- 979 Ambrose RE, Pfeiffer BE, Foster DJ. 2016. Reverse replay of hippocampal place cells is uniquely
 980 modulated by changing reward. *Neuron* 91: 1124-36
- 981 Bureš J, Burešová O, Nerad L. 1992. Can rats solve a simple version of the traveling salesman
 982 problem? *Behavioural brain research* 52: 133-42
- 983 Carr MF, Jadhav SP, Frank LM. 2011. Hippocampal replay in the awake state: a potential substrate for
 984 memory consolidation and retrieval. *Nature neuroscience* 14: 147-53
- 985 Cenquizca LA, Swanson LW. 2007. Spatial organization of direct hippocampal field CA1 axonal
 986 projections to the rest of the cerebral cortex. *Brain research reviews* 56: 1-26
- 987 Cortes C, Vapnik V. 1995. Support-vector networks. *Machine learning* 20: 273-97
- 988 Davidson TJ, Kloosterman F, Wilson MA. 2009. Hippocampal replay of extended experience. *Neuron*
 989 63: 497-507
- 990 de Jong LW, Gereke B, Martin GM, Fellous J-M. 2011. The traveling salesrat: insights into the
 991 dynamics of efficient spatial navigation in the rodent. *Journal of Neural Engineering* 8:
 992 065010
- 993 Delatour B, Witter M. 2002. Projections from the parahippocampal region to the prefrontal cortex in
 994 the rat: evidence of multiple pathways. *European Journal of Neuroscience* 15: 1400-07
- 995 Diba K, Buzsaki G. 2007. Forward and reverse hippocampal place-cell sequences during ripples.
 996 *Nature neuroscience* 10: 1241
- 997 Dominey PF. 1995. Complex sensory-motor sequence learning based on recurrent state
 998 representation and reinforcement learning. *Biol Cybern* 73: 265-74
- 999 Eiter T, Mannila H. 1994. Computing discrete Frechet distance. *Tech. Report CD-TR 94/64,*
 1000 *Information Systems Department, Technical University of Vienna.*
- 1001 Enel P, Procyk E, Quilodran R, Dominey PF. 2016. Reservoir Computing Properties of Neural Dynamics
 1002 in Prefrontal Cortex. *PLoS computational biology* 12: e1004967
- 1003 Euston DR, Tatsuno M, McNaughton BL. 2007. Fast-forward playback of recent memory sequences in
 1004 prefrontal cortex during sleep. *Science* 318: 1147-50
- 1005 Foster DJ, Wilson MA. 2006. Reverse replay of behavioural sequences in hippocampal place cells
 1006 during the awake state. *Nature* 440: 680-83
- 1007 Gupta AS, van der Meer MAA, Touretzky DS, Redish AD. 2010. Hippocampal replay is not a simple
 1008 function of experience. *Neuron* 65: 695-705
- 1009 Harland B, Contreras M, Fellous J-M. 2018. A Role for the Longitudinal Axis of the Hippocampus in
 1010 Multiscale Representations of Large and Complex Spatial Environments and Mnemonic
 1011 Hierarchies In *The Hippocampus-Plasticity and Functions*: IntechOpen
- 1012 Jaeger H. 2001. The "echo state" approach to analysing and training recurrent neural networks-with
 1013 an erratum note'. *Bonn, Germany: German National Research Center for Information*
 1014 *Technology GMD Technical Report 148*
- 1015 Ji D, Wilson MA. 2007. Coordinated memory replay in the visual cortex and hippocampus during
 1016 sleep. *Nat Neurosci* 10: 100-7
- 1017 Johnson A, Redish AD. 2005. Hippocampal replay contributes to within session learning in a temporal
 1018 difference reinforcement learning model. *Neural Netw* 18: 1163-71
- 1019 Karlsson MP, Frank LM. 2009. Awake replay of remote experiences in the hippocampus. *Nat Neurosci*
 1020 12: 913-8
- 1021 Kudrimoti HS, Barnes CA, McNaughton BL. 1999. Reactivation of hippocampal cell assemblies: effects
 1022 of behavioral state, experience, and EEG dynamics. *J Neurosci* 19: 4090-101
- 1023 Lukosevicius M. 2012. A practical guide to applying echo state networks In *Neural networks: tricks of*
 1024 *the trade*, pp. 659-86: Springer
- 1025 Maass W, Natschlagler T, Markram H. 2002. Real-time computing without stable states: a new
 1026 framework for neural computation based on perturbations. *Neural Comput* 14: 2531-60

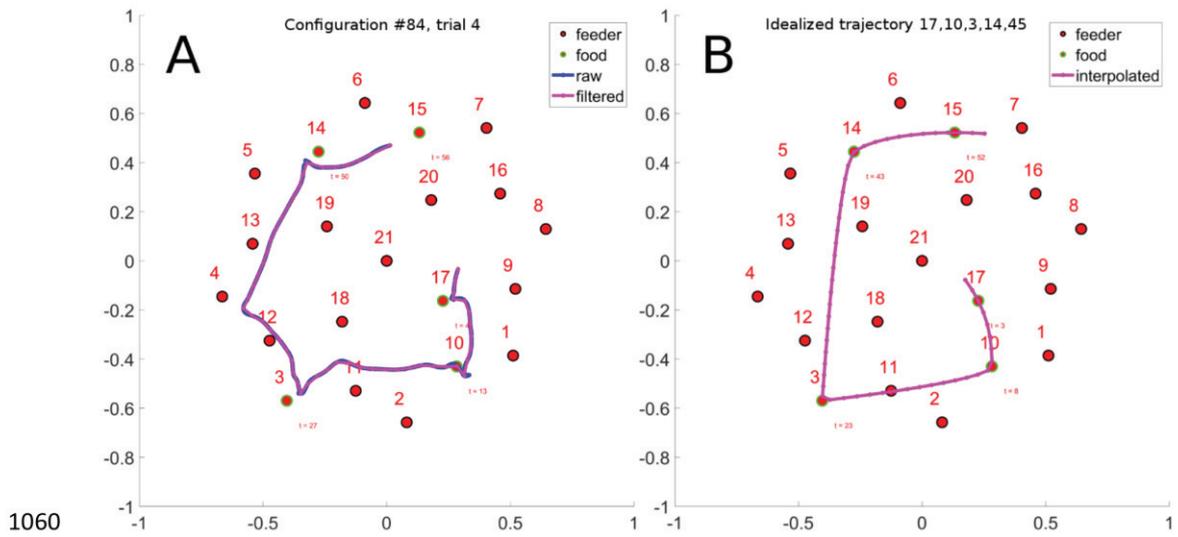
1027 McClelland JL, McNaughton BL, O'Reilly RC. 1995. Why there are complementary learning systems in
1028 the hippocampus and neocortex: insights from the successes and failures of connectionist
1029 models of learning and memory. *Psychol Rev* 102: 419-57
1030 Ram DJ, Sreenivas TH, Subramaniam KG. 1996. Parallel simulated annealing algorithms. *Journal of*
1031 *parallel and distributed computing* 37: 207-12
1032 Rigotti M, Barak O, Warden MR, Wang X-J, Daw ND, et al. 2013. The importance of mixed selectivity
1033 in complex cognitive tasks. *Nature*
1034 Shin JD, Jadhav SP. 2016. Multiple modes of hippocampal–prefrontal interactions in memory-guided
1035 behavior. *Current opinion in neurobiology* 40: 161-69
1036 Singer AC, Frank LM. 2009. Rewarded Outcomes Enhance Reactivation of Experience in the
1037 Hippocampus. *Neuron* 64: 910-21
1038 van der Meer MA, Johnson A, Schmitzer-Torbert NC, Redish AD. 2010. Triple dissociation of
1039 information processing in dorsal striatum, ventral striatum, and hippocampus on a learned
1040 spatial decision task. *Neuron* 67: 25-32
1041 Vertes RP, Hoover WB, Szigeti-Buck K, Leranath C. 2007. Nucleus reuniens of the midline thalamus:
1042 link between the medial prefrontal cortex and the hippocampus. *Brain research bulletin* 71:
1043 601-09
1044 Widrow B, Hoff ME. 1960. Adaptive switching circuits, STANFORD UNIV CA STANFORD ELECTRONICS
1045 LABS
1046 Wilson SW. 1991. The animat path to AI.
1047 Wylie TR. 2013. *The discrete Fréchet distance with applications*. Montana State University-Bozeman,
1048 College of Engineering
1049 Zhang K, Ginzburg I, McNaughton BL, Sejnowski TJ. 1998. Interpreting neuronal population activity by
1050 reconstruction: unified framework with application to hippocampal place cells. *Journal of*
1051 *neurophysiology* 79: 1017-44
1052
1053

1054 **Figures and Tables**



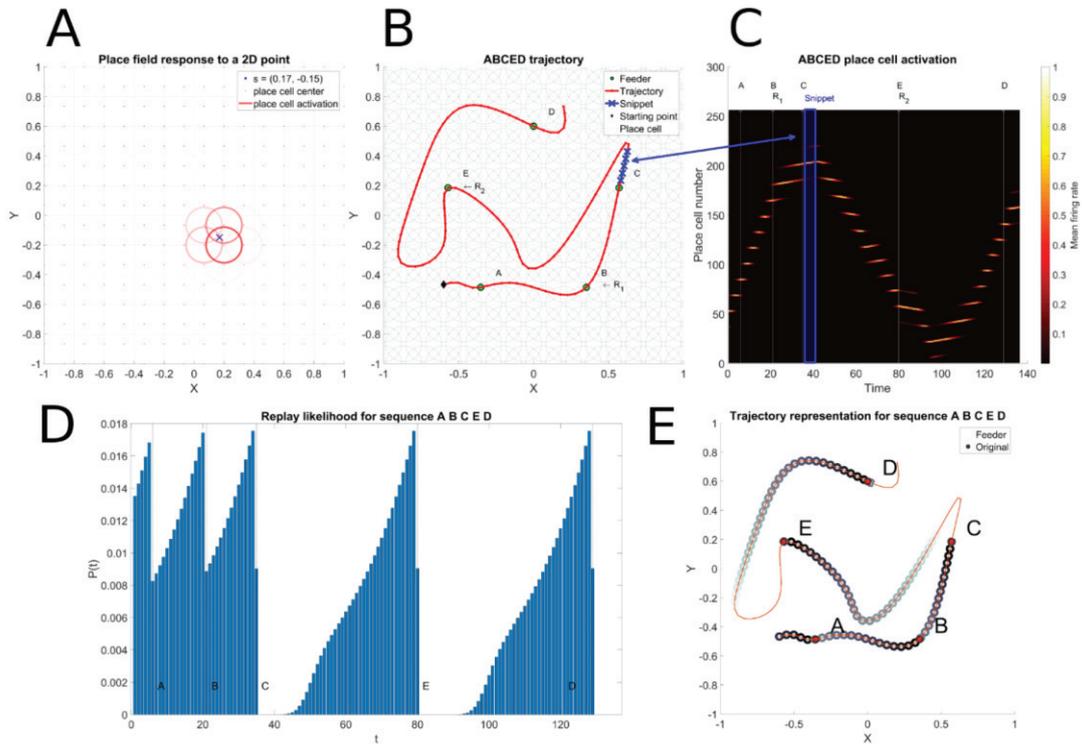
1055

1056 Figure 1: An optimal trajectory between feeders ABCDE is represented in panel A. Panel B, C and D display non optimal
 1057 trajectories that contain a sub trajectory of the ABCDE trajectory. The sub trajectory shared with the ABCDE trajectory is
 1058 displayed in red and the non-optimal parts in blue. Panel B contains the ABCED, panel C the EBCDA trajectory and panel D
 1059 the BACDE trajectory.



1060
 1061 Figure 2: Example Rat trajectory. A. One trial of the rat traversal of configuration 84. B. Idealized version of this trajectory
 1062 that can be compared with trajectories generated by the model.

1063



1064

1065

1066

1067

1068

1069

1070

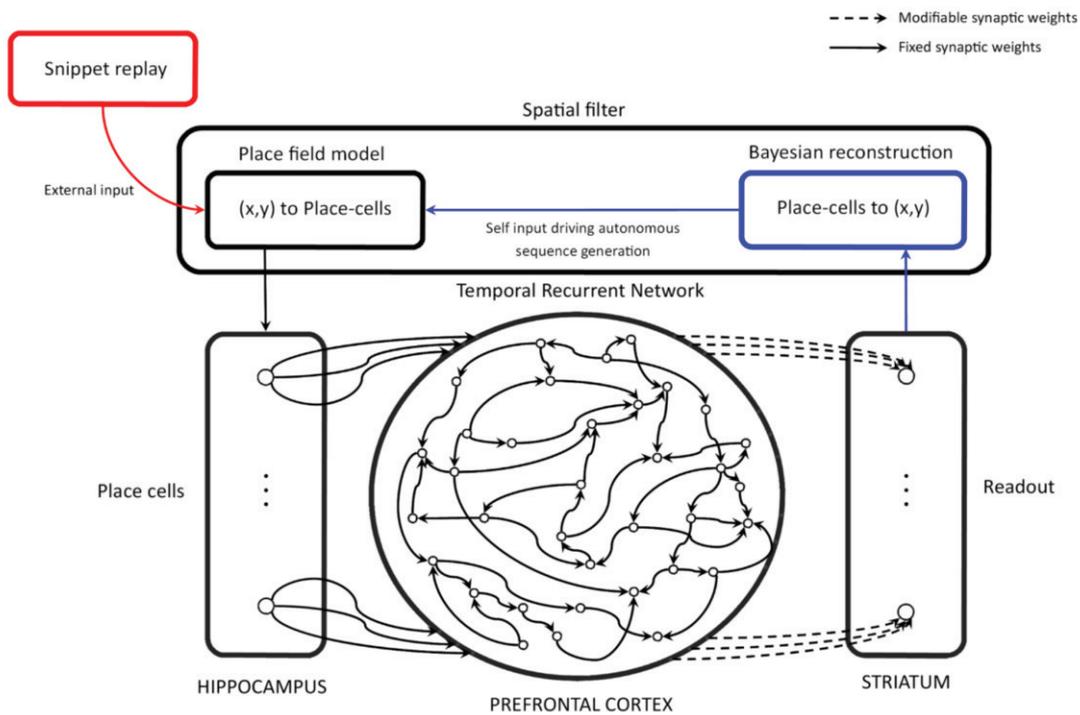
1071

1072

1073

1074

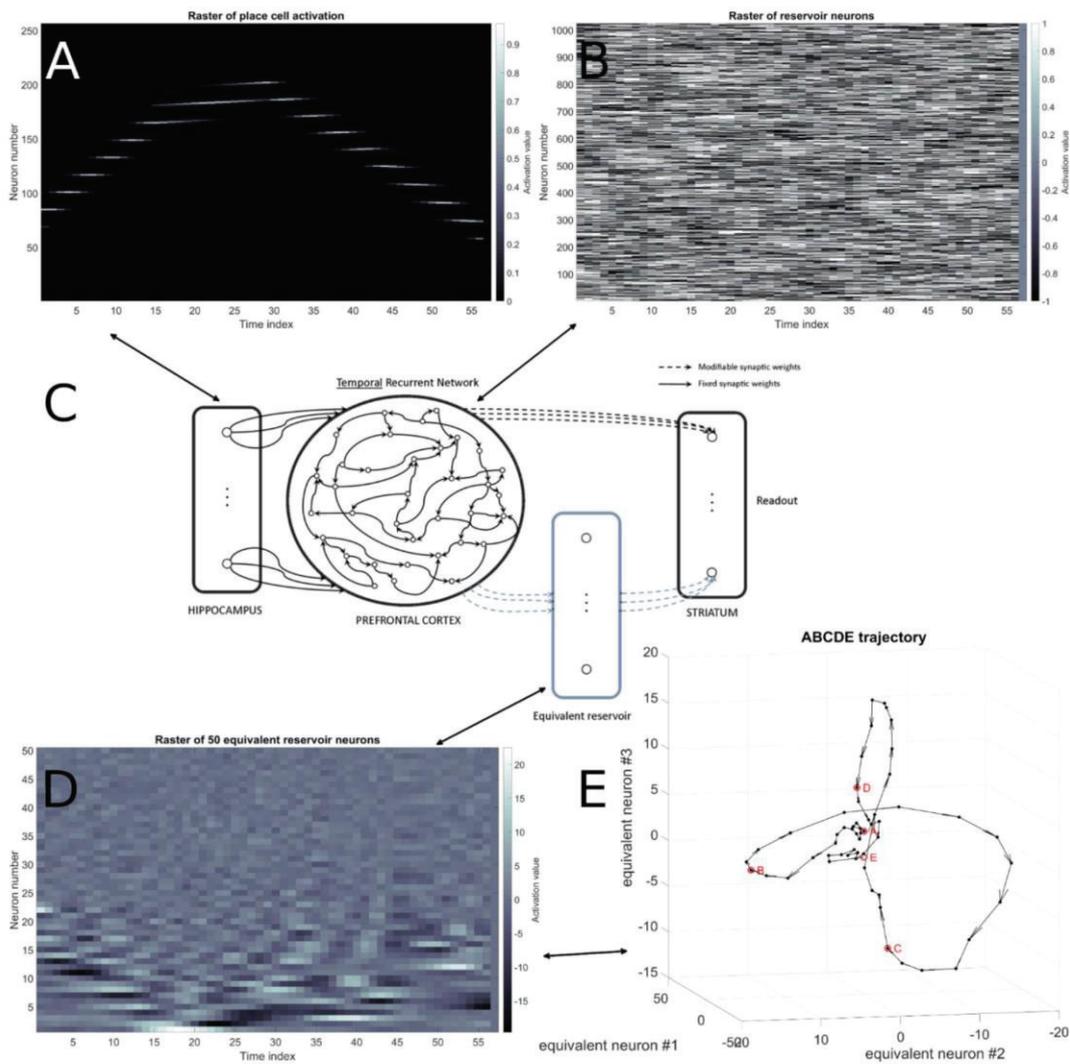
Figure 3: Place-cell and Snippet coding. Panel A represents the place-cell activations that correspond to a single point. Place-cell centers are represented by red points and the mean firing rate of each place-cell by a red circle with a fixed radius, centered on the place-cell center. The transparency level of the circle represents the magnitude of the mean firing rate. Panel B depicts the ABCED trajectory, the two randomly selected reward sites R_1 and R_2 and a snippet randomly drawn between R_1 and R_2 . The snippet length is $s = 5$. Panel C represents the raster of the place-cell activation along the ABCED trajectory. The time index where feeders A, B (R_1), C, D and E (R_2) are encountered during the ABCED trajectory are tagged above the raster and represented by a thin white vertical line. The snippet represented in panel B is emphasized by a blue rectangle in panel C. Panel D represents the snippet replay likelihood as learnt by the Hippocampal replay model and panel E represents the spatial extent of the snippet replay likelihood



1075

1076 Figure 4: Reservoir Computing Model. The Temporal Recurrent Network (TRN) is a model of the prefrontal cortex (PFC)
 1077 that take into account cortico-cortical loops by defining a fixed recurrent adjacency matrix for the leaky integrator neurons
 1078 that model PFC neurons. Inputs of the TRN are modelled hippocampus (HIPPP) place-cells. During the training phase, place-
 1079 cells activations are provided by the algorithmic model of SWR replay (red pathway), and the striatum model learns to
 1080 predict the next place-cell activation from the PFC model states by modifying the synaptic weights that project the PFC
 1081 model into the striatum model according the delta learning rule. During the generation phase, the model is no longer learning
 1082 and the place-cell activation patterns result from the new position of the agent, reconstructed with a Bayesian algorithm from
 1083 the next place-cell activation prediction of the modeled striatum (blue pathway)

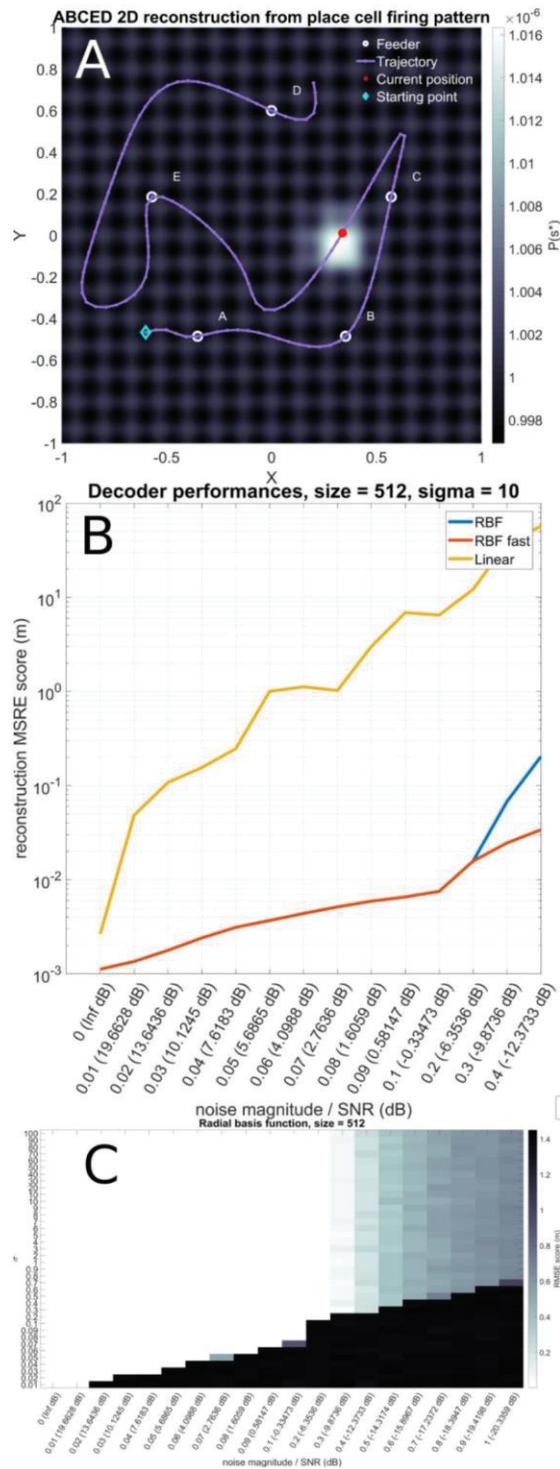
1084



1085

1086 Figure 5: Internal Representations in the reservoir . Panel A shows the raster of simulated hippocampus place-cell neurons
 1087 activation over time. Panel B represents the raster of the reservoir's neurons. Panel C depicts the hippocampus, prefrontal
 1088 cortex and striatum as we modelled them and features a supplementary artificial neuron group called equivalent reservoir
 1089 whose raster is represented in panel D. An equivalent reservoir neuron represents in fact a principal component, as computed
 1090 by the PCA algorithm. Panel E represents the raster observed in panel D as a 3D trajectory resulting from the 3 first principal
 1091 components.

1092



1093

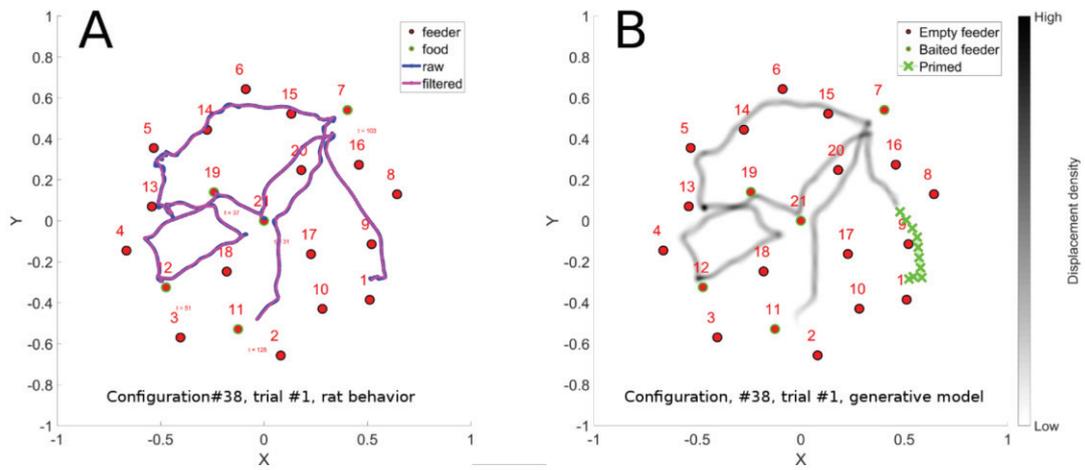
1094

1095

1096

Figure 6: Spatial Filter. Panel A represents the location probability map computed by our Bayesian reconstruction method and result from a Gaussian kernels with $\sigma = 10$ applied to the norm between the place-cell activation of each point of the map and the predicted next place-cell activation pattern computed by the reservoir model. The trajectory ABCED is overlaid

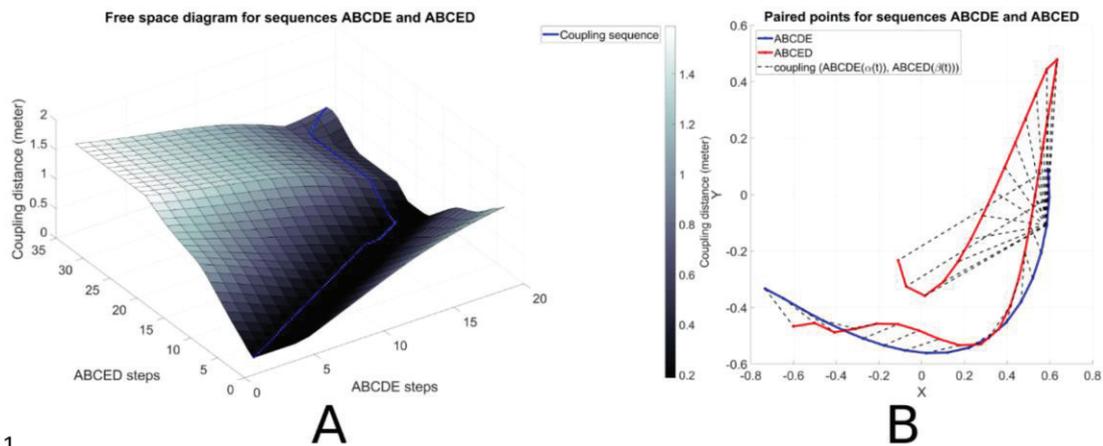
1097 on the map and the current position corresponding to the input activation pattern is shown in red. Panel B depicts the
1098 reconstruction error in function of the noise magnitude/signal to noise ratio of the linear direct model (Average of the place-
1099 cells centers weighted by the normalized mean firing rate of the place –cells), our radial basis function method (the Gaussian
1100 kernel is a radial basis function) computed over the full map and a faster version of our method where the computation is
1101 limited to a circle surrounding the current position. Panel C shows an extensive evaluation of the firing rate map based kernel
1102 decoding algorithm. Noise varies from 0% to 100% of mean firing rate and sigma varies from 0.01 to 100 in a semi
1103 logarithmic manner.



1104

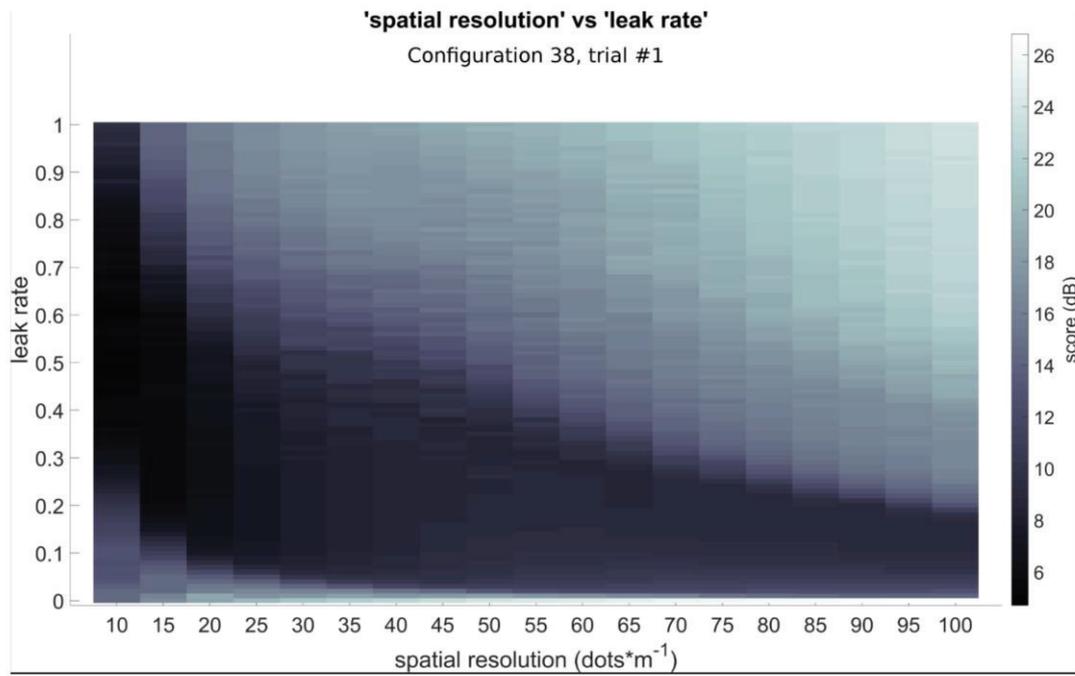
1105 Figure 7: Sequence learning. Panel A illustrates a long convoluted trajectory taken by a rat in configuration 38. Panel B
 1106 illustrates the probability maps of trajectories generated by the trained model in autonomous sequence generation mode. Note
 1107 that there are two locations where the trajectory crosses itself, which introduces ambiguity that the model is able to
 1108 disambiguate. This illustrates that the model is well able to learn such sequences.

1109



111
 1111 Figure 8: Fréchet Distance. Panel A shows the free space diagram used during the computation of the Fréchet distance
 1112 between ABCED and ABCDE trajectories. X and Y axis represent the discrete steps between points of ABCED and ABCDE
 1113 trajectories. Z axis relates in meter the coupling distance between two points. The coupling sequence is represented with a
 1114 blue line following the optimal alignment between the two trajectories (i.e. sequences of points). Panel B represents the
 1115 paired points of ABCDE and ABCED trajectories. During the common prefix ABC, curves are relatively similar and the
 1116 paired points are located close to each other. This is related by the diagonal part of the coupling sequence in panel A. Then
 1117 the diverging parts of ABCDE and ABCED are still paired but we can clearly observe a one-to-many relationship, near point
 1118 (0.6,-0.1) and the free space diagram relates it by variations along the ABCED steps axis and no variation along the ABCDE
 1119 steps axis.

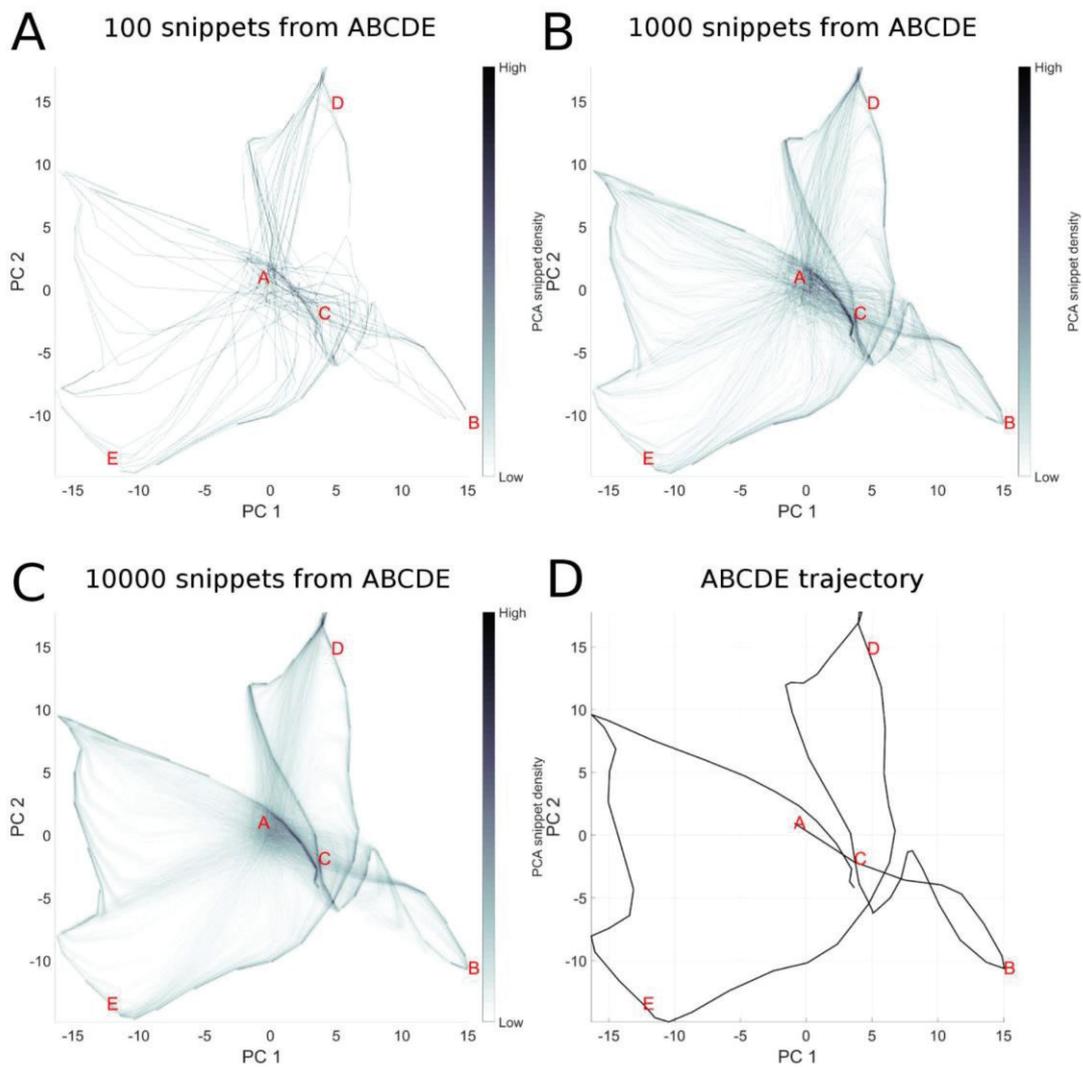
1120



1121

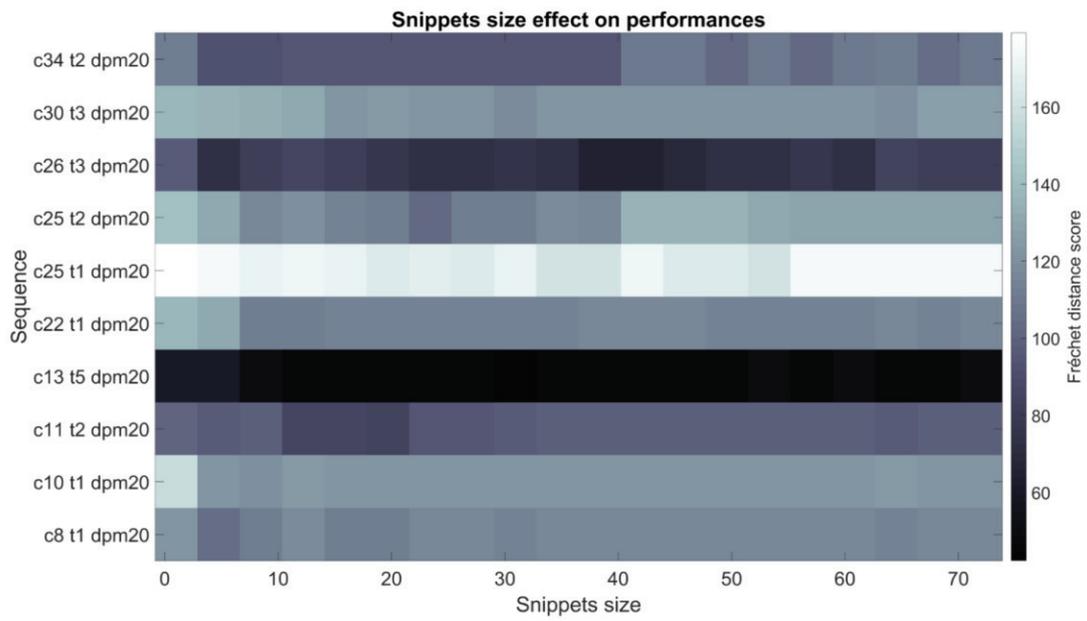
1122 Figure 9: Spatial resolution vs. reservoir neuron leak rate. As the spatial resolution increases, the number of discrete points
1123 that describe a sequence increases. We observe that as this occurs, the corresponding leak rate decreases. This indicates the
1124 relation between sequence length and local temporal memory implemented at the individual neuronal level.

1125



1126

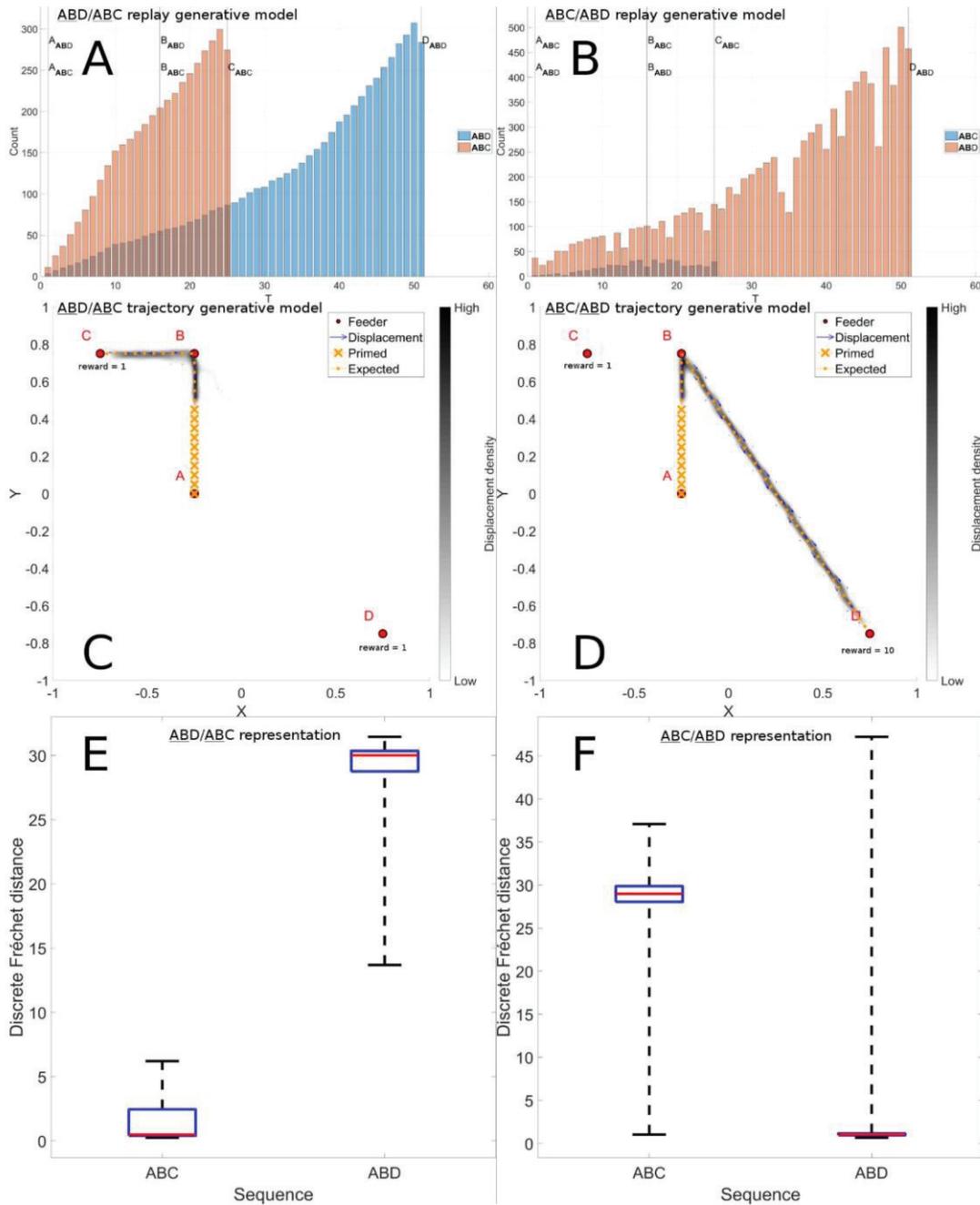
1127 Figure 10: Illustration of snippet integration in reservoir state space. Here we visualize the high dimensional reservoir space
 1128 in a low (2D) PCA space, in order to see how pieces (snippets) of the overall sequence are consolidated. In this experiment,
 1129 the sequence ABCDE is broken into snippets, which are then used to train the model. The challenge is that only local
 1130 structure is presented to the model, which must consolidate the global structure. Panels A-C represent the state trajectory of
 1131 reservoir activation after 100, 1000 and 1000 snippets. While each snippet represents part of the actual trajectory, each is
 1132 taken out of its overall spatial context in the sequence. Panel D represents the trajectory of reservoir state during the complete
 1133 presentation of the intact sequence. Panel C reproduces this trajectory, but in addition we see “ghost” trajectories leading to
 1134 the ABCDE trajectory. These ghost elements represent the reservoir state transitions from an initial random state as the first
 1135 few elements of each snippet take the reservoir from the initial undefined state onto the component of the ABCDE trajectory
 1136 coded by that snippet.



1137

1138 Figure 11: Effects of snippet size on sequence learning for sequences of varying difficulty. Sequences from ten
 1139 configurations of the rat TSP data set were used with snippets of different sizes varying from 0 to 70. We observe that for
 1140 most sequences, error is reduced as snippet size increases.

1141

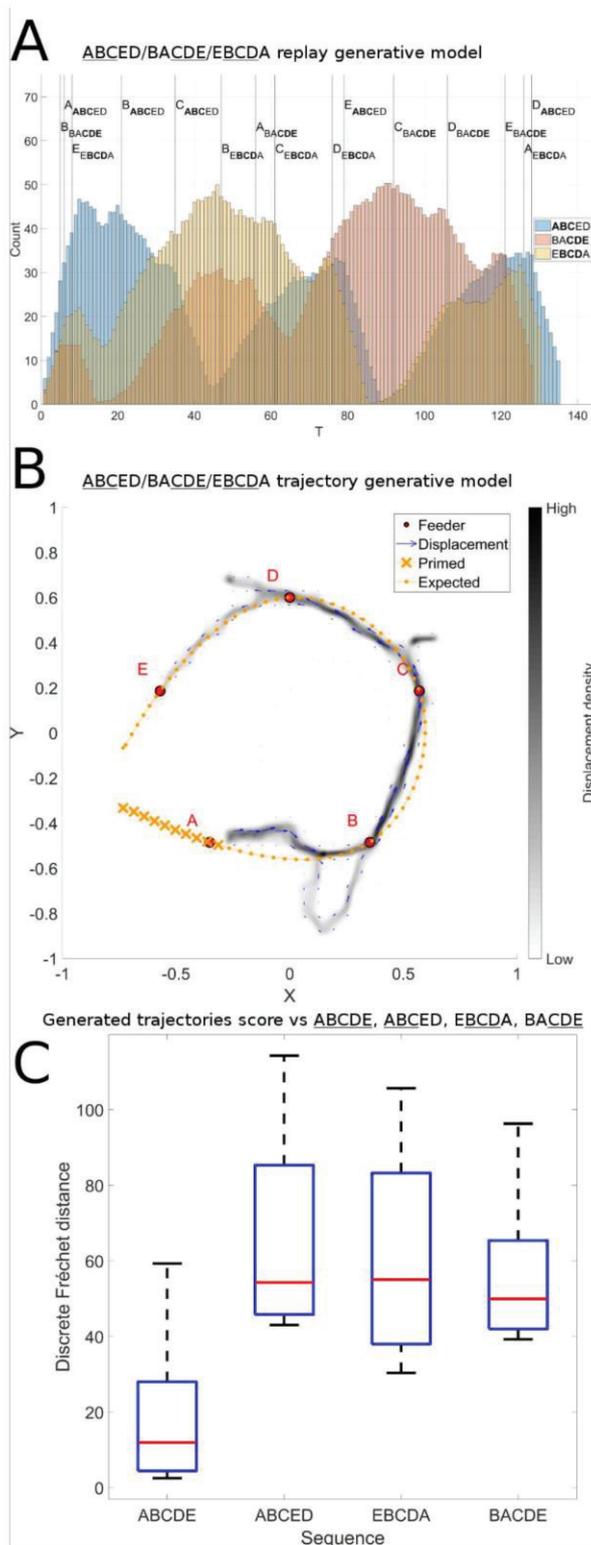


1142

1143 Figure 12: Longer paths are rejected. Panels A and B illustrate snippet counts for T maze trajectories pictured in panels C
 1144 and D. In Panel C, sequences begin at location A, and rewards are given at locations C and D. Based on the reward proximity
 1145 and propagation, there is a higher probability of snippets being selected along path AC than path AD. This is revealed in
 1146 panel A, a histogram of snippets for the sequences ABC (in Blue) and ABD (in Orange). Panels B and D illustrate how
 1147 distance and reward intensity interact. By increasing the strength of the reward, a longer trajectory can be rendered virtually
 1148 shorter and more favored, by increasing the probability that snippets will be selected from this trajectory, as revealed in Panel

1149 B. Panel E and F confirm a neat tendency to generate autonomously sequences significantly similar to the ABC and ABD
1150 sequence respectively (p-value = 0).

1151



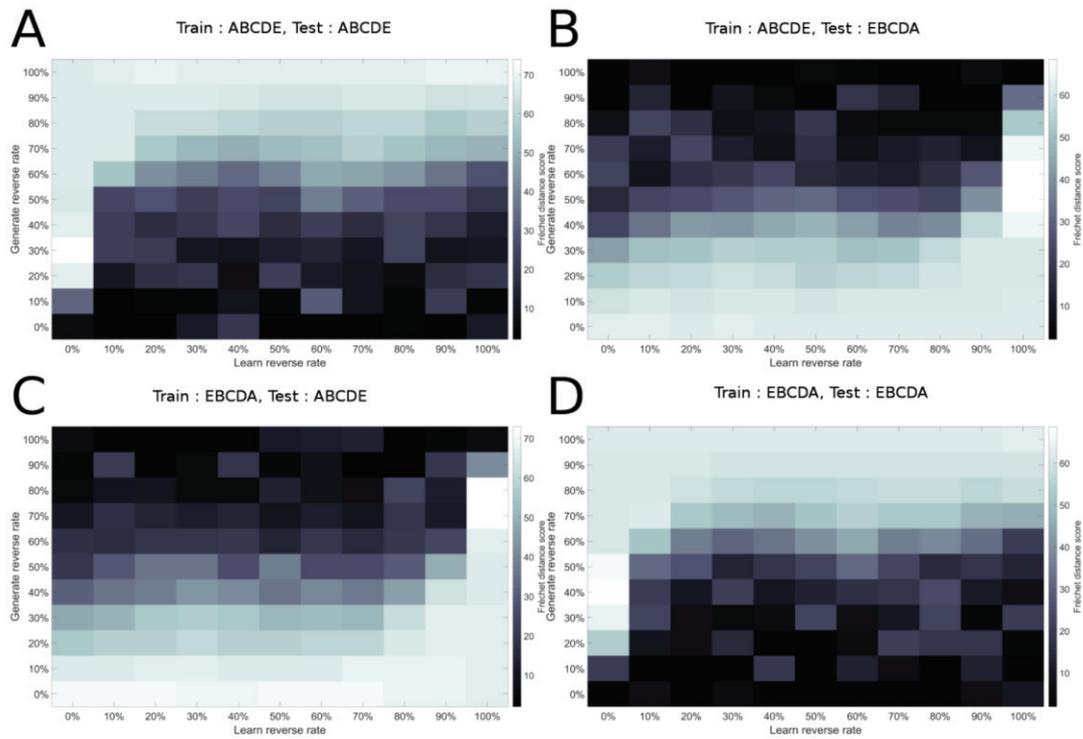
1152

1153 Figure 13: Efficient Sequence Synthesis. A. Distribution of snippets drawn from the sequences illustrated in Figure 1 B, C

1154 and D. Globally we observe snippet selection favors snippets from the beginning of sequence ABCED (blue), the middle of

1155 EBCDA (yellow), and the end of sequence BACDE (pink), which corresponds exactly to the efficient subsequences (ABC,
1156 BCD, and CDE) of these three sequences. This distribution of snippets is used to train the model. The results of the training
1157 are illustrated in panel B. Here we see a 2D histogram of sequences generated by the model in the ABCDE recombination
1158 experiment. 10 batches of 100 reservoirs each were trained and each model instance was evaluated 10 times with noise. Panel
1159 C confirms that the trajectories generated autonomously are significantly more similar to the target sequence ABCDE (p-
1160 value = 5.9605e-08) These results are very robust and satisfying, demonstrating that our hypothesis for efficient sequence
1161 discovery based on reward-modulated replay is validated.

1162



1163

1164

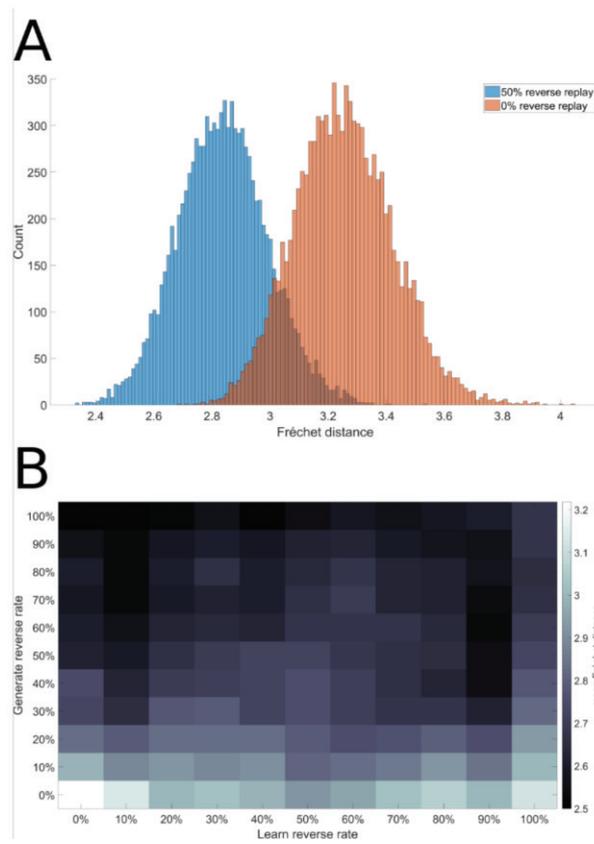
1165

1166

1167

1168

Figure 14: Effects of reverse replay and initial sequence direction. When a sequence is learned and tested in the same direction, introduction of reverse replay gradually impairs performance (panels A and D). When training and testing direction are reversed, then the introduction of reverse replay gradually improves performance. Interestingly, a mixture of forward and reverse replay leads to a general compromise for all situations.

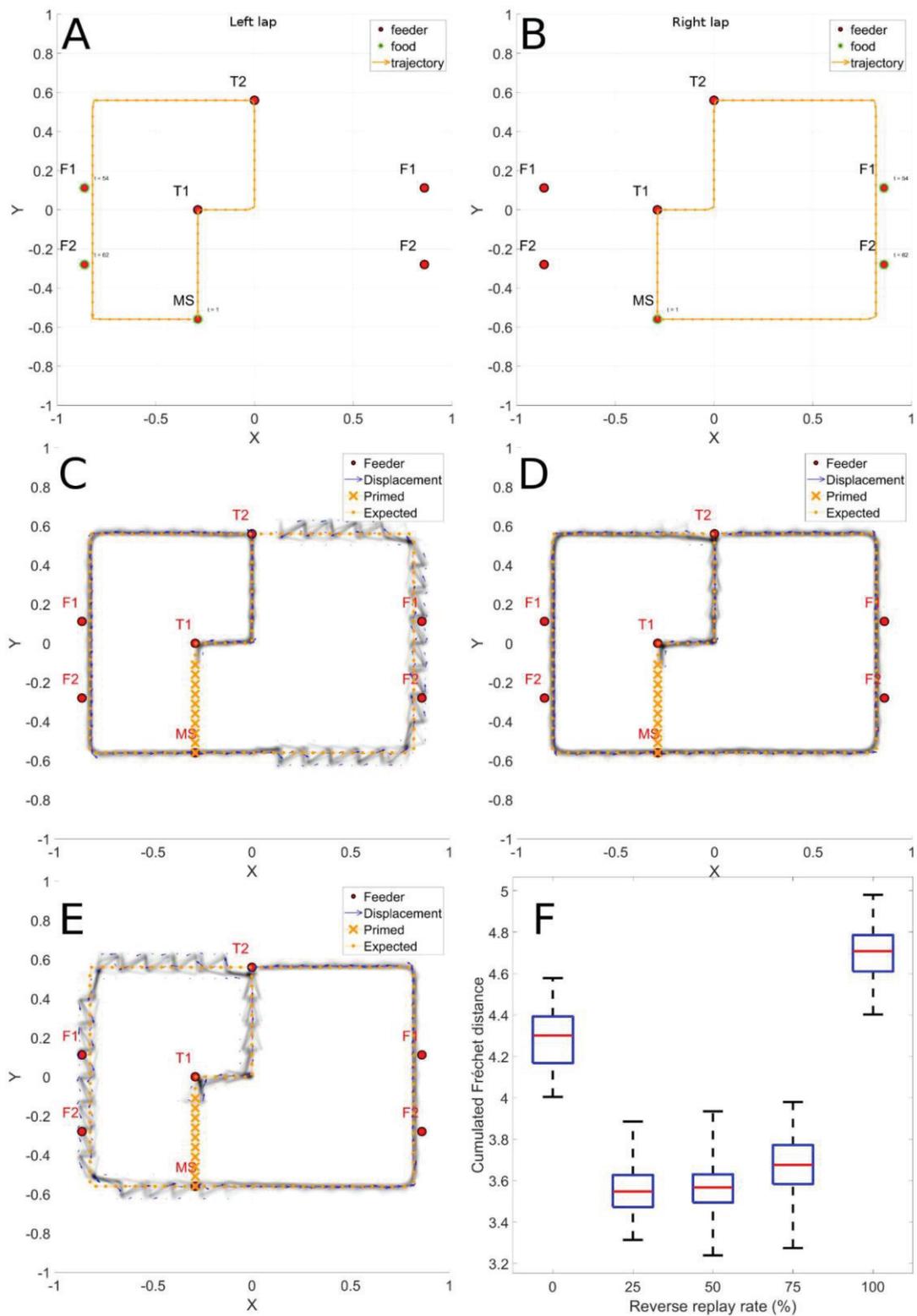


1169

1170 Figure 15: Reverse replay facilitates efficient sequence discovery. Using the same sequences illustrated in Figure 1, we

1171 reversed the direction of sequence EBCDA, and then tested the model's ability to synthesize the ABCDE sequence from

1172 ABCED ACDBE and BACDE



1173

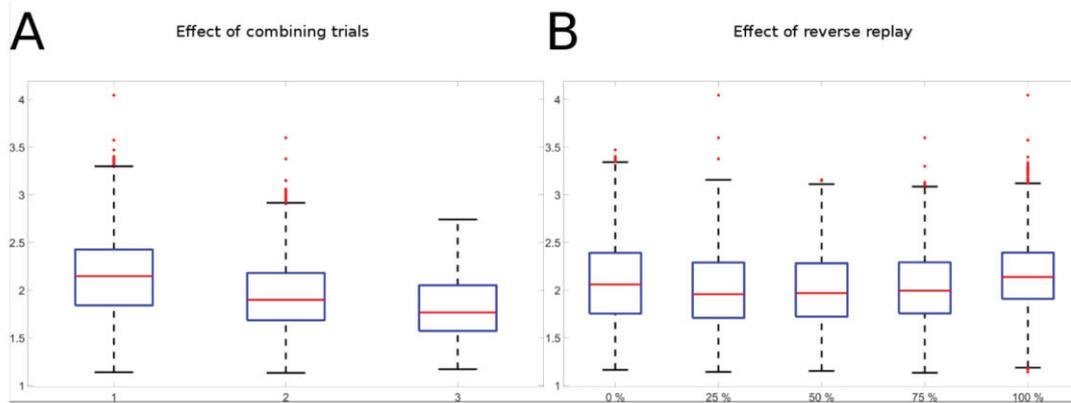
1174 Figure 16: Reverse replay allows novel shortcut path generation. Panels A and B illustrate the trajectories for left and right

1175 trajectories, based on Gupta et al. After training on these two trajectories, we test the ability to generate a shortcut that makes

1176 the complete outer loop in one direction. Panel C – without reverse replay, significant spatial errors are revealed when the
1177 system attempts to complete the counter-clockwise loop on the right side of the maze Panel D illustrates the beneficial effects
1178 of reverse replay during trajectory learning. Panel E illustrates the effect of a model training with 100% reverse replay. It is
1179 similar to using a 0% reverse replay but the effect is observed on the left lap trajectory part. Panel F – when reverse replay is
1180 introduced, this error is attenuated.

1181

1182



1183

1184 Figure 17: Consolidation and reverse replay applied to behavioral data. Data from the rat TSP configurations are used for
1185 training and testing the model. A. Effects of consolidation: as successive trials are added to the replay repertoire; the
1186 trajectory reconstruction error is significantly reduced. B. Effects of reverse replay: as reverse replay is introduced in snippet
1187 formation for training the PFC model, reconstruction error is significantly reduced.

1188

Solution number	in_scale	leak_rate	rec_scale	score
1	'6.379720'	'0.494997'	'6.553708'	1,1879078
2	'5.212805'	'0.481997'	'5.542781'	1,1900330
3	'6.429717'	'0.485997'	'6.553708'	1,1994574
4	'5.189806'	'0.480997'	'5.542781'	1,2003332
5	'5.404791'	'0.483997'	'5.542781'	1,2313991
6	'5.256802'	'0.508997'	'5.542781'	1,2524340
7	'9.084041'	'0.445998'	'8.420773'	1,2691536
8	'5.211805'	'0.495997'	'5.542781'	1,2712017
9	'6.322724'	'0.492997'	'6.952679'	1,2836924
10	'8.297723'	'0.459998'	'5.542781'	1,3022404

1189 Table 1: summary of the 10 parameter sets optimized by the parallel simulated annealing algorithm. Column in_scale
1190 represents the scaling constant applied to the feedforward connectivity matrix W_{in} , rec_scale represents the scaling
1191 constant applied to the recurrent connectivity matrix W_{rec} , normalized by $\frac{1}{\sqrt{N}}$ and ensuring an approximately constant
1192 spectral radius of $\rho \approx 3.87$. Column leak_rate represents different values of the leak rate of the reservoir neurons and
1193 solution number, the rank of the parameter set sorted in increasing order

1194

Snippets size		1	5	9	12	16	20	23	27	31
best	Score	0.53583	0.13909	0.037858	0.037005	0.021245	0.015277	0.014615	0.013871	0.020241
	Population %	29.9	52.1	91.2	98.6	98.2	94.9	83.7	77.8	99
others	Score	0.98145	0.92174	0.73686	1.0041	0.87086	<u>0.029701</u>	0.025379	0.022013	0.95653
	Population %	70.1	47.9	8.8	1.4	1.8	<u>5.1</u>	16.3	22.2	1

Table 2: Performance (revealed by Fréchet distance to desired trajectory) as a function of snippet size.

1195

1196

Reverse rate	0%	25%	50%	75%	100%
0%	N.A.	1.1921e-07	1.1921e-07	1.1921e-07	1.1921e-07
25%	1.1921e-07	N.A.	1.1921e-07	0.017098	1.1921e-07
50%	1.1921e-07	1.1921e-07	N.A.	1.1921e-07	1.1921e-07
75%	1.1921e-07	0.017098	1.1921e-07	N.A.	1.1921e-07
100%	1.1921e-07	1.1921e-07	1.1921e-07	1.1921e-07	N.A.

1197 Table 3: p-values of different combinations of reverse replay rate (ANOVA1)

1198

Clusters of conditions number		Edge direction		
		Forward	Backward	Both
Required number of trial(s)	1	∅	84	41,43,53,78,84
	2	8,25,33,41,44,48,52,70,78	11	8,11,25,29,32,33,38,40,44,52,56,70,82
	3	97	58,80	48,80,97
Total number of configurations		10	4	21

1199 Table 4: Classification of rat behavioral configurations by direction in which executed trajectories relate to desired trajectory.

1200





Click here to access/download

Supporting Information - Compressed/ZIP File Archive
rat behavior.zip



2.5. Real-time sensory-motor integration of hippocampal place-cell replay and prefrontal sequence learning in a simulated rat robot (Experimental results to come)

1

2

3 **Title:** Real-Time Sensory-Motor integration of Hippocampal Place Cell Replay and
4 Prefrontal Sequence Learning in a Simulated Rat Robot

5 **Authors:** Pablo Scleidorovich¹, Nicolas Cazin², Bruce Harland³, Jean-Marc Fellous³, Alfredo
6 Weitzenfeld¹, Peter Ford Dominey²

7

8 1. College of Engineering, University of South Florida, USA

9 2. Human and Robot Cognitive Systems Group, INSERM, France

10 3. Department of Psychology, University of Arizona, USA

11

12

13 **Abstract:** An open problem in the cognitive dimensions of navigation concerns how previous
14 exploratory experience is re-organized in order to allow the creation of novel efficient
15 navigation trajectories. This behavior is revealed in the “traveling sales-rat problem” when
16 rats discover the shortest path linking baited food wells after a few exploratory traversals. We
17 have developed a model of navigation sequence learning, where sharp wave ripple replay of
18 hippocampal place cells transmit “snippets” of the recent trajectories that the animal has
19 explored to the prefrontal cortex (PFC). PFC is modeled as a recurrent reservoir network that
20 is able to assemble these snippets into the efficient sequence. While this demonstrates the
21 theoretical feasibility of the PFC-HIPP interaction, the integration of such a dynamic system
22 into a real-time sensory-motor system remains a challenge. In the current research we test the
23 hypothesis that the PFC reservoir model can operate in a real-time sensory-motor loop. Place
24 cell activation encoding the current position of the simulated rat robot feeds the PFC reservoir
25 which generates the successor place cell activation that represents the next step in the
26 reproduced sequence. This is played into the robot simulation of the navigating rat, which
27 advances to the coded location and then generates de-novo the current place cell activation. A
28 crucial aspect of this interaction is the transformation of the place cell vector output of PFC to
29 a physical displacement, and then the recoding of the new location in a place cell activation
30 vector. This recoding allows for a normalization of the place cell input to the PFC which
31 yields significant performance improvements. We demonstrate how this integrated sensory-
32 motor system can learn simple navigation sequences, and then, importantly, how it can
33 synthesize novel efficient sequences based on prior experience. The model of hippocampal
34 replay generates a distribution of snippets as a function of their proximity to a reward, thus
35 implementing a form of spatial credit assignment that solves the TSP task. The integrative
36 PFC reservoir reconstructs the efficient TSP sequence based on exposure to this distribution
37 of snippets that favors paths that are most proximal to rewards. This contributes to the
38 understanding of hippocampal replay in novel navigation sequence formation.

39

40

41

42

43 Introduction:

44 As rats learn to search for multiple sources of food or water in a complex environment,
45 processes of spatial sequence learning and recall in the hippocampus (HC) and prefrontal
46 cortex (PFC) are taking place. Recent studies show that spatial navigation in the rat
47 hippocampus involves the replay of place cell firing during awake and sleep states generating
48 small sequences of spatially related place cell activity we call “snippets”. These “snippets”
49 occur primarily during sharp-wave-ripple (SPWR) events. Much attention has been paid to
50 replay during sleep in the context of long term memory consolidation. Here we focus on the
51 role of replay during the awake state, as the animal is learning across multiple trials. We
52 hypothesize that these “snippets” can be used by the PFC to achieve multi-goal spatial
53 sequence learning. We develop an integrated model of hippocampus and PFC that is able to
54 form spatial navigation sequences based on snippet replay. This extends our existing spatial
55 cognition model for simpler goal-oriented tasks (Barrera and Weitzenfeld, 2008; Barrera et al,
56 2011) with a new replay-driven model for memory formation in the hippocampus and spatial
57 sequence learning and recall in PFC.

58

59 *** Insert Figure 1: Rodent and robot experiments. ***

60 In contrast to existing work on sequence learning that relies heavily on sophisticated learning
61 algorithms and synaptic modification rules, we propose to use an alternative computational
62 framework known as ‘reservoir computing’ (Dominey 1995) in which large pools of prewired
63 neural elements process information dynamically through reverberations. This reservoir
64 computational model consolidates snippets into larger spatial sequences that may be later
65 recalled by subsets of the original sequences. We constrain the model using
66 electrophysiological recordings in rodents in a multi-goal spatial task that is known to involve
67 the hippocampus and the PFC (Watkins de Jong et al 2011). The resulting integrated system
68 allows us to generate a new understanding of the role of replay in memory acquisition in
69 complex tasks such as sequence learning. That operational understanding is leveraged and
70 tested on a simulated robotic platform. The originality and contribution of this work includes
71 1) the use of awake hippocampal replay to create small chunks of valid trajectories (snippets)
72 that favor the TSP sequence, 2) the use of reservoir computing to learn spatial sequences
73 using inputs generated by our hippocampus model, 3) our constraining and testing of the

74 model using electrophysiological data in rats and 4) the use of the resulting model in the
75 embodied-cognitive real-time framework of a robot.

76 The following sections overview the key background areas for the proposed research: (1)
77 sleep and awake replay in rats, (2) place cell models of spatial cognition, and (3) reservoir
78 computing and sequence learning.

79 Snippet Replay in Rats

80 The hippocampus stores information during the acquisition of new memories and these
81 memories are replayed as “snippets” during sleep as part of a memory consolidation process
82 (Marr, 1971; Stickgold & Walker, 2007). Consolidation is believed to involve synaptic
83 changes in the neocortex reflecting the integration and refinement of memory representations
84 (McClelland, McNaughton, & O'Reilly, 1995; Schwindel & McNaughton, 2011). This replay
85 involves neural populations that were active during a task immediately preceding the sleep
86 period. Reactivations of specific neural activity patterns during sleep have been observed in
87 several brain areas including the hippocampus, amygdala, neocortex and striatum (Bendor &
88 Wilson, 2012; Carr et al., 2012 ; Euston et al. 2007; Foster & Wilson, 2006; Hoffman &
89 McNaughton, 2002; Ji & Wilson, 2007; Karlsson & Frank, 2009; Kudrimoti et al., 1999; Lee
90 & Wilson, 2002; Nadasdy et al. 1999; Pennartz et al., 2004; Peyrache et al. 2009; Popa et al.
91 2010; Ribeiro et al., 2004; Sutherland & McNaughton, 2000; Tatsuno et al. 2006; Cutsuridis
92 & Hasselmo 2011; Cutsuridis & Taxidis 2013). Other evidence suggests that replay may also
93 occur during the awake state indicating online memory processes or the planning of behaviors
94 yet to be performed (Carr et al. 2011; Davidson et al. 2009; Diba & Buzsaki, 2007; Gupta, et
95 al. 2010). In the hippocampus, it has been shown that reactivation occurs primarily in a
96 compressed manner, during the occurrence of fast (150-200 Hz) and short (60-120ms)
97 oscillations called sharp waves/ripples complexes (SPWR). Different subsets of cells
98 reactivate in different SPWRs, each cell emitting only a few spikes. The interspike interval
99 between reactivating cells is within the range of that required to induce spike-timing
100 dependent synaptic plasticity (STDP). One hypothesis therefore is that the sequence of
101 reactivation episodes allows for online (awake replay, focus of this proposal) and offline
102 (sleep replay) synaptic modifications that will eventually lead to the consolidation and
103 integration of specific memory items.

104 Interestingly, the presence of rewards increases replay in hippocampus and ventral striatum
105 (Lansink et al., 2008; Singer & Frank, 2009), suggesting an interaction between reinforcement

106 learning and replay. The computational mechanisms underlying these reactivations and their
107 possible consequences on learning have been investigated in hippocampus (Hasselmo, 2008;
108 Johnson & Redish, 2005). It was suggested that replay may improve reinforcement learning
109 algorithms, in simple tasks in which rats had to memorize a sequence of turns in a corridor-
110 type maze with a single goal location (Johnson & Redish, 2005).

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

116 Our working hypothesis is that forward replay during active learning triggers the formation of
117 ‘snippets’, short sequences of place cell firings ordered in the order they were traversed. We
118 also hypothesize that backward replay allows for ‘credit assignment’ of the snippets when a
119 reward is obtained, hence reinforcing those paths that yielded a reward (Foster & Wilson,
120 2006).

121 There have been many quantitative analyses of individual SPWRs in hippocampus and of
122 their synchronization with other areas such as the PFC. There has however been little detailed
123 account of how the neuronal population that fires during each SPWR varies from trial to trial,
124 as the animal learns. Here, we give such an account, as the animal learns to ‘link’ (create a
125 sequence of) multiple rewarded cup locations.

126 These hypotheses have never been implemented and tested in a complex task in which the
127 animal is free to learn and recall at will (i.e. ‘operant’ learning: no supervised/directed
128 ‘learning’ phase). Many fundamental questions remain to be answered.

129 Computational Models of Spatial Cognition in Hippocampus and Robotics Experimentation

130 The study of behavioral and neurophysiological mechanisms in rats responsible for spatial
131 cognition has inspired the development of many computational models of hippocampus place
132 cells in the context of goal-oriented learning tasks in robotic systems. Some of the most
133 important models developed in the past years include those of Burgess, Recce, and O’Keefe
134 (1994), Brown and Sharp (1995), Redish and Touretzky (1997), Guazzelli, Corbacho, Bota,
135 and Arbib (1998), Arleo and Gerstner (2000), Gaussier, Revel, Banquet, and Babeau (2002),
136 Filliat and Meyer (2002), Arleo, Smeraldi, and Gerstner (2004), Milford and Wyeth (2007,

137 2010), Dollé, Sheynikhovich, Girard, Chavarriaga, and Guillot (2010), Alvernhe, Sargolini,
138 and Poucet (2011), Caluwaerts et al (2012), and Barrera and Weitzenfeld (2008, 2011). The
139 complete system provides a spatial cognition model for navigation in simulated and real robot
140 in corridor-based mazes and open environments exploiting external landmarks and/or path
141 integration. The work described in Barrera & Weitzenfeld (2008) replicated O’Keefe and
142 Nadel (1978) and O’Keefe (1983) experiments with T-mazes and 8-arm mazes learning the
143 interaction between visual and idiothetic cues (Jeffery and O’Keefe 1999). Barrera, et al
144 (2011) extended the experiments to cyclical corridors.

145 In this model, place cells represented spatial locations that were used to navigate in goal-
146 oriented tasks based on topological maps integrating information from path integration
147 (Mittelstaedt & Mittelstaedt 1982) and landmarks as described in Weitzenfeld et al (2012). In
148 Tejera, Barrera, Fellous, Llofriu and Weitzenfeld (2013) we described a model extension to
149 spatial cognition in open arena environments, i.e. having no corridors, as in the Morris water
150 maze (1981). Recently, the computational model was extended to incorporate grid and head
151 direction cells, i.e. “biological odometry” as described in Tejera, Barrera, Llofriu and
152 Weitzenfeld (2013). This system provides the robotic simulation framework for our integrated
153 experiments linking a sensory-motor system to the PFC reservoir model for sequence
154 generation.

155 Reservoir Computing and Sequence Learning

156 Reservoir computing refers to a class of neural network models in computational
157 neuroscience and machine learning (Lukosevicius and Jaeger 2009). These systems are
158 characterized by a sparsely connected recurrent network of neurons (spiking or analog), with
159 fixed connection weights (excitatory and inhibitory). Because of the recurrent connections,
160 this “reservoir” is a dynamical system that has inherent sensitivity to the serial and temporal
161 structure of input sequences. Reservoir neurons are connected to readout neurons by
162 modifiable connections, and these can be trained in different task (e.g. sequence recognition,
163 prediction, classification). The first instantiation of such models was by Dominey, with the
164 reservoir corresponding to recurrent prefrontal cortical networks, and the modifiable readout
165 connections corresponding to the corticostriatal projections, with dopamine-modified
166 synapses (Dominey 1995; Dominey, et al. 1995). These models addressed sensorimotor
167 sequence learning, and demonstrated the inherent sensitivity of these recurrent systems to
168 serial and temporal structure in motor behavior and in language (Dominey 1998a; Dominey

169 1998b; Dominey and Ramus 2000; Dominey et al. 2009; Hinaut and Dominey 2013). Maass
170 developed a related approach with spiking neurons and demonstrated the non-linear
171 computational capabilities of these systems (Maass et al. 2002). In the machine learning
172 context, Jaeger demonstrated how such systems have inherent signal processing capabilities
173 (Jaeger and Haas 2004). Interestingly, these reservoir properties appear to be found in cortex.
174 Electrophysiological studies have revealed that cortical neurons in primary sensory areas (e.g.
175 V1) have reservoir properties of fading memory (Nikolic et al. 2009). That is, stimuli
176 presented in the past tend to resonate in the recurrent network and influence the processing of
177 subsequent stimuli. Equally interestingly, when these networks are exposed to inputs with
178 multiple dimensions (e.g. target identification, serial order, match/non-match) neurons
179 represent non-linear mixtures of these dimensions (Dominey et al. 1995; Rigotti, et al. 2010).
180 Such nonlinear mixed effects have recently been seen in primate frontal cortex (Rigotti, et al.
181 2013). This argues in favor of a reservoir-like function in recurrent networks of the cortex in
182 general, and in prefrontal cortex specifically. We have demonstrated how such recurrent
183 networks can learn about sequential and temporal structure (Dominey 1998a), including serial
184 order regularities that are expressed in sequence segments (Dominey and Ramus 2000).
185 However, so far, reservoir computing has not been exploited in terms of its inherent ability to
186 allow the concatenation of multiple contiguous subsequences into a coherent sequence, thus
187 addressing a major open question in navigation trajectory learning.

188 The most prevalent theories of how memories are formed and consolidated rely on dynamic
189 changes in synaptic strengths and the creation of strongly connected neural assemblies. This
190 Hebbian view of memory has dominated the field for decades (Hebb 1949). While there is
191 strong experimental evidence for several aspects of the theory, there are also significant
192 difficulties that include 1) the unclear nature of fast induction and exquisite control of
193 synaptic modifications in the presence of realistic amount of neuronal and synaptic noise and
194 unreliability, 2) the low probability that 2 cortical neurons are in fact connected by synapses
195 and 3) the pluri-functionality of cortical neural networks whereby small groups of cells may
196 be involved in various and seemingly distinct neural computations in addition to memory
197 formation (how these multiple functions co-exist with memory is unclear).

198 Reservoir Computing offers an alternative and complementary approach in which synaptic
199 plasticity within the recurrent network is de-emphasized and where neural dynamics and
200 reverberations are made central to neuronal processing. The focus is on computations that
201 emerge from neural interactions rather than on computations that are ‘burnt in’ or

202 'programmed' by synaptic modifications. The current research will test the hypothesis that
203 these neural dynamics can be exploited to concatenate overlapping subsequences ('snippets'
204 formed via hippocampal replay) into a coherent sequence. The major questions we will
205 address include:

206 Q1: Is reservoir computing able to allow the concatenation of multiple contiguous
207 subsequences into a coherent sequence, thus addressing a major open question in complex
208 spatial navigation trajectory learning?

209 Q2: How can hippocampal replay be modulated by reward in order to contribute to the TSP
210 behavior? Should the hippocampus model store everything that is being learned on the basis
211 of place field overlap only? Should it be selective to the location of rewards or other task
212 contingencies?

213 Q3: How to optimize neural representation and model memory acquisition at the network
214 level in hippocampus and PFC?

215

216 Integrated Model of Hippocampus Replay and PFC Integration

217 Our approach to answering these questions involves the development of a model of sequence
218 learning using replay in hippocampus and reservoir computing in PFC. The modeling of
219 replay is constrained by experimental data and the full model is tested on simulated robotic
220 platforms. The tests involve navigation between multiple goals (baited cups in rat experiments
221 and rewarded floor landmarks in a robot arena – see Figure 1) to generate a model supporting
222 the current hypothesis while offering predictions that will further prompt modeling
223 refinements and new rodent experiments in relation to replay, hippocampus-PFC functional
224 interactions and spatial sequence learning.

225 *** Insert Figure 2. Hippocampus-PFC model architecture. ***

226 Replay-Driven Model of Spatial Sequence Learning in the Hippocampus-PFC network using 227 Reservoir Computing

228 The computational model consists of 3 main components: Place Representation module (left
229 module in Figure 2), Hippocampus Replay module (top right in Figure 2, and PFC Reservoir
230 for Sequence Learning module (bottom right in Figure 2). The goal of the extended model is

231 to implement online replay-based memory acquisition in the Hippocampus and reservoir-
232 computing based sequence learning in PFC. The Hippocampus-PFC architecture generates
233 snippets of place fields which are used in the PFC reservoir to reconstruct complete spatial
234 sequence trajectories. After successful learning, when primed with a subsequence, the PFC
235 completes that initial trajectory as the system becomes entrained by the input sub-sequence.
236 The main processing steps of the Hippocampus-PFC network model are:

237 Replay-driven snippet formation in Hippocampus. During a trial, the model features
238 artificially generated sharp wave-ripple complexes and awake replay in hippocampus that
239 forms ‘snippets’ of disjoint bouts of trajectories. Reward at baited cups will lead to non-
240 uniform snippet reinforcement, based on propagation of reward value from rewarded sites
241 along the trajectory, thus implementing a spatial credit assignment. The result is that snippets
242 on trajectories that lead to rewards will have an increase probability of being replayed. We
243 predict that some of the most strongly linked snippets will act as hippocampal indexes (Nadel,
244 2000). The place cell layer consists of a 16x16 array of neurons with a homogenous
245 organization and similar field size with overlapping place fields implemented as leaky
246 integrators. SPWR characteristics are based on data from existing literature.

247 Sequence Learning in PFC using reservoir computing. We use a reservoir model of PFC,
248 extending concepts developed in Enel et al. (2016). The inputs are artificially constructed
249 hippocampal ‘snippets’, sequences of place cell activations. The reservoir learns to
250 concatenate the ‘snippets’ in the proper order and identify those snippets that will represent
251 the entire sequence.

252 The PFC will inherently group individual shorter sequences, through replay, to form a full
253 sequence (Fig 3). Specifically, the reservoir model will consist of a population of order 1000
254 leaky integrator neurons with tanh() activation functions, as specified in Enel et al (2016).
255 Based on the vector of activation of place cells from Hippocampus, the PFC will learn a form
256 of identity operation: Given an input sequence segment of place cell activation, it will
257 generate/predict the subsequent sequence of place cell activations. These place cell readout
258 neurons will be used for navigation, and will also be connected to the reservoir units with
259 feedback connections. Feedback connections from these readout units to the reservoir will
260 allow the predicted outputs to serve as inputs when in sequence recall mode. Training will
261 use a modified version of the FORCE learning method developed by Sussillo and Abbott
262 (2009), in which the learning of connection weights between reservoir and readout neurons is

263 based on an online process of weight adjustment that allows for the sampling of the readout
264 error by the system.

265 Feedforward Integration from Hippocampus into PFC.

266 The Hippocampus-PFC network model will be initially simulated by using prerecorded paths
267 of the rat inside the environment as the rat moves from one cup to the next in search of food
268 (Watkins de Jong, et al., 2011). Through propagation of reward during replays between trials,
269 the hippocampus will learn a model of place cell snippet generation based on reward
270 proximity, and then replay small sets of successive snippets so that PFC can learn the
271 transitions and create the global sequence. As trials proceed, the quality of individual
272 snippets (increased density of replay around rewarded targets) will improve due to learning in
273 the Hippocampus. Here we consider an example, based on Figure 2 and 3. After each trial,
274 restSPWRs will lead to replay of several successive snippets (e.g. S5-S6-S7; S2-S3; S3-S4-
275 S5). Though the entire sequence is never replayed, when provided with such subsequences,
276 we predict that the PFC reservoir model will be able to construct a coherent representation of
277 the complete set of snippet transitions, thus defining the overall sequence. Such construction
278 is one of the defining characteristic of reservoirs (see research background section), though it
279 was never tested in this particular spatial navigation context.

280

281 *** Insert Figure 3. Cartoon of snippet formation ***

282 - Because snippets that link rewarded sites by shorter trajectories will be replayed with greater
283 density than those that are on inefficient trajectories far from rewards; we predict that the
284 model will naturally converge towards short sequences (a phenomenon observed
285 experimentally). This will be tested in the model and in the simulated robot.

286 - Because of the creation of snippets during replay, the 'units' of spatial information used by
287 the PFC will be 'chunks' of trajectories, rather than individual place fields. This should speed
288 up the sequence learning process significantly over classical RL on place fields. To test this
289 hypothesis, we will systematically vary the amount of replay. We predict that the performance
290 of the model will correlate with the amount of replay, up to a point where too much replay
291 will yield aberrant snippets, and hence, learning impairments. This will generate experimental
292 predictions that could be tested in the future by optogenetic or pharmacological means of
293 manipulating SPWRs in vivo.

294

295 Sequence Learning Robot Simulation Experiments

296 Real-time experimentation is performed with a robot simulator to test the model under real
297 time conditions similar to that of rats with 5-6 baited cups. As with the rats, a subset (5-6 of
298 21) of the spatial targets in the arena will be baited. In the first trial, the robot will begin an
299 exploration of the arena, following a built-in rule for exploration: proceed to the closest
300 unvisited target (baited or not). During the traversal, place cells will be activated. Their
301 activation will lead to synaptic modification, linking them together if overlapping. When a
302 reward is detected (this will be provided by the control program that will give a reward signal
303 when the robot approaches a baited target), the strength of the connections between the
304 recently visited place cells will be increased through a reinforcement learning process. After
305 the robot has visited all of the rewarded targets, it will enter the inter-trial period. During this
306 period the hippocampus will mostly replay snippets that were reinforced during the previous
307 trial. This replay will be provided as a place cell activation sequence to the PFC model. Over
308 multiple trials, as reinforcement learning proceeds in hippocampus, the formation of snippets
309 that occur during a traversal from one rewarded target to another will increase. Progressively,
310 the inputs to the PFC will consist of concatenated snippets that will form part of the global
311 sequence to be learned. We will show that, whereas the hippocampus can learn local
312 regularities, it cannot bind them into a coherent whole. The PFC model will perform this
313 global linkage of replayed snippets to produce a complete beginning to end sequence.
314 During the active exploration period of successive trials, the PFC will be in a
315 prediction/navigation mode: it will receive place cell activation inputs, and will in real-time
316 predict/generate the next place cell activation. In early trials, before significant learning has
317 occurred, these predictions will not be useful and will only slightly bias the 'nearest neighbor'
318 strategy. As learning proceeds, however, these will become more and more accurate, and will
319 aid the robot to proceed to each baited target, ignoring the unbaited ones. A measure of
320 confidence as a function of the output neuron activation levels will be used to determine when
321 the PFC should contribute to the choice when in conflict with the hippocampus.

322 Methods:

323 The robot simulator is illustrated in Figure 4. It simulates the movement of a rat in a 2x2m
324 square arena. The rat's internal representation of this space is realized by a regular 16x16 grid
325 of place cells on this surface. The navigation module generates a sequence of place cell field

326 activations that begins when the rat is placed in the maze and finishes when the rat has visited
 327 all feeders. PC activations are implemented as Gaussian fields, where the activation of a cell
 328 is strongest at the center and becomes weaker the further away. During each episode the
 329 navigation module: (a) calculates the activation of all place cells (which we call an activation
 330 pattern), (b) records the activation pattern along with the reward received, and (c) performs
 331 navigation actions. After completion of each episode the navigation module sends an input
 332 sequence to the reservoir module consisting of all the PC activation patterns and rewards
 333 received.

334 Insert Figure 5. Place cell coding and Snippet formation.

335 Hippocampal replay based on proximity to rewards

336 Hippocampus replay phenomenon observed during SPW-R complexes in active rest phase is
 337 modeled by a training set made of condensed subsequences of place-cell activation pattern
 338 replayed at random. The distribution that is sampled for drawing a random place-cell
 339 activation pattern might be uniform or modulated by new or rewarding experience as
 340 described in ([Carr et al 2011](#)) : In particular, we model a random replay based on reward. In
 341 ([Ambrose et al 2016](#)) the authors show that during SPW-R sequences place-cell activation
 342 occur in reverse order at the end of a run. We will focus on reward modulated and reverse
 343 replay.

344 We define a snippet as the concatenation of successive place-cell activation pattern:

$$345 \quad S(n; s) = X_{in}(t_{n-s+1} \rightarrow t_n) \quad (1)$$

346

347 Where:

- 348 • s is the number of concatenated place-cell activation pattern.

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

$$351 \quad E(s) = \{S(n; s)\} \quad (2)$$

352

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

355 In **Erreur ! Source du renvoi introuvable.**, Panel B represents a particular trajectory
 356 encountering feeders A, B, C, E and D. The depicted snippet is a subsequence of 5 contiguous
 357 locations belonging to the ABCED sequence. The B and E feeders are baited and marked as

358 rewarding (R_1 and R_2). Panel B shows the spatial extend of a given snippet chosen in
 359 sequence ABCED and panel C shows the place-cell activation pattern of the ABCED
 360 trajectory and the corresponding snippet location in the raster.

361 The snippet replay model favors snippets that are on efficient paths linking rewarded sites
 362 (e.g. paths linking feeders A, B and C in panel B figure 3), and not those that are on
 363 inefficient paths (as in paths linking C, D and E in the same panel). This is achieved by
 364 propagating errors backwards from rewarded locations, and calculating the probability of
 365 replay as a function of proximity to a reward. Panel D of Figure 3 illustrates the resulting
 366 probability distributions for snippet selection along the complete path. Note that the paths
 367 linking A, B and C have the highest probabilities for snippet replay.

368 Hippocampus place-cell replay can occur in forward or backward direction as suggested in
 369 ([Foster & Wilson 2006](#)). We model the reverse replay as follows: For a given trajectory k of
 370 N_k samples, there are $N_k - s$ possible snippets that can be replayed but only a limited number
 371 of snippets will be selected to fit the time budget T . A snippet $S(n)$ have a likelihood of being
 372 replayed if it is related to a reward prediction. A generative model of snippet replay
 373 likelihood is first learnt by propagating a time delayed reward information according to the
 374 replay direction and the snippet duration. The timespan of a snippet acts as a propagation
 375 vector during the estimation phase of the snippet replay likelihood.

376 The reward prediction $V(t_1 \rightarrow t_{N_k})$ is learnt by initializing it to small positive random
 377 values and then iteratively refined by applying the procedure (6) K times:

378 1. Draw a random contiguous time index subset $\tau \equiv T(n, s, r; \beta_{learn})$ according to the
 379 reverse rate β_{learn} :

380 a. Select a time-step t_n such that $n \in \{1 \dots N_k\}$ according to the replay likelihood defined
 381 by:

$$382 \quad P(t_1 \rightarrow t_{N_k}) = \frac{V(t_1 \rightarrow t_{N_k})}{\sum_{i=1}^{N_k} V(t_i)} \quad (3)$$

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

$$386 \quad \tau = \begin{cases} t_{\max(1, n-s+1)} \rightarrow t_n, & r < \beta_{learn} \\ t_n \rightarrow t_{\min(N_k, n+s)}, & r \geq \beta_{learn} \end{cases} \quad (4)$$

388 2. Update the reward estimate V over increasing indices of τ by computing the update
389 equation:

$$390 \quad V(\tau_k) = \alpha(R(\tau_{k-1}) + \gamma V(\tau_{k-1})) + (1 - \alpha)V(\tau_k) \quad (5)$$

391

392 Where:

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

396 (6)

397 It is a convex combination of the current estimate of the reward information $V(\tau_k)$ at the next
398 time step and the instantaneous reward information $R(\tau_{k-1}) + \gamma V(\tau_{k-1})$ based on the
399 previously observed reward signal $R(\tau_{k-1})$ and delayed previous reward estimate $\gamma V(\tau_{k-1})$.
400 Equation **Erreur ! Source du renvoi introuvable.** implements a form of temporal difference
401 learning. It is sufficient to define a coarse reward signal as:

$$402 \quad R(t) = \begin{cases} 1 & \text{if a baited feeder is encountered at time } t \\ 0 & \text{otherwise} \end{cases}$$

403 The snippet generation procedure is simply the repetition of the steps a and b of procedure (6)
404 with $\beta_{generate}$ used instead of β_{learn} until the sum of time subsequences durations overflows
405 a fixed time budget duration.

406

407 *** Insert Figure6 Efficient Sequence Synthesis ***

408

409 Figure 6A illustrates how the replay mechanism in Hipp generates an augmented density of
410 snippets for the efficient subsequences that have been previously experienced. An illustration
411 of how this method yields increased

412

413 Reservoir model of PFC for snippet consolidation

414 We model the prefrontal cortex as a recurrent reservoir network. Reservoir computing refers
415 to a class of recurrent network models with fixed recurrent connections. The reservoir units
416 are driven by external inputs and the network dynamics provides a high dimensional
417 representation of the inputs from which the desired outputs can then be read out by a trained
418 linear combination of the reservoir unit activities. The principle has been co-developed in
419 distinct contexts as the temporal recurrent network ([Dominey 1995](#)), the liquid state machine

420 ([Maass et al 2002](#)), and the echo state network ([Jaeger 2001](#)). The version that we use to
421 model the frontal cortex employs leaky integrator neurons in the recurrent network. At each
422 time-step the network is updated according to the following schema:

423

424 INSERT Figure 7 HERE

425

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

$$429 \quad U_{ffwd}(t_n) = W_{ffwd} * X_{in}(t_n) \quad (7)$$

431 Where:

432 • W_{ffwd} is a fixed connectivity matrix whose values do not depend on time.
433 Synaptic weights are randomly selected at the beginning of the simulation. Various
434 probability density functions (PDF) could be sampled and one condition about on W_{ffwd} is its
435 bijectivity: Every stimulus $X_{in}(t_n)$ must have a distinct image through W_{ffwd} and each
436 U_{ffwd} must correspond to a unique $X_{in}(t_n)$. Practically speaking ([Lukosevicius 2012](#)),
437 sampling $U[-1, 1]$ a uniform distribution is sufficient. A positive synaptic weight in a
438 connectivity matrix models an excitatory connection and a negative weight models an
439 inhibitory connection between two neurons. A synaptic weight equals to zero models no
440 connection between two given neurons. A larger absolute value of the synaptic weights
441 represents a reinforced correlation between firing patterns of those two neurons.

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

$$445 \quad U_{rec}(t_n) = W_{rec} * X_{res}(t_{n-1}) \quad (8)$$

447 Where:

448 • W_{rec} is a N by N square connectivity matrix.

449 Synaptic weights are drawn from a $U[-1, 1]$ uniform distribution, scaled by a $S(N; K) =$
 450 $K \frac{1}{\sqrt{N}}$ factor. The same sign convention as in equation (7) applies for the recurrent connectivity
 451 matrix.

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

$$455 \quad U_{res}(t_n) = U_{ffwd}(t_n) + U_{rec}(t_n) \quad (9)$$

457 The membrane potential of the reservoir's neurons P_{res} then is computed by solving the
 458 following ordinary derivative equation (ODE):

$$459 \quad \tau \frac{\partial P_{res}}{\partial t} = -P_{res}(t_{n-1}) + U_{res}(t_n) \quad (10)$$

461 Where:

- 462 • τ is the neuron's time constant. It models the resistive and capacitive properties of the
 463 neuron's membrane.

464 In this article, we will consider a contiguous assembly of neurons that share the same time
 465 constant. The inverse of the time constant is called the leak rate and is noted h . By choosing
 466 the Euler's forward method for solving equation(10), the membrane potential is computed
 467 recursively by the equation:

$$468 \quad P_{res}(t_n) = h * U_{res}(t_n) + (1 - h) * P_{res}(t_{n-1}) \quad (11)$$

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

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

$$478 \quad X_{res}(t_n) = \sigma_{res}(P_{res}(t_n); \Theta_{res}) \quad (12)$$

480 Where:

- 481 • σ_{res} is the non-linear activation function of the reservoir neurons
- 482 • θ_{res} is a bias that will act as a threshold for the neuron's activation function.

483 We choose a $\sigma_{res} \equiv \tanh$ hyperbolic tangent activation function with a zero bias. Negative
 484 firing rate values represent the inhibitory/excitatory connection type in conjunction with the
 485 sign of the synaptic weight. Only the product of the mean firing rate of the afferent neuron by
 486 its associated synaptic weight is viewed by the leaky integrator neuron.

487 At this point, the reservoir's states carry information about its present and past activation
 488 values within a limited time line of sight. One single reservoir neuron does not contain the
 489 whole information but its activation fluctuations carry partial information about the serial and
 490 temporal structure of the stimulus sequence. This property corresponds to mixed selectivity
 491 ([Rigotti et al 2013](#)).

492

493 Learning in Modifiable PFC Connections to Readout

494 Based on the rich activity patterns in the reservoir, it is possible to decode the reservoir's state
 495 in a supervised manner in order to produce the desired output as a function of the input
 496 sequence. The expected output is only required to be an activation pattern that is temporally
 497 congruent to the input stimulus. This decoding is provided by the readout layer and the matrix
 498 of modifiable synaptic weights linking the reservoir to the readout layer, noted W_{ro} and
 499 represented by dash lines in **Erreur ! Source du renvoi introuvable..**

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

$$501 \quad X_{ro}(t_n) = \sigma_{ro}(W_{ro} * X_{res}(t_n); \theta_{ro}) \quad (13)$$

502
 503 Where:

- 504 • σ_{ro} is the non-linear activation function of the readout neurons
- 505 • θ_{ro} is a bias that will act as a threshold for the neuron's activation function

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

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

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

$$\begin{aligned}
513 \quad W_{ro}(t_n) &= W_{ro}(t_{n-b+1}) + \alpha * X_{res}(t_{n-b+1} \rightarrow t_n) \\
514 \quad &\quad * (X_{ro}(t_{n-b+1} \rightarrow t_n) - X_{des}(t_{n-b+1} \rightarrow t_n)) * (1 - X_{ro}(t_{n-b+1} \rightarrow t_n))^2 \\
515 \quad & \tag{14}
\end{aligned}$$

516 Where:

- 517 • α is a small positive constant called the learning rate
- 518 • $t_{n-b+1} \rightarrow t_n$ is the concatenation of b time steps from t_{n-b+1} to t_n

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

523 Empirically, $b = 32$ gives satisfying results.

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

$$\begin{aligned}
525 \quad X_{des}(t_n) &= X_{in}(t_{n+1}) \\
526 \quad & \tag{15}
\end{aligned}$$

527 The readout is not related to any motor action. It means that we firstly focus on the place-cell
528 firing pattern sequence consolidation. This is the first step towards a biologically plausible
529 model suitable for a navigation task.

530

531 An example of the snippet generation in simulation is illustrated in Figure 6B. There we see
532 for 1000 copies of the model in different starting conditions, each tested 10 times, the
533 autonomous generation of a navigation sequence after training on snippets from the sequences
534 in panels A-C of Figure 1B.

535

536 Integration of the Hipp-PFC model in a real-time sensory-motor loop with rat simulator.

537 The integration architecture is represented in Figure 8. As illustrated the system operates in
538 two modes. In learning mode, the navigation module which includes the rat simulator
539 operates autonomously and follows a spatial sequence link rewarded targets within the arena.
540 The observed sequences of place-cell activations are accumulated, and made available to the
541 replay module (corresponding the the Hipp) which implements the reward propagation
542 method described above. With experience, the Hipp begins to generate snippets with a
543 density distribution that favors shorter paths between rewards. This replay is the contents that
544 is sent to the reservoir for learning.

545 Once learning has taken place, the system can then drive the simulated rat in the environment.

546 Autonomous execution:

547 In this mode, the simulator generates place cell vectors that are input to the reservoir. Based
548 on these inputs, the reservoir is driven into a state trajectory that it has learned, and generates
549 the next place cell activation in the sequence; this is input to the simulator which then moves
550 to that location; the simulator then generates a place cell vector for the new location; this
551 place cell vector is input to the PFC model, and the cycle continues.

552

553 Experiment 1: Demonstration of real-time interaction.

554 The first experiment is used to recreate a path produced by the navigation module. The
555 experiment consists of performing an evaluation episode after each training episode. On the
556 first 4 cycles of the evaluation episode, since the reservoir is not yet initialized, actions are
557 selected directly from the original training path. From the 5th cycle on, an action is selected to
558 navigate to the next position predicted by the reservoir.

559 Experiment 2: Demonstration of learning from Snippets

560 This experiment duplicates experiment 1, but now engages the replay module to generate
561 snippets. The reservoir is thus trained on snippets, rather than on the uninterrupted whole
562 sequence, which was the case in Experiment 1.

563 Experiment 3. Demonstration of autonomous generation of the TSP solution.

564 In this experiment, the navigation module executes three sequences corresponding to those
565 illustrated in Figure 1B, panels B-D. These are sequences ABCED that contains the ABC part
566 of the ABCDE target sequence, BACDE that contains the ABC part of the ABCDE target
567 sequence, and EBCDA that contains the ABC part of the ABCDE target sequence

568 The replay module will then be executed, implementing the reward propagation algorithm,
569 and generating distributions of sequences from the three original sequences, corresponding to
570 those illustrated in Figure 6A.

571

572

573

574 Discussion:

575

576 successful integration of reservoir PFC with simulator, in a real-time closed loop system

577 successful demonstration that this can be used to allow creation of a new efficient sequence in
578 the TSP context.

579 Demonstration that in situations where data generation is difficult (as in robot navigation), the
580 use of replay, and in particular reward-modulated replay, is an effective alternate to
581 generating additional training data.

582

583

584 References:

- 585 Bendor, D., & Wilson, M. A. (2012). Biasing the content of hippocampal replay during sleep.
586 *Nat Neurosci*, 15(10), 1439-1444.
- 587 Carr, M. F., Jadhav, S. P., & Frank, L. M. (2011). Hippocampal replay in the awake state: a
588 potential substrate for memory consolidation and retrieval. *Nat Neurosci*, 14(2), 147-153.
- 589 Carr, M. F., Karlsson, M. P., & Frank, L. M. (2012). Transient slow gamma synchrony
590 underlies hippocampal memory replay. *Neuron*, 75(4), 700-713.
- 591 Davidson, T. J., Kloosterman, F., & Wilson, M. A. (2009). Hippocampal replay of extended
592 experience. *Neuron*, 63(4), 497-507.
- 593 Diba, K., & Buzsaki, G. (2007). Forward and reverse hippocampal place-cell sequences
594 during ripples. *Nat Neurosci*, 10(10), 1241-1242.
- 595 Euston, D. R., Tatsuno, M., & McNaughton, B. L. (2007). Fast-forward playback of recent
596 memory sequences in prefrontal cortex during sleep. *Science*, 318(5853), 1147-1150.
- 597 Foster, D. J., & Wilson, M. A. (2006). Reverse replay of behavioural sequences in
598 hippocampal place cells during the awake state. *Nature*, 440(7084), 680-683.
- 599 Gupta, A. S., van der Meer, M. A., Touretzky, D. S., & Redish, A. D. (2010). Hippocampal
600 replay is not a simple function of experience. *Neuron*, 65(5), 695-705.
- 601 Hasselmo, M. E. (2008). Temporally structured replay of neural activity in a model of
602 entorhinal cortex, hippocampus and postsubiculum. *Eur J Neurosci*, 28(7), 1301-1315.
- 603 Hinaut, X., & Dominey, P. F. (2013). Real-time parallel processing of grammatical structure
604 in the fronto-striatal system: a recurrent network simulation study using reservoir computing.
605 *PLoS One*, 8(2), 1-18.
- 606 Hoffman, K. L., & McNaughton, B. L. (2002). Coordinated reactivation of distributed
607 memory traces in primate neocortex. *Science*, 297(5589), 2070-2073.
- 608 Ji, D., & Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and
609 hippocampus during sleep. *Nat Neurosci*, 10(1), 100-107.

610 Johnson, A., & Redish, A. D. (2005). Hippocampal replay contributes to within session
611 learning in a temporal difference reinforcement learning model. *Neural Netw*, 18(9), 1163-
612 1171.

613 Karlsson, M. P., & Frank, L. M. (2009). Awake replay of remote experiences in the
614 hippocampus. *Nat Neurosci*, 12(7), 913-918.

615 Kudrimoti, H. S., Barnes, C. A., & McNaughton, B. L. (1999). Reactivation of hippocampal
616 cell assemblies: effects of behavioral state, experience, and EEG dynamics. *J Neurosci*,
617 19(10), 4090-4101.

618 Lansink, C. S., Goltstein, P. M., Lankelma, J. V., Joosten, R. N., McNaughton, B. L., &
619 Pennartz, C. M. (2008). Preferential reactivation of motivationally relevant information in the
620 ventral striatum. *J Neurosci*, 28(25), 6372-6382.

621 Lee, A. K., & Wilson, M. A. (2002). Memory of sequential experience in the hippocampus
622 during slow wave sleep. *Neuron*, 36(6), 1183-1194.

623 Marr, D. (1971). Simple memory: a theory for archicortex. *Philos Trans R Soc Lond B Biol*
624 *Sci*, 262(841), 23-81.

625 McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are
626 complementary learning systems in the hippocampus and neocortex: insights from the
627 successes and failures of connectionist models of learning and memory. *Psychol Rev*, 102(3),
628 419-457.

629 Nadasdy, Z., Hirase, H., Czurko, A., Csicsvari, J., & Buzsaki, G. (1999). Replay and time
630 compression of recurring spike sequences in the hippocampus. *J Neurosci*, 19(21), 9497-9507.

631 Pennartz, C. M., Lee, E., Verheul, J., Lipa, P., Barnes, C. A., & McNaughton, B. L. (2004).
632 The ventral striatum in off-line processing: ensemble reactivation during sleep and
633 modulation by hippocampal ripples. *J Neurosci*, 24(29), 6446-6456.

634 Peyrache, A., Khamassi, M., Benchenane, K., Wiener, S. I., & Battaglia, F. P. (2009). Replay
635 of rule-learning related neural patterns in the prefrontal cortex during sleep. *Nat Neurosci*,
636 12(7), 919-926.

637 Popa, D., Duvarci, S., Popescu, A. T., Lena, C., & Pare, D. (2010). Coherent
638 amygdalocortical theta promotes fear memory consolidation during paradoxical sleep. *Proc*
639 *Natl Acad Sci U S A*, 107(14), 6516-6519.

640 Qin, Y. L., McNaughton, B. L., Skaggs, W. E., & Barnes, C. A. (1997). Memory reprocessing
641 in corticocortical and hippocampocortical neuronal ensembles. *Philos Trans R Soc Lond B*
642 *Biol Sci*, 352(1360), 1525-1533.

643 Ribeiro, S., Gervasoni, D., Soares, E. S., Zhou, Y., Lin, S. C., Pantoja, J., et al. (2004). Long-
644 lasting novelty-induced neuronal reverberation during slow-wave sleep in multiple forebrain
645 areas. *PLoS Biol*, 2(1), E24.

646 Schwindel, C. D., & McNaughton, B. L. (2011). Hippocampal-cortical interactions and the
647 dynamics of memory trace reactivation. *Prog Brain Res*, 193, 163-177.

648 Singer, A. C., & Frank, L. M. (2009). Rewarded outcomes enhance reactivation of experience
649 in the hippocampus. *Neuron*, 64(6), 910-921.

650 Stickgold, R., & Walker, M. P. (2007). Sleep-dependent memory consolidation and
651 reconsolidation. *Sleep Med*, 8(4), 331-343.

652 Sutherland, G. R., & McNaughton, B. (2000). Memory trace reactivation in hippocampal and
653 neocortical neuronal ensembles. *Curr Opin Neurobiol*, 10(2), 180-186.

654 Tatsuno, M., Lipa, P., & McNaughton, B. L. (2006). Methodological considerations on the
655 use of template matching to study long-lasting memory trace replay. *J Neurosci*, 26(42),
656 10727-10742.

657 Watkins de Jong, L. W., Gereke, B., Martin, G. M., & Fellous, J. M. (2011). The traveling
658 salesrat: insights into the dynamics of efficient spatial navigation in the rodent. *J Neural Eng*,
659 8(6), 065010.

660 Wilson, M. A., & McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code
661 for space. *Science*, 261(5124), 1055-1058.

662

663

664

665

666

667 Figures

668

669

670

671

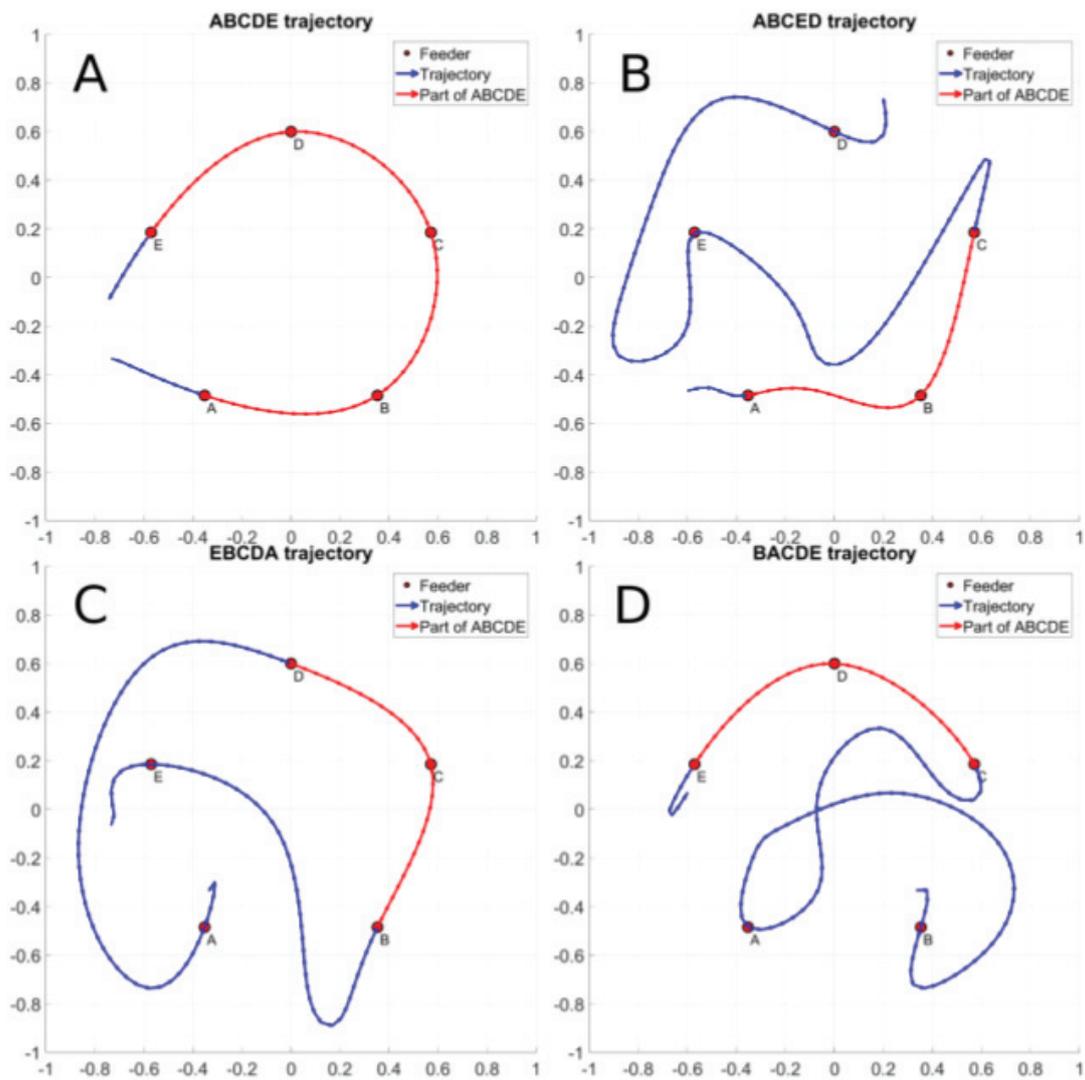


672

673 Figure 1: Rodent and robot experiments. (Left) Behavioral apparatus where a rat (no
674 hyperdrive) is shown wearing a reflective jacket for tracking purposes on the open-field with
675 21 cups (Watkins de Jong et al. 2011). (Center) Schematic representation of the sample final
676 path chosen by the animal after 10 trials. In this spatial sequence, place fields A, B and C are
677 expected to reactivate while place field F is not.

678

679



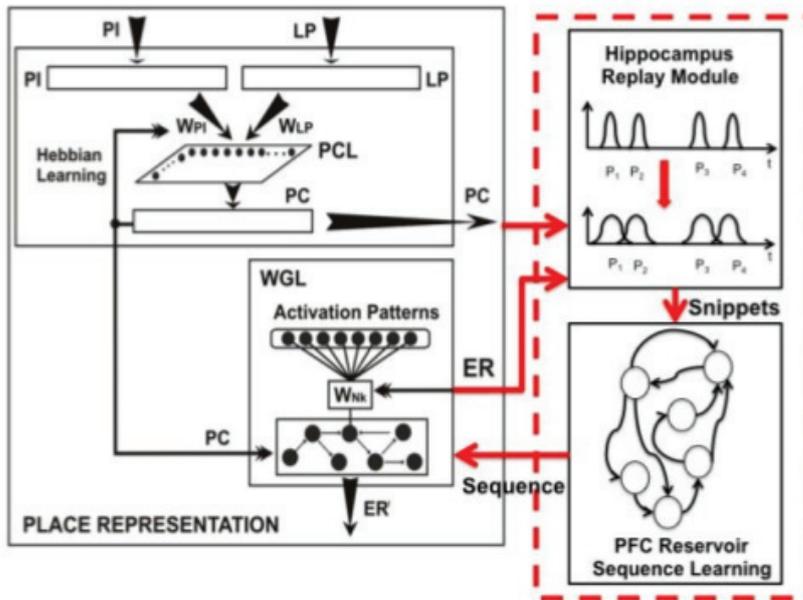
680

681

682 Figure 1B. TSP behavior

683

684



685

686 Figure 2. Hippocampus-PFC model architecture. (Left) Modified Place Representation module
 687 consisting of World Graph Layer (WGL) receiving activation from Place Cells (PC) in the
 688 hippocampus. (Right) The new modules are shown in the red box consisting of Hippocampus Replay
 689 and PFC Reservoir for Sequence Learning. The Replay module will strengthen place cell firing
 690 between spatially adjacent place fields thus generating snippets of place cell firing as shown for place
 691 cells P1P2 and P3P4 respectively. The proposed PFC Reservoir architecture will receive a sequence of
 692 Place Cell activations or snippets to construct full trajectory sequences. Snippet generation is based on
 693 proximity to reward. Abbreviations: ER: Expectation of Max Reward, WPI: Path Integration
 694 connection Weight, WLP: Landmark Pattern connection Weight, PI: Path Integration, LP: Landmark
 695 Pattern, PCL: Place Cell Layer)

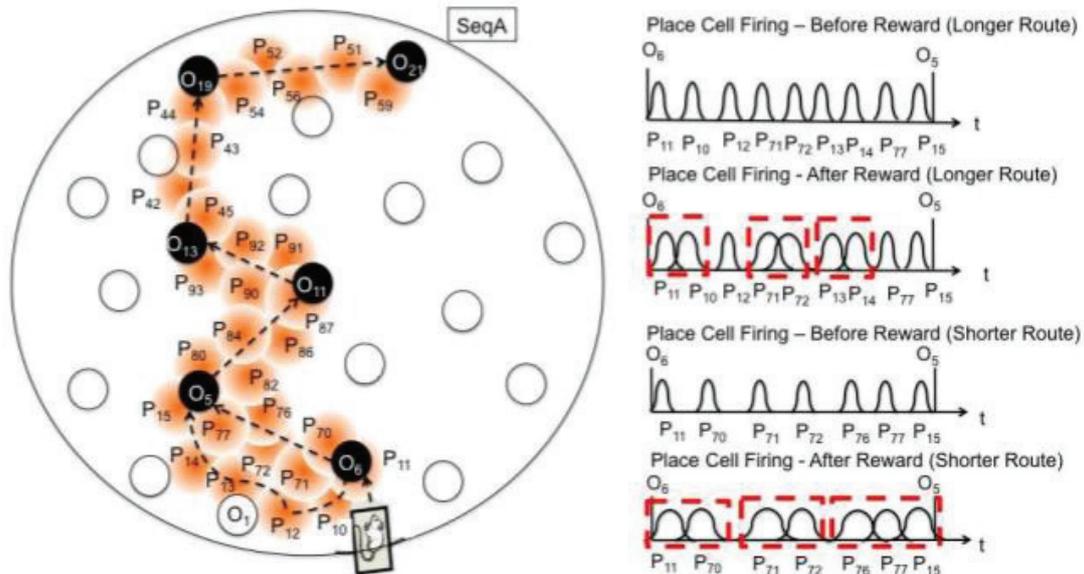
696

697

698

699

700



701

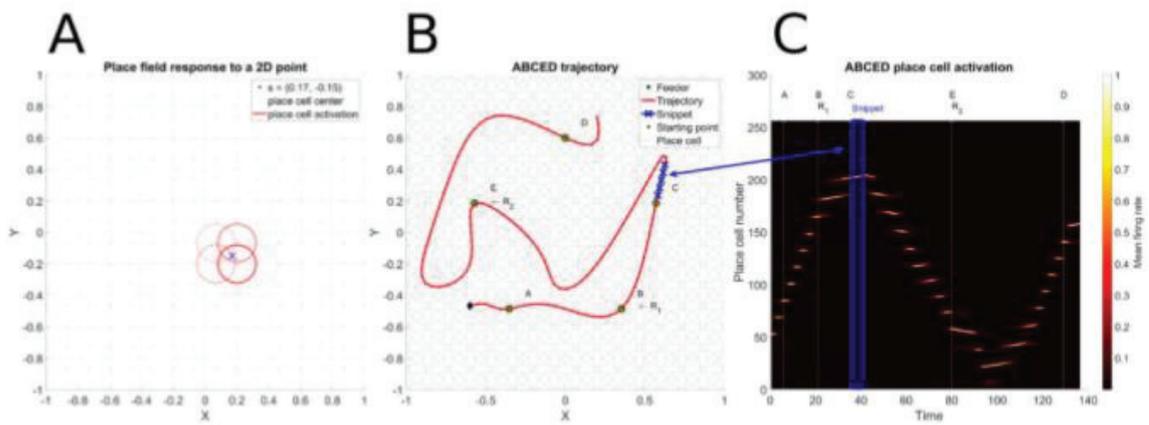
702

703 Figure 3: Cartoon of snippet formation. (Left) Rodent and robot experiments on an open field with
704 white (non baited), and black (baited during SeqA) circles corresponding to 21 cups named Oi.
705 Colored areas labeled Pi correspond to place fields. The complete learned trajectory SeqA of the rat is
706 O6-O5-O11-O13-O19-O21. (Right) During initial trials rodent usually follows “longer routes”
707 between cups, for example between O6 to O5 a possible firing sequence is shown on the top right:
708 P11-P10-P12-P71-P72-P13-P14-P77-P15. As the rat reaches O5 the hippocampus will replay
709 (acqSPWR) the place cell firing sequence just activated generating snippets (in red boxes) as
710 exemplified by the next graph. As the rodent learns the shorter route between cups, place cell firing
711 sequences may get shorter, with the following sequence: P11-P70-P71-P72-P76-P77-P15. Again as
712 the rat reaches O5 the hippocampus will replay the place cell firing sequence just activated generating
713 snippets (in red boxes, bottom graph). It can be observed that two different trajectories followed by the
714 rodent, even if involving similar cups, may yield different snippet formations, possibly involving some
715 of the same place fields. The graph shows that ‘snippets’ are formed by increasing the synaptic
716 weights between subsets of place cells that reactivate together (illustrated here as an overlap in the
717 activity curves).

718

719

738
739
740
741
742
743
744
745
746

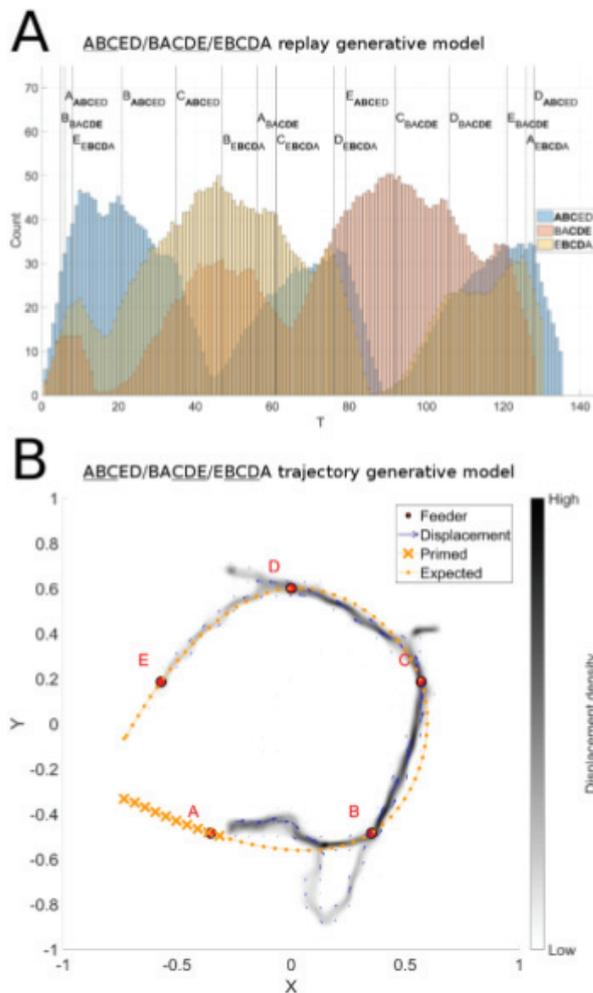


747
748
749
750
751
752
753
754
755
756

Figure 5. Place coding, snippet formation and replay.

757

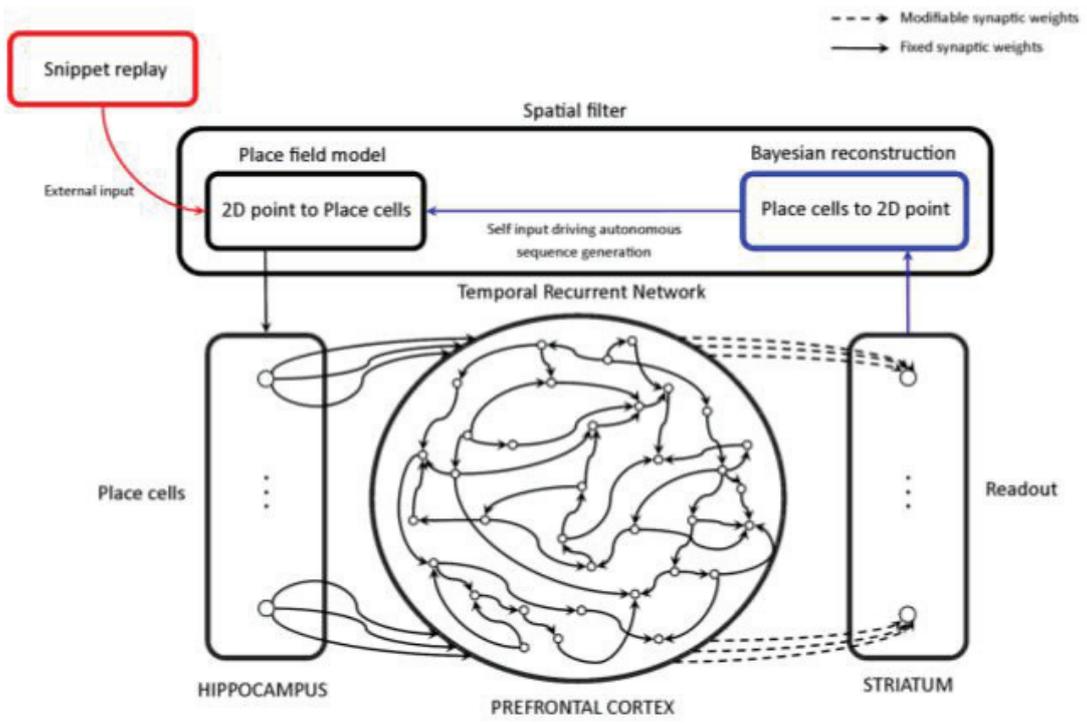
758



759

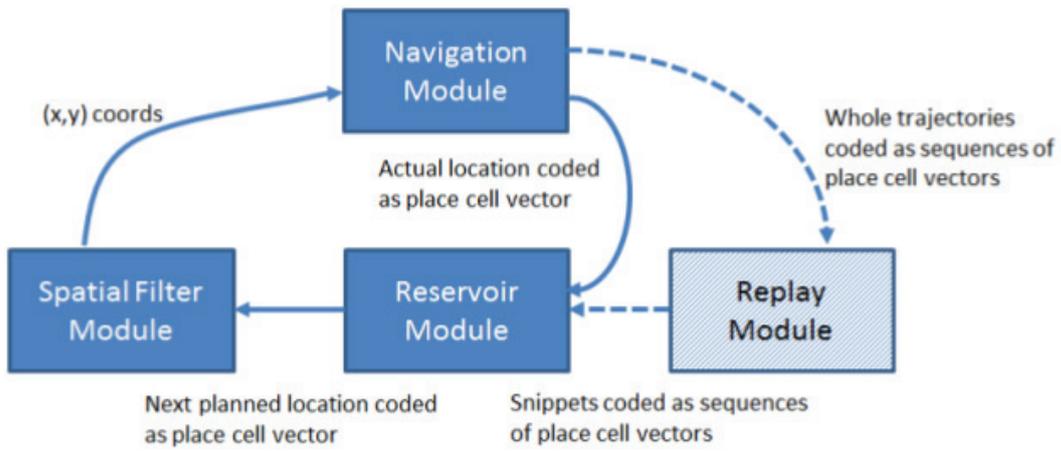
760 Figure 6. Efficient Sequence Synthesis. A. Distribution of snippets drawn from the sequences
761 illustrated in Figure 1 B, C and D. Globally we observe snippet selection favors snippets from the
762 beginning of sequence ABCED (blue), the middle of EBCDA (yellow), and the end of sequence
763 BACDE (pink), which corresponds exactly to the efficient subsequences (ABC, BDC, and CDE) of
764 these three sequences. This distribution of snippets is used to train the model. The results of the
765 training are illustrated in panel B. Here we see a 2D histogram of sequences generated by the model
766 in the ABCDE recombination experiment. 10 batches of 100 reservoirs each were trained and each
767 model instance was evaluated 10 times with noise. These results are very robust and satisfying,
768 demonstrating that our hypothesis for efficient sequence discovery based on reward-modulated replay
769 is validated.

770
771
772
773



774
775
776
777
778
779
780

Figure 7. PFC reservoir model



781

782

783 Figure 8. Integration architecture

2.6. Neurodynamic properties of the semantic network.

Title: Beyond the word and image: III. Neurodynamic properties of the semantic network

Authors:

Anne-Lise Jouen, Nicolas Cazin, Sullivan Hidot, Carol Madden, Jocelyne Ventre-Dominey, Peter Ford Dominey

Author Affiliations:

INSERM Stem Cell and Brain Research Institute U846, Bron, France;

***Corresponding author:** Peter Ford Dominey, INSERM Stem Cell and Brain Research Institute, U846, 18 Avenue Doyen Lepine, 69500-Bron, France.

Tel: 33 (0)4 72 91 34 84 -Email: peter-dominey@inserm.fr

Abstract

Understanding the neural process underlying the comprehension of visual images and sentences remains a major open challenge in cognitive neuroscience. We previously demonstrated with fMRI and DTI that comprehension of visual images and sentences describing human activities recruits a common semantic system. The current research tests the hypothesis that this common semantic system will display similar neural dynamics during processing in these two modalities. To investigate these neural dynamics we recorded EEG from naïve subjects as they saw simple narratives made up of a visual image (or a sentence) depicting a human event, followed by a second image (or sentence – always in the same modality) that was either a sequentially coherent narrative follow-up, or not, of the first image (or sentence). Analysis of the EEG signal revealed common neural dynamics for semantic processing across modalities. Late positive ERPs were observed in response to sequential incoherence for sentences and images, consistent with previous studies that examined coherence in these two modalities separately. Analysis of oscillatory power revealed increased gamma-band activity for sequential coherence, again consistent with previous studies showing gamma increases for coherence and matching in sentence and image processing. Multivariate analysis demonstrated that training on data from one modality (images or sentences) allowed reliable decoding of the sequential coherence of data from trials in the untrained modality, providing further support for a common underlying semantic system for images and sentences. Processing sequential coherence of successive stimuli is associated with neural dynamics that are common to sentence and visual image modalities and that can be decoded across modalities. These results are discussed in the context of EEG signatures of narrative processing and meaning, and more general neural mechanisms for structure processing.

Introduction:

A major function of higher cognitive processing is making sense of the world around us based on accumulated experience that is organized in a narrative structure (Bruner 1990). Binder et al (2009) note that knowledge acquired from experience underlies our ability to understand, and forms the basis of the semantic system. In a meta-study of 120 PET and fMRI studies that access meaning from words, they identified a widely distributed network that suggests that semantic representations tap into sensory, motor, affective and cognitive systems recruited in human experience.

We hypothesize that such a broadly distributed semantic coding is not restricted only to verbal material, but rather that there will be a common network for representing meaning issued from verbal or visual image input. In our first test of this hypothesis, we determined whether a common semantic network would be recruited in the comprehension of visual images, and sentences that depict human events (Jouen et al 2015). fMRI and DTI revealed the spatial organization of this network and aspects of its connectivity. Interestingly, the common network was quite similar to that identified by Binder et al (2009), including major activation in the angular gyrus and temporo-parietal cortex. This was inspired by the groundbreaking work of Vandenberghe et al (1996) in understanding the common semantic system, and extended their approach from simple images and single words, to rich images and full sentences. In our second investigation of this hypothesis we performed detailed mapping of the white matter pathways and functional connectivity linking the cortical nodes of this distributed network, with major hubs in the anterior temporal cortex and the temporo-parietal cortex (Jouen et al in revision, Jouen et al 2012). The dense connectivity of this area highlights its role in integrative processes. Such a densely connected area could serve as an anchor point for the convergence of multimodal cortical representations during experience. During comprehension, activation of such a convergence zone could then allow for divergent reactivation of multimodal areas in reconstructing the meaning (Lallec & Dominey 2013, Meyer & Damasio 2009). While fMRI and DTI thus provide a view of the spatially distributed network organization of the semantic system and its connectivity that will be engaged in making meaning from narrative, it is less optimal for characterization of the temporal dynamics of this integrative processing. Given its temporal precision, EEG is a more suitable tool for investigating the temporal unfolding of processes in narrative integration. We thus set out to compare the temporal dynamics of the EEG signal in visual

image and sentence processing using short (two-element) narratives that allowed us to manipulate the sequential coherence of the successive stimuli within the narratives.

Written and visual narratives have been investigated using EEG in protocols similar to ours where different dimensions of narrative coherence has been manipulated. Late ERP positivities with some variability in their localization and timing are frequently observed in these coherence manipulations. This can be seen during violation of the expectation or goal in visual narrative (Cohn et al 2014, Sitnikova et al 2008), and violation of semantic expectations built up earlier in a text in verbal narrative (Bornkessel-Schlesewsky & Schlewsky 2008, Brouwer et al 2012, Paczynski & Kuperberg 2012, Xiang & Kuperberg 2015). While there are a variety of experimental manipulations that can modulate late positivities, there is converging evidence that these responses can be associated with aspects of semantic incoherence (Brouwer et al 2012). These results from verbal and visual image narrative meaning processing indicate that encountering unfulfilled expectations or predictions can invoke on-line processes that are often revealed by late ERP positivities.

Such aspects of cognitive processing can also be characterized in terms of oscillatory power in neuronal activity. EEG gamma-band activity has been associated with semantic processing in language (Hagoort et al 2004, Hald et al 2006) and in multisensory semantic matching (Schneider et al 2008). The gamma-band activity appears to increase in situations of semantic congruence. Thus, gamma band responses were observed to increase in responses to semantically correct vs. violation sentences (Hald et al 2006). Likewise, gamma-band responses increased in response to semantically matching vs mismatching stimuli (Schneider et al 2008) in a multisensory semantic priming task using visual and auditory stimuli. We can thus predict that gamma-band responses would be increased in conditions of increased coherence between successive stimuli.

While the demonstration of related ERP and time-frequency effects for image and sentence processing would argue for a common underlying mechanism, demonstration of cross-modal decoding (i.e. training a decoder on image data and decoding sentence data, and vice-versa) would further argue that the neural dynamics in the two cases share a common component. The EEG signal is sensitive to diverse cognitive functions and has been used to decode cognitive states. For example, attentional focus in an auditory “cocktail party” context can be decoded from single trial EEG (O’Sullivan et al 2014), and EEG has been used to discriminate which one of seven different musical fragments a subject is listening to (Schaefer et al 2011). This type of decoding should be suitable to test our hypothesis of the existence of

common neurophysiological processes for the semantic processing of visual and written narratives.

In this context, the goal of the current research is to determine whether common neural processes will be invoked in the treatment of stimuli that are unexpected based on narrative context established in verbal and visual image input modalities. Specifically, using EEG we set out to determine if we would observe a common temporal pattern of activity in simple visual and verbal narrative processing, by examining ERPs, oscillatory activity, and cross-modal decoding. We adapt the protocol from Jouen et al (2015) to the EEG domain, retaining the overall structure of the protocol that involved the presentation of natural visual images depicting human activity, and in separate blocks of trials, short whole sentences describing the same type of human activity. For EEG we introduce an aspect of temporal succession corresponding to simple narratives. In both the sentence and the image domains, two successive stimuli are presented, separated by a variable delay. The second stimulus can be sequentially coherent or incoherent in a narrative context with respect to the first. Brain responses to these stimuli are analysed to determine if there is evidence for a common processing of sequential coherence in sentences and images.

Methods:

Participants: Eighteen healthy right-handed volunteers participated in the experiment (10 females, 8 males, native French speakers, without prior neurological history, 25.5 ± 4.1 years). The study was conducted in accordance with the Declaration of Helsinki, and all participants were advised of the physical details of the experiment, and gave their informed written consent to participate in the experiment.

***** Figure 1 About Here *****

Stimuli:

The paradigm is presented in Figure 1. For the image conditions one-hundred and twenty (120) image pairs were selected from the Getty photo database (<http://www.gettyimages.fr/>). Sixty pairs of images made up sequentially coherent narratives, and a second set of sixty images pairs made up sequentially incoherent narratives. For the sentence conditions, sixty sentence pairs were created that made up sequentially coherent narratives, and sixty pairs for incoherent narratives. In the same way that images can implicate the same people in different situations over the presentation of two successive

stimuli, the sentences describing people performing an action can follow-up (or not) in the second sentence. To reproduce a similar effect as with the images, that is, that certain information does not need to be re-analysed (e.g. the identity of the people, their different roles in the relations that link them) we chose to use a complete sentence for stimulus 1 (e.g. “The man and the woman looked at the movie announcements”) and only pronouns to refer to the same protagonists in the stimulus 2 (e.g. “They bought tickets at the counter.”). Example sentences and images are illustrated in Figure 1. The images were controlled and counterbalanced for the number of people illustrated (1 or 2). Additional sample sentences are:

Sequentially Coherent: S1- The man opens the driver side door. S2 - He starts the car.

Sequentially Incoherent: S1 - The woman lounges by the pool. S2 - She irons the laundry.

Experimental Paradigm:

Subjects were seated in front of a visual display. Visual stimuli (sentences or images) were presented at the center of the screen, subtending a visual angle of approximately 5°. Subjects saw a visual image (or read a sentence) depicting a human event, and after a pause saw a second image (or sentence) that was either sequentially coherent, or not, with the first image (or sentence). Stimulus one and two were always in the same modality (sentence or visual image). The stimuli were presented for 2 seconds, separated by a delay of 1-1.5 seconds. Trials were blocked by modality (image or sentence), two blocks per modality, for a total of 4 blocks. Each block had 30 coherent, and 30 incoherent trials. Two thirds of the trials were followed by probe questions in order to maintain vigilance. Subjects made responses with their right hand.

EEG acquisition and preprocessing:

We acquired continuous neural activity with 64 channel EEG (Biosemi, ActiveTwo, version 5.36) sampled at 2Khz while subjects performed the task. EEG data was processed using EEGLAB. Preprocessing was performed with the FASTER plugin for EEGLAB (Nolan, Whelan et al. 2010), with bandpass filter from 1 Hz to 95 Hz, notch filter (50 Hz), artifact rejection and epoching from -0.5 to 3 s relative to the stimulus onset. All electrodes were referenced with respect to the mastoids. Artifacts from eye movements, blinks and temporal muscle activity were identified and removed using a second-order blind

identification (SOBI) algorithm implemented in the EEGLAB toolbox interface (Lio and Boulinguez 2013).

EEG Analysis

The EEG signal was analysed in terms of task related ERPs, time frequency analysis, and cross-modal (sentence vs. image) decoding approaches.

a. ERP analysis: Average waveforms were computed across all trials per condition, using a baseline subtraction with the interval [-200 0 ms]. Repeated measures ANOVAs and Wilcoxon non-parametric tests were performed using MATLAB, on the factors: Modality (Image, Sentence), Sequential Coherence (Sequential, Non-Sequential) and Order (First, Second stimulus) for different time windows. Holm's method was used to correct for multiple comparisons (Holm 1979). The analysis was performed for time windows including 250-350ms corresponding to the P300, 500-625ms corresponding to the N400, 750-1000ms corresponding to the P600, and a later positivity at 1600-1800ms.

b. Near Gamma Band (NGB 35-45 Hz) Time-frequency analysis: Time-frequency (TF) event-related spectral perturbation (ERSP) representations were obtained by computing the mean squared norm of the convolution of complex Morlet wavelets with the EEG data, across the trials per condition. We used wavelets with a 3-cycle width and frequencies ranging from 1 to 80 Hz, at 1 Hz intervals. We performed a single-trial baseline correction, dividing the ERSP power at each time point by the average spectral power in the pre-stimulus baseline period at the same frequency before averaging and taking a log-transformed measure of the percentage of ERSP as described in Grandchamp and Delorme (2011).

Repeated measures ANOVAs and Wilcoxon non-parametric tests were performed using MATLAB, on the factors: Modality (Image, Sentence), Sequential Coherence (Sequential, Non-Sequential) and Order (First, Second stimulus) on the ERSP in the near-gamma band (35-45 Hz) for sliding time widows of 200 ms. Holm's method was used to correct for multiple comparisons (Holm 1979).

c. Cross-modal decoding: Principal component analysis (PCA) was performed, using Matlab, on the signals from the 64 electrodes over all conditions of sequential coherence (Sequential, Non-Sequential), modalities (Image, Sentence) and stimulus order (first and second) confounded, starting from 500ms prior to stimulus onset, to 1000 ms post offset. We then separated data from sequential vs non-sequential, image vs. sentence, and first vs. second

stimulus trials, and analysed this data in the resulting PCA space, to determine if the principal components were sensitive to these task factors, particularly the sequential coherence.

We then performed a cross-modal validation. That is, we performed PCA on data from image processing and determined whether the resulting principal components could be used to discriminate sequential coherence in the sentence data (and vice versa).

***** Figure 2 About Here *****

Results

EEG Analysis

Before the detailed analysis, we first provide an overall view of the effects of stimulus order. Figure 2 illustrates an example of the effects Order (First, Second stimulus) on ERP responses to Modality (Image, Sentence) and Sequential Coherence (Sequential, Non-Sequential) in a representative frontal central electrode FC1. Note that during extended periods between the stimulus onset and offset (e.g. 750-1000 ms) the modality and sequential coherence effects are absent for stimulus 1, and then become pronounced for stimulus 2. This is supported by the ANOVA. During the 750-1000 ms window there are significant effects for Order [$F(1,17) = 6.0, p < 0.05$], Modality [$F(1,17) = 24.4, p < 0.001$], and Sequential Coherence [$F(1,17) = 10.8, p < 0.01$]. Interestingly there is an Order * Modality interaction [$F(1,17) = 11.83, p < 0.01$], and an Order * Sequential Coherence interaction [$F(1,17) = 16.3, p < 0.001$]. Overall, all 64/64 electrodes demonstrated a significant main effect for Modality, 55/64 for Stimulus Order, and 31 for Sequential Coherence. The Modality * Order interaction was observed in 45/64, and the Sequential Coherence * Order interaction in 24/64 electrodes. This indicates rich effects that are highly influenced by the factors manipulated in the task, and that the effects of coherence emerge, as expected only with the second stimulus. We now provide a more specific analysis of these effects.

ERP results are presented for four time periods corresponding to the P300, N400, and two late positivities. The first of these late positivities is in the P600 range (750-1000ms), and the second is a later positivity (1600-1800ms). Recall that the coherence of a given pair of stimuli can only be discerned by the subject after the presentation of stimulus 2. Thus, in all statistical analyses, our selection requirement is for electrodes that display the desired effect for Stimulus 2, and not for Stimulus 1. That is, in all ERP results reported, NonSEQ – Seq is not significantly different with $p \geq 0.05$ for stimulus 1, while for stimulus 2, NonSEQ – Seq

is different at $p < 0.05$, for sentences and images, as revealed by ANOVA and Wilcoxon post-hoc tests.

***** Figure 3 About Here *****

***** Table 1 About Here *****

Early positivity (P300)

Surface maps and ERP plots are presented in Figure 3, displaying the sequential coherence contrast (NonSEQ – SEQ) for both sentences and images, as well as the modality contrast (sentences vs. images) in the 250-350 timeframe. An extend group of frontal central electrodes appear to make both coherence and modality discriminations in this timeframe. These observations are confirmed by ANOVA and Wilcoxon pot-hoc tests. Multiple central electrodes met the criteria of NonSEQ – SEQ at $p \geq 0.05$ for Stim 1, and $p < 0.05$ for Stim 2 (see Table 1). Illustrative frontal central electrode C1-12 displays a significant effects for Modality [$F(1,17) = 9.4, p < 0.01$], and Sequential Coherence [$F(1,17) = 16.6, p < 0.001$], and a significant Sequential Coherence * Order interaction [$F(1,17) = 12.5, p < 0.01$], indicating that indeed the coherence effect is dependent on whether the subject is exposed to the first or second stimulus. All of the identified electrodes display a significant Sequential Coherence * Order interaction (see Table 1).

***** Figure 4 About Here *****

N400

Surface maps and ERP plots are presented in Figure 4 displaying the sequential coherence contrast for both sentences and images in the 500-625ms time frame. We can observe that a sequential coherence discrimination with a frontal negativity is made for the second stimulus, in a number of centrally located electrodes, but only for the Image and not the Sentence modality. These observations are confirmed by ANOVA and Wilcoxon post-hoc tests. Multiple electrodes (AF7-2, AF3-3, F3-5, F5-6, AF4-36, F6-41) partially responded to the selection criteria, that is, they reliably distinguished sequential coherence for the second stimuli, but only for images. Illustrative frontal electrode AF7-2 displays a

significant effects for Modality [$F(1,17) = 7.0, p < 0.01$], a significant Sequential Coherence * Order interaction [$F(1,17) = 5.25, p < 0.01$].

***** Figure 5 About Here *****

***** Table 2 About Here *****

Late Positivities (750 – 1000 ms) and (1600 – 1800 ms)

Figure 5 illustrates ERP responses and surface maps for the sequential coherence contrast for sentences and images in the 750 – 1000 and 1600 – 1800 ms time frames. We can see similar frontal positivities for the Image and Sentence modalities for both of the late time windows. These observations are confirmed by ANOVA and Wilcoxon post-hoc tests. Multiple spatially contiguous electrodes met the criteria of NonSEQ – SEQ at $p \geq 0.05$ for Stim 1, and $p < 0.05$ for Stim 2 (see Table 2). Illustrative frontal central electrode FC1 displays a significant effects for Modality [$F(1,17) = 24.4, p < 0.001$], Sequential Coherence [$F(1,17) = 10.8, p < 0.01$], and Order [$F(1,17) = 6.0, p < 0.05$], and a significant Sequential Coherence * Order interaction [$F(1,17) = 12.5, p < 0.01$], indicating that indeed the coherence effect is dependent on whether the subject is exposed to the first or second stimulus. Wilcoxon post-hoc tests confirmed that only for stimulus 2 there was a significant sequential coherence contrast for both modalities (see Table 2). All but one of these electrodes displays the sequential coherence by order interaction. These two late positivities thus reliably distinguish the sequential vs non sequential coherence contrast for the second stimuli in both sentence and image modalities. Interestingly, as indicated in Table 2, they also make the modality distinction.

***** Figure 6 About Here *****

***** Table 3 About Here *****

Near-Gamma Time-Frequency Analysis

According to our criteria of no sequential coherence effect for the first stimulus and a significant effect both for images and sentences for the second stimulus, a group of left frontal electrodes was identified (see Table 3). Figure 6 illustrates the TF representation for an

example of one of these electrodes, FC5-9. In the period 1200-1400ms, near-gamma activity (35-45 Hz) appears to be greater for sequential vs non-sequential stimuli, only for the second stimulus. This is confirmed by ANOVA. In the period 1200-1400 there are significant effects for Modality [$F(1,17) = 5.3, p < 0.05$], and Sequential Coherence [$F(1,17) = 4.6, p < 0.05$], a significant Modality * Stimulus Order interaction [$F(1,17) = 13.0, p < 0.01$], and a significant Sequential Coherence * Stimulus Order interaction [$F(1,17) = 9.1, p < 0.01$]. This interaction means that the sequential coherence response depends on the stimulus order. Post-hoc tests reveal that in all cases, near-gamma power increases in the sequentially coherent condition for stimulus 2. Summary statistics for all significant effects are provided in Table 3.

***** Figure 7 About Here *****

***** Table 4 About Here *****

Cross-modal decoding:

In order to determine the principal components accounting for variability in the EEG signal, we first performed PCA on data from sentences and images. As illustrated in Figure 7 the first two components of the PCA appear to represent the two predominant dimensions in the data which are, respectively, the Modality of the stimuli (sentences vs. pictures), and the Sequential Coherence of stimulus 2. Wilcoxon tests on the sequential coherence differences reveal that PC1 reliably makes the coherence distinction for Sentence 2 in the Late 2 period (1600-1800ms), and PC2 reliably makes the coherence distinction for Image 2 in the Late 1 (750-1000ms) and late 2 periods (see Table 4). In all comparisons the coherence distinction is never significant for the first stimuli. This indicates that the PCA is sensitive to the distinction between coherent and incoherent stimuli for the key second stimulus.

Cross-Modal Validation:

PCA was performed on data from image trials, and then the resulting principal components were used to decode the sequential coherence using data from sentence trials. The symmetrical decoding operation (PCA on sentence trials, decoding image trials) was then performed. Tables 5 and 6 display the Wilcoxon statistics for the sequentially incoherent/coherent distinction in this cross-modal validation, for the first, and second stimuli, respectively. Recall that when the first image or sentence is presented, the subject cannot yet determine if it is coherent or incoherent, and so we predict no decoding can be made on the

data from stimulus 1. Indeed, this is confirmed in Table 5. No coherent vs incoherent differences are significant for stimulus 1. We see that in Table 6, the data for stimulus 2 reveals a different story. When the PCA was performed on the image data, the first component reliably distinguished between sequential coherence for images, in the two late periods. More importantly, this component also reliably made the coherence distinction for the sentence trials, in both late periods. In the complementary sense, PCA performed on sentence data could then reliably make the coherence distinction both for sentences and images. This argues that the processing of sequential coherence has common neural substrates for images and sentences.

Discussion:

In the current research we analyze neural activity generated while subjects are exposed to complex human-activity-related stimuli presented in two-event narrative sequences. Our objective is to test the hypothesis that common neural processes are involved in making sense of these narrative sequences in the sentence and image conditions. We previously performed a related experiment where we recorded brain activation using fMRI while subjects were involved in understanding single visual images and written sentences. This revealed a network of brain regions common to processing sentences and images suggesting that the understanding process involves mapping the understood situation into one's own experiential representation of meaning (Jouen et al 2015). This is coherent with the semantic network identified by Binder et al (2009), and extends this into the visual image domain. Here we examine the neural dynamics associated with engaging the semantic system as required for understanding two-element narratives made up of images or sentences.

Already by 300 ms. we observe two effects – a modality distinction, between sentences and images, and interestingly, for both sentence and image modalities we also observe the sequential coherence effect. Such early responses have been observed in language processing for syntactic class expectation violations (Neville et al 1991), and in image processing in response to comic strip panels in which the motion lines were in conflict with the depicted motion, or when they were absent (Cohn & Maher 2015). This suggests that this early positivity is a response to a violation of an expectation with respect to the narrative structure linking the two successive stimuli. Interestingly we observed an N400 in response to the incoherent images, but not sentences. Cohn et al (2012) similarly observed N400 effects in response to image sequences that violated narrative structure and semantic structure, and

Paczynski and Kuperberg (2012), note different combinations of stimuli can produce late positive responses with and without the N400.

Late Positive Response to Narrative Incoherence

The main result is the reliable observation of late centrally distributed positive ERP responses during the processing of sentences and images in the sequentially incoherent condition. The late positivity that we observe in the 750-1000 ms timeframe is similar to those that have been observed when sentences and images do not fit with the previously established context (Bornkessel-Schlesewsky & Schlewsky 2008, Brouwer et al 2012, Kuperberg 2007, Xiang & Kuperberg 2015). While it is beyond the scope of the current analysis to retrace the details of the evolution of the P600 from its syntactic processing origins (Osterhout & Holcomb 1992) to its more recent manifestation in the complex world at the interface of syntax and semantics, we can attempt to situate our finding of a late frontal positivity that is associated with a form of making meaning both for text and visual narratives.

Thus, while frontal positivities were initially produced in response to syntactic anomalies, they have been identified in a number of situations that do not involve syntactic recovery. For example Kaan and Swaab (2003) observe a late (500-900 ms) frontal positivity associated with ambiguity resolution and/or increases in discourse complexity. Kuperberg (2007) reviews studies where late positivities are evoked, with and without syntactical anomalous sentences. Of particular interest, in the sentence “Every morning at breakfast the eggs would eat ...”, the response to eggs (which is semantically anomalous as inanimate eggs cannot eat) was a robust P600, in the absence of an N400 (Kuperberg et al 2003). Kuperberg suggests that some degree of semantic association between a verb and its arguments may trigger a P600. Similarly, Xiang and Kuperberg (2015) have argued that a late posterior positivity component is triggered when a near certain prediction is followed by an input that requires a switch to a new generative model representing relationships between events. Paczynski and Kuperberg (2012) advocate the P600 as reflecting a conflict between semantic memory-based predictions, and the detection of propositional incoherence. Brouwer et al (2012) consider that late positivities are invoked by semantic integration processes. Our sequentially incoherent condition would thus invoke such integration processes producing a prediction error. We can suggest that this type of situation can be established in a narrative context, so that while the second sentence contains no inherent violation, it is anomalous with respect to the narrative context established by the first sentence. This is similar to the discourse effects observed by van Berkum and Hagoort (Hagoort & van Berkum 2007, van

Berkum et al 2003, Van Berkum et al 2005, Van Berkum et al 1999), where an N400 can be evoked by an otherwise neutral sentence that has been rendered anomalous by preceding context.

While the actual cognitive processing that is associated with these late positivities remains to be clarified, we can conclude that sequential incoherence in short narratives reliably evokes late positivities with similar timing and topography for sentences and images. This common ERP profile is evidence for common underlying neurophysiological processing.

Time-Frequency Analysis

Gamma band activity provides further evidence. Gamma band activity is associated with a number of higher cognitive functions including controlling and organizing memory traces (Keizer et al 2010), and semantic processing in sentence comprehension (Hagoort et al 2004, Hald et al 2006). We observed increases of near-gamma band activity (35-45 Hz) for sequentially coherent stimuli, both for sentence and image processing. Such increases in gamma band activity have been proposed to reflect integrative processing through enhanced coherence of neural activity in multiple interacting regions. In the sequentially coherent conditions, participants have established a context with the first stimulus and can then proceed with further integration and processing with the second stimulus when it is sequentially coherent. The increase in near-gamma that we observe may be related to integration that would involve multiple brain regions in an extended semantic network (Jouen et al 2015).

Cross-Modal Decoding:

The ERP and time frequency analyses revealed late positivities that were common to sentence and image processing, suggesting a common underlying neural mechanism. If the neural dynamics are related for processing the sequentially incoherent stimuli for sentences and images, then a pattern decoder trained on sentences should be able to decode sequential coherence in images, and vice versa. The principal component analysis revealed that the EEG signal reliably encodes information about the modality of the stimuli (sentence vs image), and the status of the second stimulus as being sequentially coherent or incoherent with respect to the first. As a multivariate analysis tool, PCA allows the opportunity of generating principal components from one set of data, and then testing another set of data using those components in order to determine whether the second data set is related to the first (Brouwer & Heeger

2009). Analyses were thus performed to determine whether the principal components that characterize sentence processing would allow decoding for image processing, and vice versa.

The cross-validation results revealed exactly that. When PCA was applied to sentences, the first component reliably distinguished coherent vs. incoherent sentences and images, and only for the crucial second stimulus of the pair. The same positive outcome was observed when applying the complementary analysis, i.e. running PCA on the image data and testing on the sentence data. Thus, this cross-modal generalization argues that the same processes are at work in narrative integration of the successive stimuli for sentences and images.

Behavioral Correlates

This leads to the question of what it is that subjects are actually doing in our task. One suggestion is that without any intention or will to do so, subjects naturally engage with the stimuli and attempt to make sense of them. Data from studies of embodied language indicate that this is the case for language processing (Aziz-Zadeh et al 2006), and we can consider that it is similar for image processing. We have begun to behaviorally investigate this integrative process in the domain of sentence processing. As in the current study, we exposed subjects to successive sentences in pairs that were either sequentially coherent or not. Subjects found the second sentence in the sequentially incoherent pairs less easy to mentally imagine than those in the sequentially coherent condition, and they also required additional time to make the judgement (Madden-Lombardi et al 2015). This suggests that in the sequentially coherent pairs, the representation of the second sentence is already (partially) included in the representation evoked by the first sentences, whereas in the sequentially incoherent pairs, additional processing is required. This additional processing may be reflected in the late positivities we observe, while the integration of the coherent image may be reflected by the increase near gamma response. This processing may be part of attempting to make sense in the context of narrative integration. According to the event indexing model of Zwaan and colleagues (Zwaan 1999, Zwaan et al 1995) coherence can be measured along five dimensions: time, space, causation, motivation and protagonist. In the sequentially incoherent conditions, while relations in the dimensions of time and space may be broken, the protagonist relation remains, so that the sequentially incoherent conditions can be considered as a shift in time or storyline rather than a completely incoherent or unrelated event (Madden-Lombardi et al 2015).

How can such structural processing be mapped onto neural dynamics? In a long-term effort to respond to this question, and based on the property of massive local recurrent connections in primate cortex (Goldman-Rakic 1987) we have developed a class of recurrent network models that are sensitive to sequential structure, from the serial and temporal structure of sensorimotor sequences (Dominey 1998a, Dominey 1998b), to abstract rule structure in language and artificial grammars (Dominey 2005, Dominey et al 2009, Hinaut & Dominey 2011, Hinaut & Dominey 2013). Based on structural regularities in the sentence-meaning data used during training, the recurrent network model can anticipate the next words in an incoming sentence, and when low probability words arrive, the model displays a perturbation of its activity due to adjustment between the anticipatory activity that is inappropriate to what is actually perceived. These perturbations are related to the P600 that is observed in response to sentence structure violations (Hinaut & Dominey 2013).

We have recently extended the model of grammatical construction to that of narrative construction. By analogy, where a grammatical construction defines the relation between words in a sentence and their relation to arguments in a predicate-argument event representation, a narrative construction defines the structural relations between sentences in a narrative and multiple interrelated events and relations between them in a situation model (Mealier et al 2017). Exploiting this analogy, at the level of the narrative construction, we would predict that when incoming sentences that have low probability (with respect to narrative structures learned in the training corpus) arrive, they will likewise result in a perturbation of the model's prediction corresponding to a late positivity.

This suggests the existence of generalized structure processing mechanisms whose response to unpredicted structure will be a late positivity. Interestingly, there is precedence for these predictions. Lelekov et al (2000) presented non-linguistic geometric sequences that could be characterized in terms of their surface structure and abstract structure. Violations of abstract structure produced a reliable late positivity similar to what we observe in the current study. Patel observed similar late positive responses to structural violations in music (Patel et al 1998). His shared syntactic integration resource hypothesis proposes that what is important in these observations across different processing domains is the notion of structural integration, with (Patel 2003). Here we begin to accumulate evidence that such structural processing may apply as well at the level of narrative structure.

Conclusion

From the groundbreaking work of Vandenberghe et al (1996), there is evidence for a common semantic system for words and pictures. Jouen et al (2015) extended this line of research, revealing a more broadly extended semantic system common to the representation of the semantics of human event evoked by sentences and complex images, with extended anatomical and functional connectivity (Jouen et al in revision). The current research reveals evidence for common neural dynamics for semantic processing of short narrative sequences made up of sentences or images. Using EEG we observe common spatiotemporal patterns of brain activity when subjects encounter sequentially incoherent stimuli in the image or sentence modalities. This is revealed in the ERP domain as a set of late central positivities, and in the time-frequency domain as an increase in near-gamma power. This is coherent with related research examining sentence and image processing separately. Importantly, there is nothing inherently wrong with these sequentially incoherent stimuli, it is only their relation with the first stimulus that is manipulated. Further, multivariate analysis discovers this sequential coherence effect in one modality, and can then be used to decode sequential coherence in the other modality, further arguing for a common neurodynamics of meaning integration, independent of the input modality. This demonstrates that the brain tracks coherence not only within individual sentences and images, but across multiple stimuli in a narrative context, and that this mechanism is at least partially independent of the stimulus modality. The question of the nature of the contents of these representations remains a topic of future research. A growing body of research now addresses the representation and decoding of the contents of meaning, e.g. (Wehbe et al 2014). Future research should attempt to apply these techniques across sentence and visual modalities in the further characterization of the common semantic system.

Acknowledgements:

This research was supported by funding from the French ANR Comprendre, and from the European Community through grants FP7-ICT 231267 Project Organic, FP7-ICT-270490 Project EFAA, and FP7-ICT-612139 Project WYSIWYD.

References

- Aziz-Zadeh L, Wilson SM, Rizzolatti G, Iacoboni M. 2006. Congruent Embodied Representations for Visually Presented Actions and Linguistic Phrases Describing Actions. *Current Biology* 16: 1818-23
- Binder JR, Desai RH, Graves WW, Conant LL. 2009. Where is the semantic system? A critical review and meta-analysis of 120 functional neuroimaging studies. *Cereb Cortex* 19: 2767-96
- Bornkessel-Schlesewsky I, Schlewsky M. 2008. An alternative perspective on "semantic P600" effects in language comprehension. *Brain Res Rev* 59: 55-73
- Brouwer GJ, Heeger DJ. 2009. Decoding and reconstructing color from responses in human visual cortex. *The Journal of Neuroscience* 29: 13992-4003
- Brouwer H, Fitz H, Hoeks J. 2012. Getting real about semantic illusions: rethinking the functional role of the P600 in language comprehension. *Brain Research* 1446: 127-43
- Bruner JS. 1990. *Acts of meaning*. Harvard University Press.
- Cohn N, Jackendoff R, Holcomb PJ, Kuperberg GR. 2014. The grammar of visual narrative: Neural evidence for constituent structure in sequential image comprehension. *Neuropsychologia* 64: 63-70
- Cohn N, Maher S. 2015. The notion of the motion: The neurocognition of motion lines in visual narratives. *brain research* 1601: 73-84
- Cohn N, Paczynski M, Jackendoff R, Holcomb PJ, Kuperberg GR. 2012. (Pea) nuts and bolts of visual narrative: Structure and meaning in sequential image comprehension. *Cognitive psychology* 65: 1-38
- 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. 2005. From sensorimotor sequence to grammatical construction: Evidence from simulation and neurophysiology. *Adaptive Behavior* 13: 347-61
- 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
- Goldman-Rakic PS. 1987. Circuitry of Primate Prefrontal Cortex and Regulation of Behavior by Representational Memory. *Handbook of Neurophysiology* 5: 40
- Grandchamp R, Delorme A. 2011. Single-trial normalization for event-related spectral decomposition reduces sensitivity to noisy trials. *Frontiers in Psychology* 2
- Hagoort P, Hald L, Bastiaansen M, Petersson KM. 2004. Integration of word meaning and world knowledge in language comprehension. *Science* 304: 438-41
- Hagoort P, van Berkum J. 2007. Beyond the sentence given. *Philos Trans R Soc Lond B Biol Sci* 362: 801-11
- Hald LA, Bastiaansen MC, Hagoort P. 2006. EEG theta and gamma responses to semantic violations in online sentence processing. *Brain Lang* 96: 90-105
- Hinault X, Dominey PF. 2011. A three-layered model of primate prefrontal cortex encodes identity and abstract categorical structure of behavioral sequences. *J Physiol Paris* 105: 16-24
- Hinault X, Dominey PF. 2013. Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. *PLoS one* 8: 1-18
- Holm S. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*: 65-70

- Jouen A-L, Ellmore TM, Madden-Lombardi CJ, Pallier C, Dominey PF, Ventre-Dominey J. in revision. Beyond the Word and Image: II Structural and Functional Connectivity of a Common Semantic System.
- Jouen A-L, Ellmore TM, Madden CJ, Hidot S, Dominey PF, Ventre-Dominey J. *Neurobiology of Language, San Sebastian, Spain, 2012.*
- Jouen A, Ellmore T, Madden C, Pallier C, Dominey P, Ventre-Dominey J. 2015. Beyond the word and image: characteristics of a common meaning system for language and vision revealed by functional and structural imaging. *NeuroImage* 106: 72-85
- Kaan E, Swaab T. 2003. Repair, revision, and complexity in syntactic analysis: An electrophysiological differentiation. *Cognitive Neuroscience, Journal of* 15: 98-110
- Keizer AW, Verment RS, Hommel B. 2010. Enhancing cognitive control through neurofeedback: A role of gamma-band activity in managing episodic retrieval. *NeuroImage* 49: 3404-13
- Kuperberg GR. 2007. Neural mechanisms of language comprehension: challenges to syntax. *Brain Res* 1146: 23-49
- Kuperberg GR, Sitnikova T, Caplan D, Holcomb PJ. 2003. Electrophysiological distinctions in processing conceptual relationships within simple sentences. *Cognitive Brain Research* 17: 117-29
- Laljee S, Dominey PF. 2013. Multi-modal convergence maps: from body schema and self-representation to mental imagery. *Adaptive Behavior* 21: 274-85
- Lelekov T, Dominey PF, Garcia-Larrea L. 2000. Dissociable ERP profiles for processing rules vs instances in a cognitive sequencing task. *Neuroreport* 11: 1129-32
- Madden-Lombardi C, Jouen A-L, Dominey PF, Ventre-Dominey J. 2015. Sequential Coherence in Sentence Pairs Enhances Imagery during Comprehension: An Individual Differences Study. *PLoS one* 10: e0138269
- Mealier A-L, Poiteau G, Mirliaz S, Ogawa K, Finlayson M, Dominey PF. 2017. Narrative Constructions for the Organization of Self Experience: Proof of Concept via Embodied Robotics *Frontiers in Psychology: Language*
- Meyer K, Damasio A. 2009. Convergence and divergence in a neural architecture for recognition and memory. *Trends in neurosciences* 32: 376-82
- Neville H, Nicol JL, Barss A, Forster KI, Garrett MF. 1991. Syntactically based sentence processing classes: Evidence from event-related brain potentials. *Journal of cognitive Neuroscience* 3: 151-65
- O'sullivan JA, Power AJ, Mesgarani N, Rajaram S, Foxe JJ, et al. 2014. Attentional selection in a cocktail party environment can be decoded from single-trial EEG. *Cerebral cortex* 25: 1697-706
- Osterhout L, Holcomb PJ. 1992. Event-related brain potentials elicited by syntactic anomaly. *Journal of memory and language* 31: 785-806
- Paczynski M, Kuperberg GR. 2012. Multiple influences of semantic memory on sentence processing: Distinct effects of semantic relatedness on violations of real-world event/state knowledge and animacy selection restrictions. *Journal of memory and language* 67: 426-48
- Patel AD. 2003. Language, music, syntax and the brain. *Nature neuroscience* 6: 674
- Patel AD, Gibson E, Ratner J, Besson M, Holcomb PJ. 1998. Processing syntactic relations in language and music: An event-related potential study. *Journal of cognitive neuroscience* 10: 717-33
- Schaefer RS, Farquhar J, Blokland Y, Sadakata M, Desain P. 2011. Name that tune: decoding music from the listening brain. *NeuroImage* 56: 843-49
- Schneider TR, Debener S, Oostenveld R, Engel AK. 2008. Enhanced EEG gamma-band activity reflects multisensory semantic matching in visual-to-auditory object priming. *Neuroimage* 42: 1244-54
- Sitnikova T, Holcomb PJ, Kiyonaga KA, Kuperberg GR. 2008. Two neurocognitive mechanisms of semantic integration during the comprehension of visual real-world events. *Journal of cognitive neuroscience* 20: 2037-57

- van Berkum JJ, Brown CM, Hagoort P, Zwitterlood P. 2003. Event-related brain potentials reflect discourse-referential ambiguity in spoken language comprehension. *Psychophysiology* 40: 235-48
- Van Berkum JJ, Brown CM, Zwitterlood P, Kooijman V, Hagoort P. 2005. Anticipating upcoming words in discourse: evidence from ERPs and reading times. *J Exp Psychol Learn Mem Cogn* 31: 443-67
- Van Berkum JJ, Hagoort P, Brown C. 1999. Semantic integration in sentences and discourse: Evidence from the N400. *Cognitive Neuroscience, Journal of* 11: 657-71
- Vandenberghe R, Price C, Wise R, Josephs O, Frackowiak RS. 1996. Functional anatomy of a common semantic system for words and pictures. *Nature* 383: 254-6
- Wehbe L, Murphy B, Talukdar P, Fyshe A, Ramdas A, Mitchell T. 2014. Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses. *PloS one* 9: e112575
- Xiang M, Kuperberg G. 2015. Reversing expectations during discourse comprehension. *Language, cognition and neuroscience* 30: 648-72
- Zwaan RA. 1999. Five dimensions of narrative comprehension: The event-indexing model. *Narrative comprehension, causality, and coherence: Essays in honor of Tom Trabasso*: 93-110
- Zwaan RA, Langston MC, Graesser AC. 1995. The construction of situation models in narrative comprehension: An event-indexing model. *Psychological science*: 292-97

Table 1. Discrimination of sequential coherence by an Early ERP positivity. Statistical p-values for Wilcoxon signed rank comparisons. Right two columns (Coherence Discrimination) illustrate electrodes with significant coherence differences for sentences and images in the Early Positivity (250 – 350 ms) window. We also illustrate (Modality Discrimination) how in the same window several of these electrodes distinguish sentence and image. This demonstrates a form of mixed selectivity. All of these electrodes display a significant sequential coherence x order interaction in the ANOVA.

Electrode	Modality Discrimination		Coherence Discrimination	
	Coherent	Incoherent	Image	Sentence
Central (C1-12)	0,005	0,025	0,0084	0,0074
Centro-Frontal (FCz-47)	0,006	0,008	0,0074	0,0096
Centro-Parietal (CPz-32)	-	-	0,0084	0,0057
(CP1-19)	-	-	0,0096	0,0043
(CP2-56)	0,01	0,043	0,005	0,0038
Parietal (Pz-31)	-	-	0,0074	0,0043
Temporal (T8-52)	-	-	0,0074	0,0096

Table 2. Discrimination of sequential coherence by Late ERP Positivity. Right two columns (Coherence Discrimination) illustrate electrodes displaying significant sequential coherence differences for sentences and images in the 750-1000 ms window, and the corresponding statistical p-values for Wilcoxon signed rank comparisons. All electrodes except FT8 displayed a significant sequential coherence interaction. A significant effect in the late positivity 1600-1800ms time window was observed in all of these electrodes with the exception of FT8-43, and CP2-56. We also illustrate (Modality Discrimination) how in the same window these electrodes distinguish sentence and image. This demonstrates a form of mixed selectivity.

Electrode	Modality Discrimination		Coherence Discrimination	
	Coherent	Incoherent	Image	Sentence
Central (C4-50)	0.0018	0.0012	0.0084	0.0475
Fronto-Central				
(FC1-11)	0.0018	0.0065	0.0014	0.0475
(FC2-46)	0.0018	0.0025	0.0009	0.0386
(FC3-10)	0.0033	0.0084	0.0043	0.0311
(FC4-45)	0.0029	0.0021	0.0016	0.0475
(FC6-44)	0.0043	0.0007	0.0025	0.0429
Centro-Parietal				
(CP2-56)	0.0002	0.0009	0.0156	0.0386
Fronto-Temporal				
(FT8-43)	0.0279	0.005	0.0347	0.0429

Table 3. Discrimination of sequential coherence by Near-Gamma oscillatory power. Right two columns (Coherence Discrimination) illustrate electrodes displaying significant sequential coherence differences for sentences and images in the near-gamma (34-45 Hz) range, and the corresponding statistical p-values for Greenhouse-Geisser corrected post-hoc comparisons.

Electrode	Time window	Coherence Discrimination	
		Image	Sentence
Central (C1-12)	2000-2200	0.030	0.0241
Fronto-Central			
(FC5-9)	1200-1400	0.0449	0.0064
(FC3-10)	2200-2400	0.0315	0.0214
(FC1-11)	2000-2200	0.0374	0.0365

	Late 1 (750-1000)		Late 2 (1600 – 1800)	
	SENT	IMA	SENT	IMA
PC1	0.159	0.200	0.005*	0.159
PC2	0.132	0.015*	0.141	0.017*

Table 4. Sequential coherence discrimination for Stimulus 2. Wilcoxon p values for NonSEQ – SEQ differences for the second stimulus, based on projection of SENT and IMA data for SEQ and NonSEQ trials onto the first two PCs generated by applying PCA to the entire data set. We can observe that between PC1 and PC2 the NonSEQ – SEQ distinction reliably be made.

	Late 1 (750-1000)		Late 2 (1600 – 1800)	
	Test on Image1	Test on Sentence1	Test on Image1	Test on Sentence1
Train on Images	0.616	(0.845)	0.647	(0.145)
Train on Sentences	(0.586)	0.777	(0.616)	0.144

Table 5. Sequential coherence cannot be discriminated for Stimulus 1. Control test of PCA decoding of coherence of the first stimulus, depending on training and testing conditions. Wilcoxon p values for Sequentially Coherent - Incoherent differences for stimulus 1. Significant values ($p < 0.05$) indicate that the coherence difference is reliable. Upper row provides results from training on Image data, lower row from training on Sentence data. Parentheses () indicate the crucial transfer conditions where training is in one modality, and testing in the other. Here we expect no differences to be significant, since the coherence distinction cannot be made at the first stimulus.

	Early (750-1000)		Late (1600-1800)	
	Test on Image2	Test on Sentence2	Test on Image2	Test on Sentence2
Train on Images	0.016	(0.048)	0.031	(0.009)
Train on Sentences	(0.020)	0.043	(0.039)	0.009

Table 6. Sequential coherence is cross-modally decoded for Stimulus 2. Transfer test of PCA decoding of sequential coherence of the second stimulus, depending on training and testing conditions. Wilcoxon p values for NonSEQ – SEQ differences for stimulus 2. Significant values ($p < 0.05$) indicate that the Sequential Coherence difference is reliable. Upper row provides results from training on Image data, lower row from training on Sentence data. Parentheses () indicate the crucial transfer conditions where training is in one modality, and testing in the other. Here we expect all differences to be significant.

Figure captions:

Figure 1. Temporal unfolding of the experimental protocol for the image (A) and sentence (B) conditions.

Figure 2. Comparison of ERP responses to first and second stimuli. A. Stimulus 1 – small effects of modality are seen, but no effects of sequential coherence are visible (as expected). B. Sequential coherence effects become visible.

Figure 3. P300 (250-350ms) effect. This positivity makes a modality distinction, with a greater positive effect for sentences. It also makes the NonSEQ (sequentially incoherent) vs. SEQ (sequentially coherent) distinction, with an increased positivity for the sequentially incoherent stimuli. Illustrated with Electrode FC1 (11).

Figure 4. In a late N400 timeframe (525-600ms) there is a significant negativity for NonSEQ (sequentially incoherent) vs. SEQ (sequentially coherent) for images but not sentences. Electrodes 2[AF7], 3[AF3], 6[F5], 36[AF4], 41[F6] display a significant effect, only for the second stimulus.

Figure 5. Late positivities (750-1000, and 1600-1800 ms) are sensitive to the NonSEQ (sequentially incoherent) vs. SEQ (sequentially coherent) distinction for sentences and images. Scalp maps for sentences and images INCO-sequential coherence contrast for 750-1000ms displayed on left, and for 1600-1800 displayed on right. Illustrated with Electrode FC1 (11).

Figure 6. Near Gamma (Hz) oscillations are increased for Stim 2 in the sequentially coherent condition. Panels a-d illustrate TF data for electrode FC5-9 in response to Stim 1 for sentences and images in the non-sequential and sequential conditions. No difference between Non-SEQ and SEQ for sentences, nor for images. Panels a*-d* zoom in on the 1200-1400ms 35-45 time-frequency window with increased resolution. Again no Non-SEQ vs. SEQ differences. Same spatial organization for panels a2-d2*, but for Stim 2, where Non-SEQ vs. SEQ differences are predicted, and observed. For sentences and images there is increased near gamma activity in the 1200-1400ms time-frequency window.

Figure 7. A. Projection of IMA and SENT, SEQ and Non-SEQ data onto PC1. Clear separation of IMA vs SENT trials in multiple time windows. B. Spatial distributions of contributions of electrodes to PC1. C. Projection of IMA and SENT, SEQ and Non-SEQ data onto PC2. Clear separation of SEQ and Non-SEQ trials in the 750-1000 and 1600-1800 timeframes. D. Spatial distributions of contributions of electrodes to PC2.

Figure 1

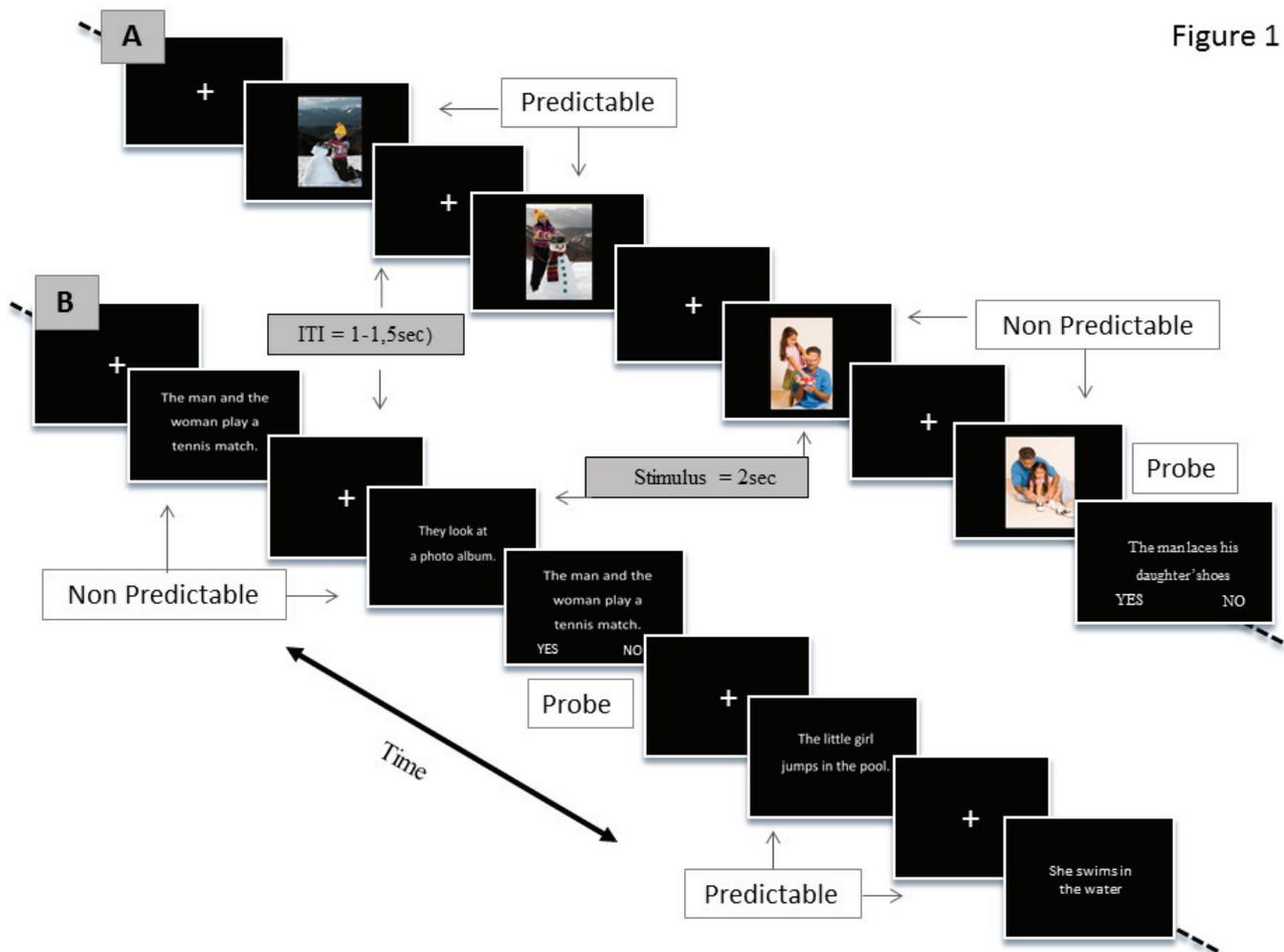


Figure 2

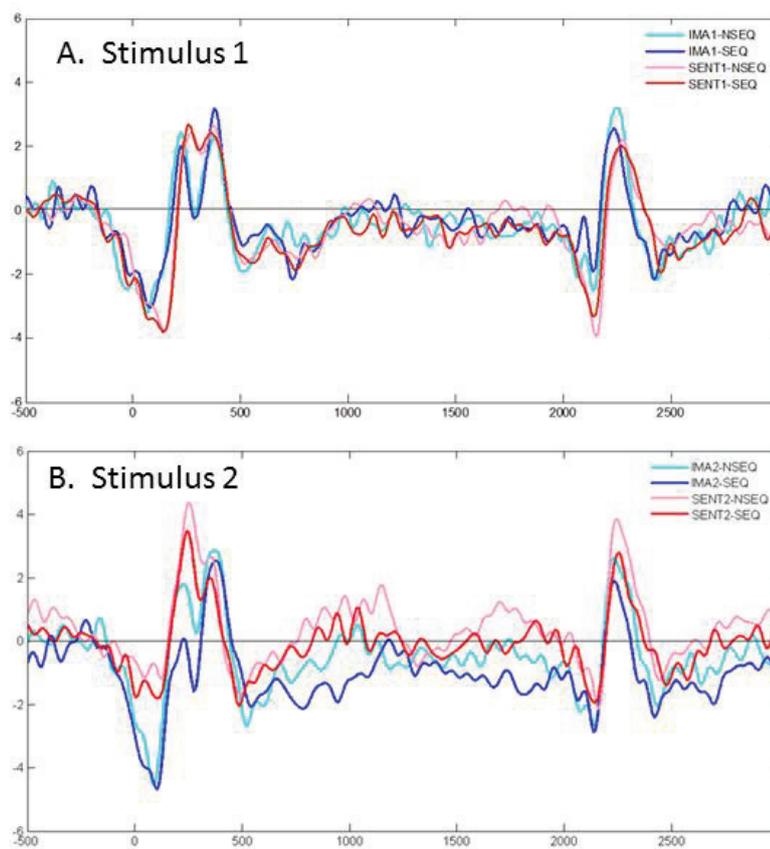


Figure 3

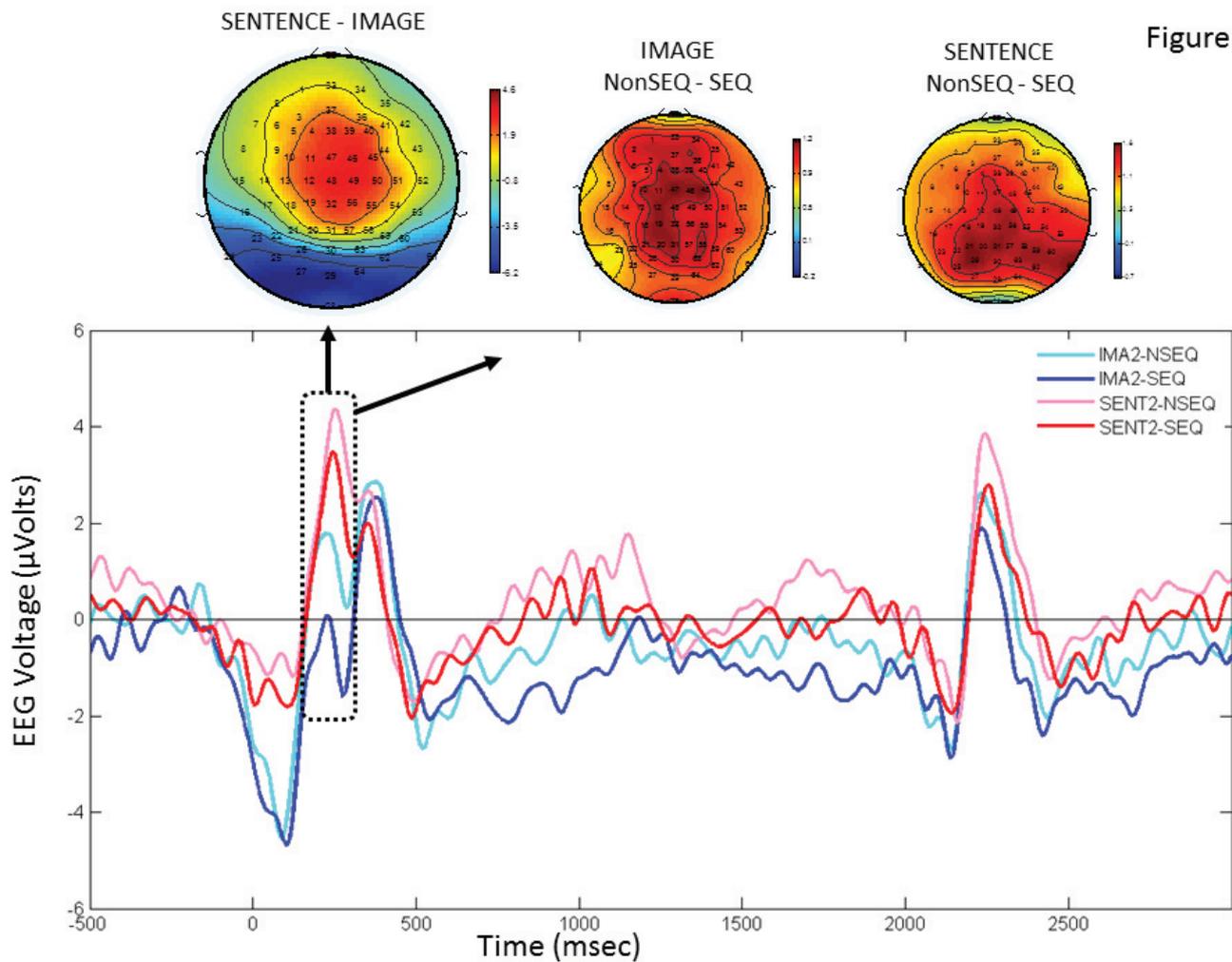


Figure 4

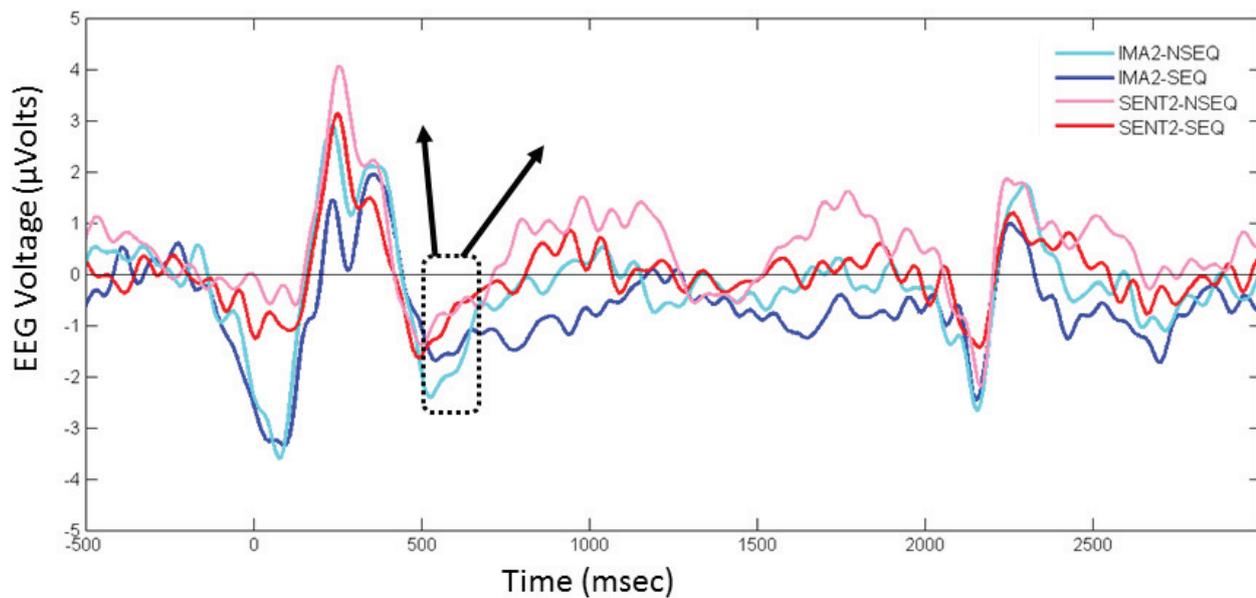
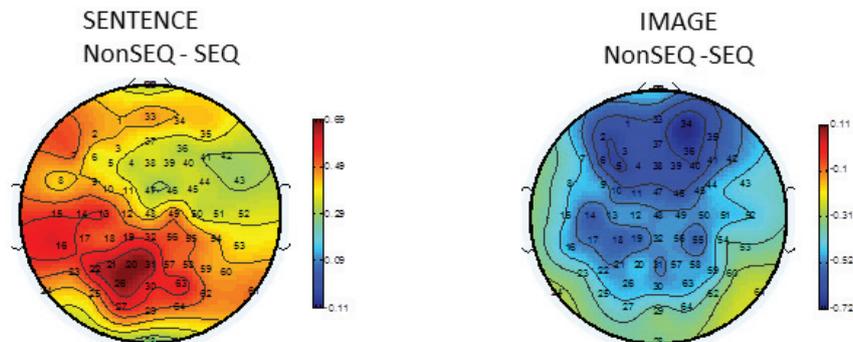


Figure 5

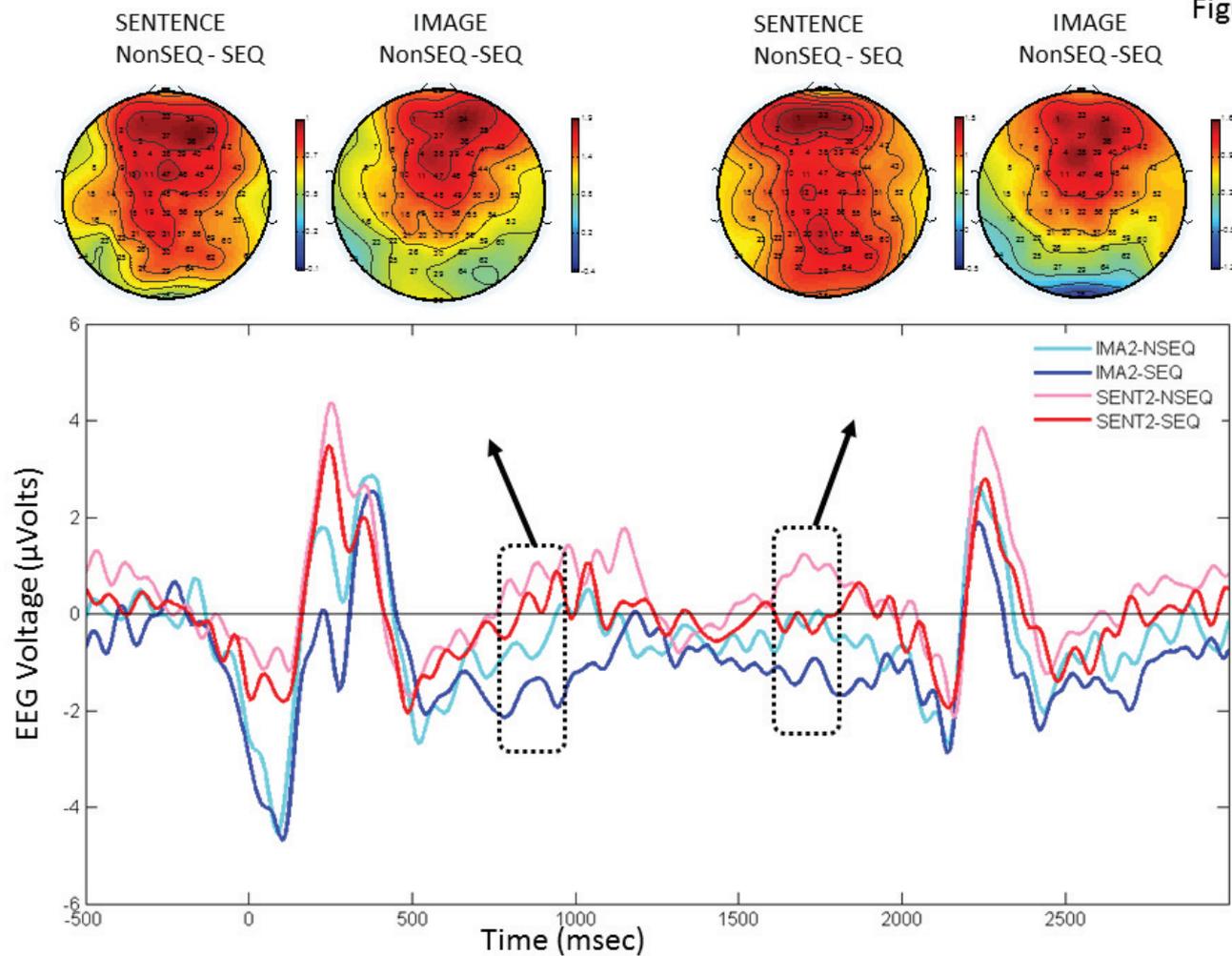


Figure 6

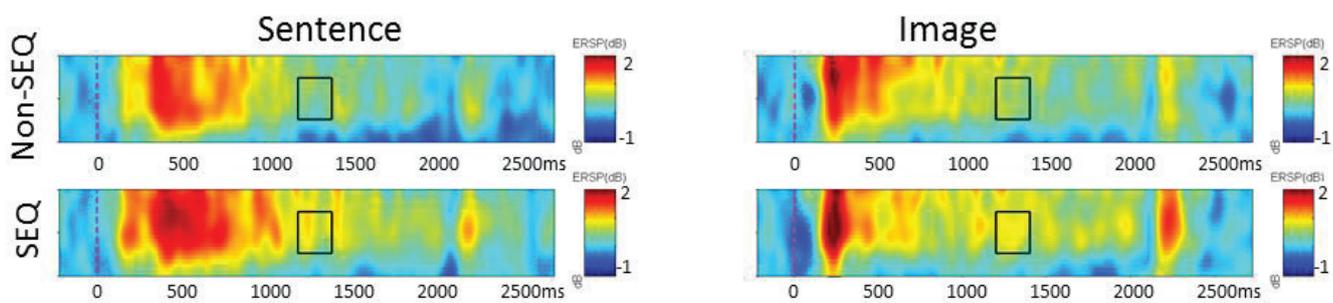
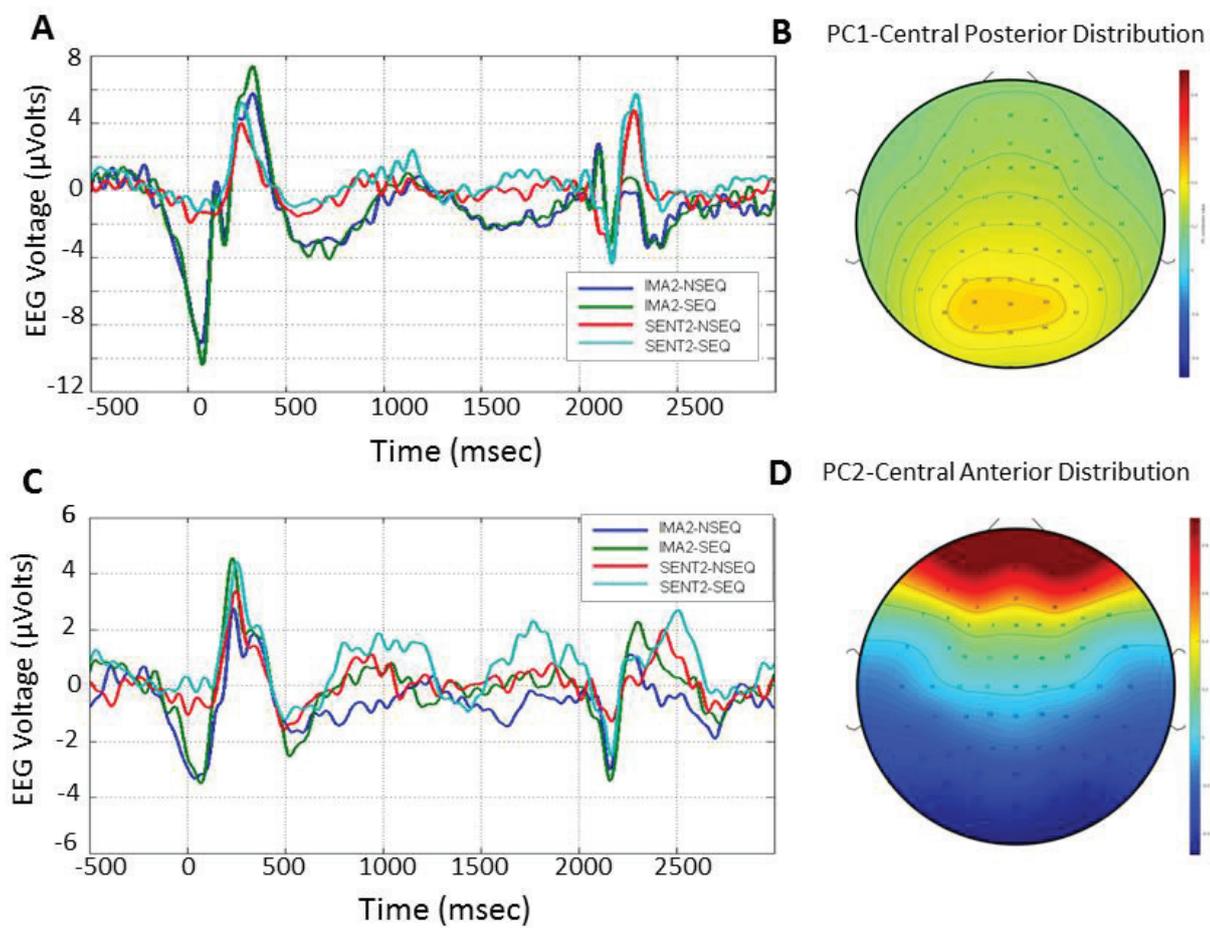


Figure 7



3. Discussion

We set out to determine how the replay of experienced behavior could allow for a form of optimization of learning, within the behavioral and neurophysiological context of the “traveling salesrat” paradigm. Here, rats are observed to find the most efficient path linking a set of baited food wells, in a surprisingly short number of trials. We hypothesized that the organized replay of recently experienced behavior could allow the rat to amplify its experience, and contribute to the fast resolution of the problem. A major question then concerns how the replay can bias learning towards the optimal solution. During the thesis, we developed a replay algorithm that propagates reward value along the replayed trajectory, thus implementing a form of spatio-temporal credit assignment in a reinforcement-learning context, by biasing replay probability to favor snippets that are on an efficient path to a rewarded location. We then demonstrated how the PFC-like reservoir could learn from this biased distribution of replay in order to consolidate the most efficient path. In the following sections, we situate this work in a broader context and discuss its pertinence within these domains.

The TSP problem is a global optimization problem and the behavior of an agent that is able to solve this problem could be demonstrated with a reinforcement learning algorithm like Q-Learning (Sutton and Barto 1998; Barrera et al. 2015) by learning the successful action sequence maximizing the expected reward. It requires an explicit discretization of perceptual observable states and an explicit reward but does not require a model of the environment.

Experiments with an animat require defining an embodied and situated agent equipped with sensors and actuators and a behavioral control architecture that relates its perceptions to its actions and allows it to survive in its environment. We defined a sensor model by considering the hippocampus place-field response to a particular location and a transition model by restricting the locations the agent can reach at each time step to a circle around the center of its body. Thus, we defined a simple navigation policy by selecting at each time step the most probable location encoded in the prediction of the next place-cell activation pattern, based on forward and backward random replay of snippets taking into account the notion of reward. By doing this, we defined two generative models. A generative model is “an internal model that encodes the probabilistic relations between states, actions, and rewards. Such a model permits to generate observable data given some other hidden (non-directly observable) parameters, and ultimately permits to estimate the value of a plan“ (Pezzulo et al. 2014). The first generative model provides short place-cells activation sequences given a stream of sensory input and the associated reward information. It emulates the replay phenomenon observed during awake SPW-R. Snippets are randomly drawn from a replay likelihood estimated with a recursive algorithm based on $TD(\lambda)$, described in (Sutton and Barto 1998) and implementing a form of reinforcement learning. The snippet replay direction has the effect of propagating the reward forward and backward in time, allowing one to have an estimate of the future reward. The recursive definition allows the generative model to be compatible with a recurrent neural network structure, also found in area CA3 of the hippocampus. Once learnt, the replay generative model is able to produce snippets that reflect the accessible rewards information. When coupled with a stimulus coding a space information, it is thus possible to generate a dynamical training set that represent sub-trajectories associated to a reward. This dynamical training set is used for training the second generative model which is the consolidation model (see Figure 6) implemented by a PFC and ST model (see Figure 7). This model exploits the reservoir’s ability to align in neurons state space the activation sequences sharing a common subsequence. This result in the concatenation of overlapping subsequences of place cell activation and when overlapping subsequences reflect the overlapping subsequences related to a reward, it is then possible to concatenate efficient parts of trajectories experienced before. The learning rule is very simple and as opposed to other recurrent neural network model using Backpropagation through time

or other complex algorithms, we solved the credit assignment problem by pairing two models, implementable by random recurrent networks.

At this point, multiple efficient trajectories might exist and it is possible to give an account of all the trajectories the agent can generate based on the consolidated knowledge of the trajectories previously encountered and the reward information associated with salient points of these trajectories by simulating several random walks of an agent implementing this generative model. Findings described in section 2.4 provide a link between awake SPW-R and hippocampal memory processes involved in a goal directed behavior. Simulated agents exhibited a behavior suggesting an incremental update of their internal representation of the world, based only on the cumulated and rewarded experiences acquired during the previous trials. Snippet replay allowed the agents to recombine rewarded parts of the previous trajectory and to consolidate a trajectory between feeders that privileges paths between baited feeders. It is possible to consolidate the multiple trajectories that exist between baited feeders by exposing a sequence learning model more often to place-cell subsequences related to a reward. Finally, we studied the TSP, which is a particular optimization problem by implementing a generative model of the optimal sequence with a HC-PFC-ST joint model featuring a particular form of replay during awake SPW-R that emphasizes short rewarding sub-paths. An interesting perspective could be to extend this study by implementing a more comprehensive model of hippocampal replay as characterized in (Gupta et al. 2010).

Hierarchical hidden Markov model

Reservoir states form a lattice of observable states whose transition probabilities result from the likelihood of a snippet to be replayed. In fact, the PFC model based on reservoir computing allows the model to respect the Markov assumption by maintaining and providing at each time step an account of the recent observable (place-cell activation) states. It associates to any stimuli sequence a state that is characteristic of the recent history of stimuli sequence and the mnemonic abilities of the PFC model are mainly determined by the number of neurons, the spectral radius (maximum absolute value of the eigenvalues) of the recurrent connectivity matrix and the leak rate that reflects the time constant of the neurons by emulating the resistive and capacitive properties of neurons membrane. Reverse replay (Foster and Wilson 2006b; Ambrose, Pfeiffer, and Foster 2016) is a potential mechanism to the credit assignment problem encountered in reinforcement learning : In addition of virtually exposing the agent to transitions between locations in both directions as demonstrated in (Gupta et al. 2010) and in section 2.3, we demonstrated that reverse replay of snippets allows the backward propagation of the reward through the reverse consolidation of place-cell transitions related to trajectory parts emphasized by reward and forward replay emphasizes forward transitions between place-cell patterns (Wikenheiser and Redish 2014).

We proposed a hierarchy of discrete time models which could be viewed individually as an autonomous hidden Markov model (Markov 1913; Dugad and Desai 1996; Jurafsky and Martin 2017). It is then possible to view the animat model as a hierarchical hidden Markov model (Fine, Singer, and Tishby 1998)

With a fully specified hierarchical hidden Markov model implemented by our joint HC-PFC-ST model, it is then possible to benefit from several algorithms:

➔ Forward algorithm: It is possible to evaluate a belief state of the agent at a given time which is the probability of a position at a particular time, given the history of place-cells activation values by applying the forward algorithm on the HHMM. It consists in recursively evaluating the probability of the agent's position given successive place-cell activation patterns. With the help of the forward algorithm on a HHMM, is then possible to:

1. Filter a position by estimating the posterior distribution of current position given all available place-cell activations

2. Predict a position by estimating the posterior distribution of a future position given all available place-cell activations

The recursive definition of the forward algorithm allows one to evaluate the belief state of the agent's position given all available place-cell activations (the likelihood of the trajectory) online during navigation and opens the door to online planning.

→ Viterbi algorithm: It is possible to find a sequence of observable states (reservoir states) that reflects a sequence of hidden states (position of the agent) that maximizes the reward and minimizes the walking distance of the agent during a trial by using a Viterbi algorithm (Viterbi 1967). This is a solution to the TSP problem and we demonstrated that our joint HC-PFC-ST model is sufficient for learning a state transition model that does not require an explicit discretization of perceptual (observable) states. Viterbi algorithm rely on the forward algorithm and instead of estimating recursively the posterior distribution of a position at a particular time given all the place-cells activations observed so far, one can estimate recursively the probability of the most probable path to each position given all available observations. Practically speaking a Viterbi algorithm can be implemented by selecting at each time step of a trajectory the location that maximizes the current forward path probability, which is given by the product of the previous forward path probability, the transition probability given by the transition model and the state observation likelihood given by the sensor model. The preplay of place-cell subsequences related to the current location of the agent observed in (Diba and Buzsáki 2007b) could be interpreted as a prospective mechanism playing a role in the evaluation of possible solutions related to the current agent's situation (Pezzulo et al. 2014). Since the forward algorithm is recursive, it is possible to maintain an estimate of the current position with the place-cell activations encountered gradually and to evaluate the outcomes of the possible trajectories derived from the current trajectory and the remaining suffixes suggested by the snippets being replayed. Implementing this algorithm in our model would imply to implement the online prospecting mode described in Figure 7 panel C.

→ Forward-backward algorithm: A hidden Markov model relies on parameters transition probabilities and emission probabilities. The standard algorithm for training a HMM and to find the model that explains a given sequence is the forward backward algorithm or the Baum-Welch algorithm (Baum and Petrie 1966; Welch 2003). It is a special case of the Expectation-Maximization algorithm. The algorithm is iterative and consists in 2 successive procedures repeated for each iteration until convergence that can be summarized as follows:

1. Expectation: For $1 < t < T$ apply the following algorithms:
 - Forward: The forward algorithm is used in order to estimate the probability of observing t available place-cell activations and being in the current position on time t .
 - Backward: The backward algorithm is also a recursive algorithm on a HMM that is used for smoothing a position by estimating the posterior distribution of a past position given all available place-cell activations. It computes the probability of ending with $T-t$ place-cell activation given the current position at time t
2. Maximization: Update the actual emission and transition probabilities and the initial position distribution estimates by using the results computed in the expectation step

There is no trivial link between the forward backward algorithm and the snippet driven learning paradigm we are using for training a random recurrent neural network but in (Unkelbach, Yi, and Schmidhuber 2009), the authors propose an expectation maximization algorithm training algorithm for recurrent neural network (Greff, van Steenkiste, and Schmidhuber 2017) performing a time series prediction task. In (Ma and Ji 1998), the authors demonstrate a learning algorithm for echo states networks (Jaeger 2001) whose formulation is very close to temporal neural network (Dominey 1995). There exists also a training method based on Kalman filter for training a recurrent neural network (Puskorius and Feldkamp 1994).

We hypothesize that snippet driven learning of a reservoir model fits in this framework. The expectation step, which consists in estimating the joint probability distribution over positions, conditioned on the observed place-cell activation sequences given the current readout synaptic weights, could be implemented by using random replay of snippets in forward and backward direction. The maximization step could be assimilated to the gradient descent procedure used for training readout weights. Finally, the iteration of the expectation maximization algorithm until convergence could be implemented by allowing the snippet driven learning to occur between each trial during several trials, until the generated trajectories converge to an efficient trajectory. This corresponds to the awake snippet replay we modelled and studied.

We then propose to extend the snippet driven learning paradigm by integrating the trajectory generated during the previous trial as a part of the awake SPW-R content. An additional exploratory navigation policy might be required and this allows the animat to mimic the rat behavior. This is the subject of a collaborative work with USF Tampa described in section 2.5

Free energy and active inference

Hidden Markov model is a particular case of dynamic Bayesian network. It is possible to describe the animat implementation of a joint HC-PFC-ST model under the light of the free energy framework as a hierarchical Bayesian model (K. Friston 2008). By describing our model within a unified framework, it allows one to view the model with a high level of abstraction while relying on correct neurophysiological basis and to extend the model with ad-hoc or neuro-mimetic components for studying other problems that could be explained with an active inference model. An example could be the work accomplished in section 2.6 which relies on the same mixed selectivity property observed in reservoir computing (Rigotti et al. 2013; Enel et al., n.d.) or the hierarchical model used in this thesis and illustrated in Figure 6. A collaboration with ISTC CNR had been initiated during a visit (see Rome-ITALY 2017 Presentation) and aims at proving more rigorously that the animat as we modelled it in this thesis learns a generative model minimizing free energy, based on the existing work about free energy based models (Klaas Enno Stephan et al. 2010; Chumbley, Dolan, and Friston 2008; Kiebel et al. 2009; K. Friston 2003; K. J. Friston, Daunizeau, and Kiebel 2009; K. Friston et al. 2016; K. J. Friston et al. 2017).

Global optimization

The TSP is a global optimization problem and from this point of view, it is interesting to compare the heuristic implemented in this thesis with existing algorithms able to solve the TSP.

The number of possible sub trajectories combinations is dramatically reduced when taking into account the reward and using reverse replay when training the consolidation model allows one to consider only a non-directed graph problem by virtually exposing the consolidation model to trajectory subsequences in forward and reverse direction. It is necessary but not sufficient for allowing an agent to navigate. We had to evaluate the animat's trajectories in non-autonomous mode when reverse replay was required for consolidating solutions derived from sub trajectories in reverse direction. It caused a tendency to revisit feeders already visited and no longer containing food. One needs to implement a preplay mechanism (Diba and Buzsáki 2007b) that will replay snippets related only to future rewards, given the current position of the animat and excluding the snippets associated to a reward perception that minimizes the error with the reward prediction by updating the learning rule of the replay model. It is then possible to realize a Viterbi algorithm that will allow the animat to generate the optimal sequence.

The closest heuristic that solves the TSP is the ant colony system (ACS) (Dorigo and Gambardella 1997; Yang et al. 2008), where multiple agents (ants) deposit an amount of 'pheromones' that is proportional to the length of their tour in the solution space. Each agents moves stochastically according to the concentration level of

pheromones in the solution space and increases this concentration level each by releasing more pheromones along the trajectory they just described. The possible shortest routes emerge as the paths following the most concentrated pheromone trails. This is very analogue to the model we developed in this thesis if one considers that an ‘ant’ corresponds to the simulated agents when establishing a random walk map. The pheromone trail that is shared and built collectively by the ant colony is emulated by the incremental learning of the snippet replay likelihood through multiple replay episodes. The consolidation model allows one agent to generate trajectories related to rewarded paths (analogue to a path along the pheromone trail), either in an offline simulation during the consolidation process, or during an online generation/exploration process that will allow new paths to be discovered/associated and consolidated after the end of the current trial during SPW-R.

Multiple timescale

The recurrent networks we used in our implementation were relative to the same time constant, allowing them to function only in a restricted timescale that matches the intrinsic timescale of the stimuli sequence. We demonstrated multiple rewarded sequences consolidation properties with simplified trajectories displaying a constant speed and thus featuring a single characteristic timescale. Several authors demonstrated the functioning principle and some properties of recursive recurrent neural network (Tani and Nolfi 1999; Jaeger 2007; Yamashita and Tani 2008). The main idea is to consider two or three interacting recurrent neural networks having a characteristic timescale (leak rate). It is thus possible to view each network as a time basis: Any stimuli sequence could be decomposed in several spatio-temporal features resulting from one or more timescales. The readout layer can then select the spatio-temporal features correlated to the expected signal with an online and supervised learning rule. Intuitively, it is similar to a wavelet analysis/synthesis process. Learning selects the best time/space basis for representing a high-dimensional stimuli sequence featuring fast and slow variations.

Deep learning

Recent advances in machine learning and the raising computing power of modern computers allowed the emergence of the deep learning as a new applied research field. Briefly, it consists in processing a massive data amount with several hierarchical layers of neural networks, each layer learning more and more abstract representations of the dataset. A particular type of deep neural network is the recurrent neural network (RNN) whose static architecture is similar to the model used in this thesis. RNN are typically trained with a backpropagation through time (Werbos 1990) or backpropagation through structure (Goller and Kuchler 1996). The subject is complex and an entire PhD thesis had been dedicated to the training of recurrent neural network (Sutskever 2013). The training algorithm we used for the consolidation model as described in 2.4 can be assimilated to a truncated back propagation through time (TBPTT) algorithm, learning only synaptic weights between the output layer and the unique hidden layer. Recurrent weights are never modified. The credit assignment problem inherent to sequence learning is solved very simply by training a replay generative model beforehand. Instead of modifying the synaptic weights of the recurrent connections according to a delayed form of reward, we used a form of $TD(\lambda)$ algorithm where the time delayed reward information is propagated through the place cell subsequences replay phenomenon observed during SPW-R. The resulting replayed place cell subsequences constitutes a dynamic training set for the consolidation model where only the place cell subsequences related to a reward are represented. This replay model emulates the activations of the hippocampus. The training process of the replay model might involve other areas of the brain, in particular for justifying at least the reward circuit, and the notion of novelty associated to place-cells transitions. At this point, we can state that our model is biologically inspired but is not necessarily biologically plausible. The replay and consolidation model are learnt only during SPW-R in order to elaborate a navigation policy based on a reward estimate. It is a form of reinforcement learning and the use of continuous recurrent neural network

allows one to consider continuous sensory and action states and a continuous time (a series of discrete time events separated by a delay that could be arbitrarily small). This model should belong to the deep reinforcement learning (Arulkumaran et al. 2017) family of algorithms because it “scales to decision-making problems that were previously intractable, i.e., settings with high-dimensional state and action spaces”. The overall model developed in this thesis might be considered as a deep temporal model (K. J. Friston et al. 2017)

Cognitive maps

We demonstrated that hippocampal replay is a suitable mechanism playing a role in offline consolidation and suggested that online planning could use this mechanism as well.

In (Pezzulo et al. 2014), a generative model of plan values supporting vicarious trial-and-error and goal-directed behavior is described. We implemented a generative model, implemented by a random recurrent neural network and described as a dynamic Bayesian network. We demonstrated the ability of our joint HC-PFC-ST model to learn a generative model in a compact manner, robust to noise and benefiting from the neural network generalization ability. This work is an empirical attempt to demonstrate the argument developed in (Wikenheiser and Redish 2014) where the authors conclude that “sequences play a more active and complex role in information processing than encoding veridical experience. Their role in flexibly manipulating and permuting representations of space to generate novel paths that might aid action selection meshes well with the cognitive map envisioned by Tolman (Tolman 1948; Johnson and Crowe 2008)”

Piaget schema

The notion of schema had been introduced to cognitive psychology by (Piaget 1967) and (Bartlett 1932) in their efforts to understand how new information is integrated with pre-existing knowledge.’ A schema can be viewed as ‘any organized network of overlapping representations that has the following properties:

1. New information is better remembered when it fits within a pre-existing schema
2. New information that challenges schema organization may cause modification of the existing schema or development of a new schema
3. Schemas support novel inferences between indirectly related events and their generalization to new situations.’

We demonstrated in 2.4 that it is possible to establish a transitive relationship between parts of trajectories associated to a reward, based uniquely on random replay. The replay model implemented partially by the hippocampus model, proposes associations of rewarded trajectory parts and the consolidation model implemented by a model of prefrontal cortex and striatum, associates overlapping rewarded trajectories by aligning their spatiotemporal representations through the online association of the states of the reservoir’s neurons to the prediction of the next input of the consolidation model. Thus, it is possible to replay and consolidation models as the two parts of a schema generative model: Property 1 is ensured by the use of an online and supervised learning rule, which will attempt to incrementally reduce the error between the expected and generated states by adjusting synaptic weights of the readout layer (part of the striatum model) only. A new information that fits an existing schema will be encoded by the reservoir model in an area of the state space that contains state transitions learnt earlier. The required synaptic weight modification will be less important in this case and will be more important with a new information this is not yet related to consolidated (pre-existing) schema. Property 2 is also implemented by the online learning rule of the consolidation model. A challenging new information could be assimilated to an ambiguous state transition of the reservoir model, occurring when at least two overlapping snippets represent the same prefix 2D trajectory and bifurcate in two different direction. A balanced representation of the two snippets by the replay model will result in an ambiguous state transition that could be observed through multiple random walks. The existing schema is modified as the new ‘branch’ of the possible trajectories is represented and learnt. If this new branch is

overrepresented, the readout synaptic weights allowing the prediction of the old schema are progressively modified until the new schema is learnt. Finally, property 3 is implemented by the use of neural networks that have an intrinsic generalization ability. We demonstrated that the overlapping part of different snippets would be encoded in the reservoir neurons state space in similar areas. This allows the model to establish new paths between previously unrelated sub-paths.

Computational psychiatry

The prospective memory mechanism described in Figure 7 panel C was not investigated in this thesis. However, the simulation of future events contributes to the formation of plans and predictions. In (Schacter, Addis, and Buckner 2008b), the authors review “neuroimaging, neuropsychological, and cognitive studies that have examined future-event simulation and its relation to episodic memory” and “consider the applications of this work for research concerning clinical populations suffering from anxiety or depression in which pathological future thinking is a central feature”. By extending the model developed in this thesis, one could simulate past and future events associated to positive or negative outcome and simulate the ability of a subject to envision positive events. Other authors proposed to explain hysteria (Edwards et al. 2012) and psychotic symptoms (Adams et al. 2013) in terms of false inferences or beliefs. They “use a neurobiologically informed model of hierarchical Bayesian inference in the brain to explain functional motor and sensory symptoms in terms of perception and action arising from inference based on prior beliefs and sensory information”. The computational model developed in this thesis extended with a prospective memory mechanism and neuromodulation mechanism could be an implementation of a hierarchical Bayesian temporal inference model that are of major interest for the emerging discipline of computational psychiatry as illustrated in (Klaas E. Stephan, Diaconescu, and Iglesias 2016; Valton et al. 2017). Computational psychiatry has suddenly made it possible to mine data from long-standing observations and link it to mathematical theories of cognition. It’s also become possible to develop computer-based experiments that carefully control environments so that specific behaviors can be studied in detail (MIT 2017).

Appendix

Communications

Prefrontal Cortex Reservoir Network learns to reconstruct navigation sequences from random snippets replay

Nicolas CAZIN^{1,a}, Jean-Marc FELLOUS^{2,b}, Alfredo WEITZENFELD^{3,c}, Peter Ford DOMINEY^{1,d}

¹Stem cells and Brain Research Institute, INSERM U846, Lyon, FRANCE

²BioRobotics Lab, University of South Florida, Tampa, USA

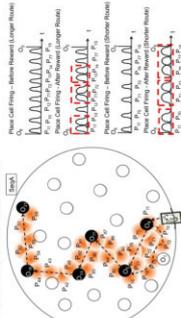
³Department of Psychology, University of Arizona, Tucson, USA

^anicolas.cazin@inserm.fr, ^bfellous@email.arizona.edu, ^cawetzenfeld@ust.edu, ^dpeter.dominey@inserm.fr



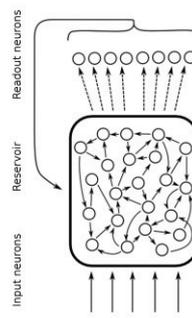
Statement

Recent studies ([1], [2]) show that spatial navigation in the rat involves the replay of place cells during awake and sleep states in the hippocampus. These small sequences of spatially related place cells are called snippets and occur primarily during sharp-wave-ripple (SPWR) events. We focus on the role of replay during the awake state, as the animal is learning across multiple trials. We hypothesize that the snippets can be used by the prefrontal cortex (PFC) in order to achieve multi-goal spatial sequence learning.



Reservoir computing

The joint modelling of a PFC/Hippocampus is able to form integrated spatial navigation sequences based on snippet replay and requires a biologically plausible sequence learning model. We use a computational framework known as reservoir computing [3]:



It is a computational metaphor of the cortico-cortical and cortico-striatal loops. The information is dynamically processed through reverberation by a large pool of neurons having fixed connections. The principle is straightforward:

1. Input neurons fire and induce a high-dimensional time dependent pattern on the reservoir's neurons
2. The expected output is reconstructed by reading the reservoir's neurons and correcting the synaptic weights in order to minimize the quadratic error between the actual output and the expected output.

Classical Least Mean Square (LMS) algorithm based on a generalized Widrow-Hoff learning rule is used. RLS algorithm is used for faster convergence.

Complex sequence learning

A stimuli sequence is said to be complex when transitions between elements are ambiguous (i.e. an element might have multiple predecessors and/or successors). Each element of the sequence is repeated 10 times before switching to its successor. The effective sequence length is 440 elements.

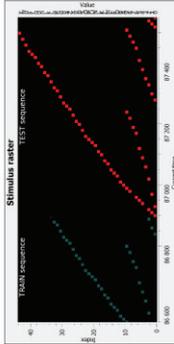


Figure 3: Stimuli used for training and testing. The reservoir naturally keeps track of the history of previous stimuli, thanks to the recurrent connections. Dynamics of the reservoir are typically very rich and depend on both current stimulus and its context.

Principal components analysis

Principal Component Analysis (PCA) is a standard dimension reduction technique useful for viewing the states of the reservoir as trajectories in a 2D space. Each similar stimulus is coded by the reservoir in a similar area of the reservoir's state space. As expected, the context provided by the recurrent connections induce slight differences in the encoding of similar stimuli in different contexts.

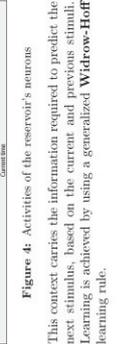


Figure 2: Trajectories of snippets. The context fading induced by the significant delay between snippets ensures that the trajectory of a given snippet is the same, in particular, when the previous snippet is not the same because of the random replay. For a given context length, a sufficient overlap between snippets will ensure a proper consolidation of the whole trajectory.

Snippets

During the navigation task, the hippocampus fires small sequences of previously encountered place cells in a random order. This replay occurs at a faster timescale and the delay between each snippet is significant. When a snippet occur, the context is no longer related to the previously replayed snippet. This is modelled by the random replay (sampling a uniform distribution) of overlapping snippets.

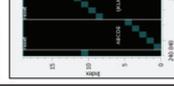


Figure 1: Snippets and whole sequence stimuli. Each snippet is separated by a reset of the network and a white noise is added before the input layer. Each element of each snippet is repeated 10 times and the whole sequence is never presented during the training process.

Noisy conditions

Surprisingly, the reservoir model is able to perform the snippets learning task under very noisy conditions. A strong uniform noise ($SNR \approx -80dB$) is added before the input layer. Each stimulus is repeated only 2 times and the reservoir leakage is tuned to 0.8 in order to avoid the time smoothing effect of a short term memory with a significant influence of the history.

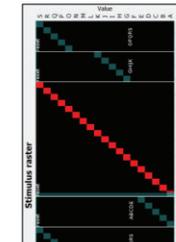


Figure 6: Snippets consolidation under noisy conditions. Even if the consolidation suffers from noise, it is still reliable and the serial order and timings are preserved. Only the intensities of the place cells are altered.

Meta-representation is reliable

Once learned at a timescale adapted to the stimuli period, the whole sequence could be replayed at different timescales. 12 orders of magnitude were successfully tested.

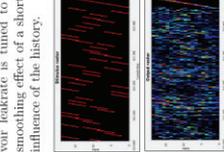


Figure 9: Snippets consolidation and generation of the whole sequence at different timescales. Modifying the leakage of the reservoir's neurons is the equivalent of modulating their resistive and capacitive properties.

Sparse snippets replay

The snippets consolidation task is even possible when considering sparse snippets. In the following example, 1 blank has been inserted at a random place in each snippet.

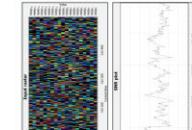


Figure 7: Sparse snippets consolidation. The underlying whole sequence is consolidated until the end and some blanks are present since there was no special fine tuning of the fading memory.

Consolidation of a complex sequence

The snippets consolidation property of a reservoir still applies for more complex sequences and is not limited to ascending only sequences.

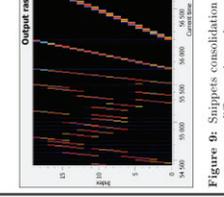


Figure 8: Snippets consolidation of a complex sequence.

Conclusion

We demonstrated that our reservoir model is able to consolidate snippets of a complex sequence. It naturally aligns overlapping subsequences of stimuli in the high-dimensional reservoir neuron's state space and this meta-representation is reliable across at least 12 orders of magnitude. Experiments show that this consolidation is still possible under noisy and sparse conditions. Moreover, consolidation of more complex sequences of place cells is still possible with the same model and the same parameters. This is the first step toward a joint prefrontal cortex/hippocampus model. The next step will involve the alignment of multiple complex sequences and we will study the ability of the reservoir to consolidate the shortest trajectories.

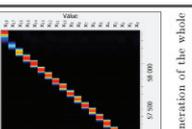


Figure 9: Snippets consolidation and generation of the whole sequence at different timescales. Modifying the leakage of the reservoir's neurons is the equivalent of modulating their resistive and capacitive properties.

References

- [1] Alejandra Barrera, Gonzalo Tejeda, Martín Llofriu, and Alfredo Weitzenfeld. A reservoir computing model of the hippocampus. *Spatial Cognition & Computation*, 13(1):27-59, 2015.
- [2] Laurel Watkins de Jong, Brian Gerde, Gerard M Martin, and Jean-Marc Fellous. The traveling salesman: insights into the dynamics of efficient spatial navigation in the rodent. *Journal of Neural Engineering*, 8(6):06010, 2011.
- [3] Peter Ford Dominey. Complex sensory-motor sequence learning through reservoir computing. *Neuroinformatics*, 10(1):1-11, 2012.

Acknowledgements

Supported by :
 • NSF-ANR CRCNS Project Spaquence
 • EU FP7 Project WYSIWYD

Neurodynamics of minimal visual and written narrative comprehension

Peter Ford DOMINEY^{1,2,3}, Anne-Lise JOUEN^{1,2,3}, Sullivan HIDDOT^{1,2,3}, Nicolas CAZINI^{1,2,3}, Carol MADDEN^{1,2,3}, Jocelyne VENTRE-DOMINEY^{1,2,3},
 1)INSERM U846 Stem Cell and Brain Research Institute, 2) University of Lyon, 3) Human and Robot Cognitive Systems

Introduction

Narrative comprehension requires the integration of successive elements into a consistent narrative structure. Successive elements that follow a predictive narrative structure should require less processing to integrate than those that are incoherent with respect to the evolving narrative structure. We recently demonstrated (Jouen et al., 2015 Neuroimage) the existence of a common extended semantic network recruited in the comprehension of visual scenes, and sentences. fMRI and DTI revealed the spatial organization of this network and aspects of its connectivity. The current research addresses temporal neurodynamics involved in the same kind of sentence and scene comprehension.

Material & Methods: Minimal Narrative

We recorded 64 channels scalp surface EEG from 18 native subjects in a protocol where they saw a visual scene (or read a sentence) depicting a human event, and after a 1-1.5 second pause saw a second scene (or sentence) always in the same modality that was either a coherent discourse follow-up, or not, of the first scene (or sentence). 30 trials per experimental condition were performed for each subject.

Objectives

Our objective was to test the predictions (P1) that comprehension of the second stimulus would require additional processing for incoherent vs. coherent stimuli, and that this would be the same for sentence and scene processing, reflecting the operation of the common meaning system, and (P2) that this processing would be manifest by a late positivity as has been recently attributed to the integration of mental representation of what is being communicated (Brouwer et al., 2012 Brain Research).

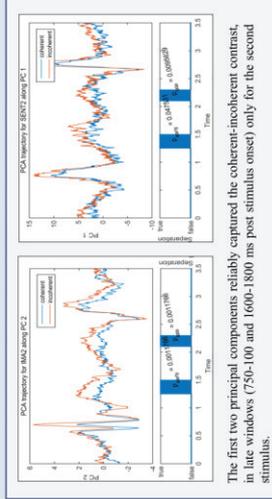
			
Image Coherent	Image Incoherent	Sentence Coherent	Sentence Incoherent

Results:

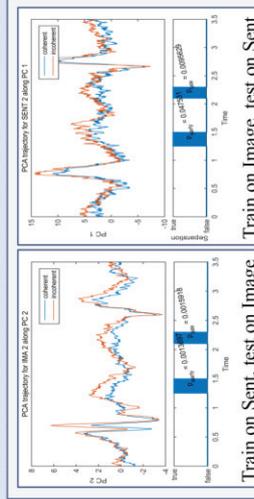
Principal Component Analysis

To characterize the neural activity, we performed PCA on the signals from the 64 electrodes over all conditions (coherent vs. incoherent), modalities (sentence vs. image) and stimuli (first and second) confounded, starting from 500ms prior to stimulus onset, to 1000 ms post offset. We then examined the data for specific conditions, modalities and stimuli in the resulting PCA space.

PCA Discovers the Coherence Effect

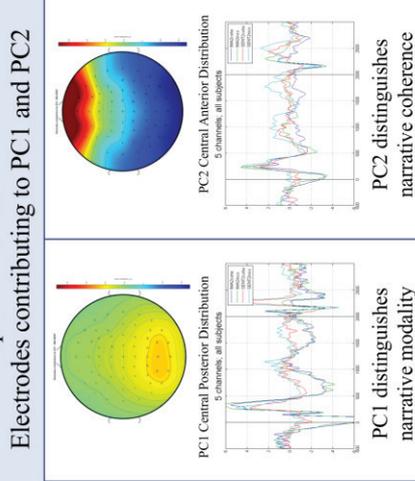


Cross-Validation in Sentence and Scene Modes



In order to determine if the same mechanisms are involved in the sentence and image comprehension, we then performed PCA only on the sentence data, and observed coherent vs incoherent trials in this PCA space for sentences and images. Again, the PCs reliably distinguish coherent vs. incoherent. The same effects were observed for the reverse operation, that is, extracting PCAs using image data, and testing on sentences and images.

Spatial Distribution of Electrodes contributing to PC1 and PC2



Summary of results:

- Unsupervised PCA processing discovered the neural correlates of precisely the experimental parameters that we manipulated – the discourse coherence of the second stimulus, in sentence and image discourse conditions.
- Cross validation revealed that the PCs identified in one modality allow a reliable “decoding” of the coherent/incoherent status in the other modality.
- Traditional ANOVA analysis of electrodes contributing to PC2 reveal a significant late positivity for INCOHERENT second stimuli in sentences and images.

We interpret the late positivity as an index of the processing required to generate a narrative representation consistent with the two successive stimuli.

Discussion

- Late positivities have been associated with syntactic violation or integration difficulty.
- In the current study, late positivities are evoked in the absence of syntactic complexity.
- Rather, the late positivities are evoked in conditions of increased narrative/discourse integration processing.
- This is consistent with Brouwer’s (2012) MRC Hypothesis, that the P600 component is a family of late positivities that reflect the word-by-word construction, reorganization, or updating of a mental representation of what is being communicated.
- We reveal for the first time a common neural process for narrative integration in both sentence and image modalities.

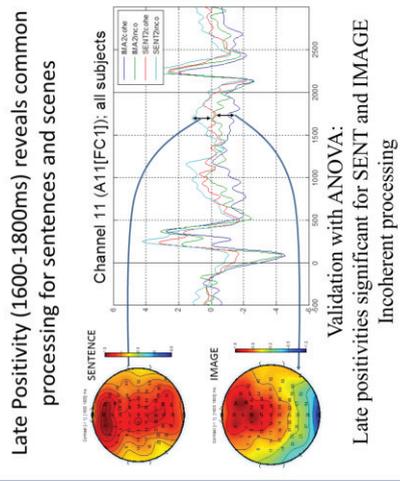
This new perspective can be situated in the context of narrative. In this context, when faced with incoming information about people and their interactions, we attempt to construct a narrative representation that “makes sense” (Bruner 1990). The processing of the second stimulus can be characterized in terms of this sense making. Our coherent stimuli will tend to be more easily integrated whereas our incoherent stimuli, rather than being processed as anomalies, would be processed so as to make sense in the context defined by the first stimulus. Our experimental coherent/incoherent manipulation would correspond to a notion of respectively relatively less or more distance or effort required to establish the narrative that would encompass the two stimuli.

References

Brouwer, H., Fitz, H., & Hoeks, J. (2012). Getting real about semantic illusions: rethinking the functional role of the P600 in language comprehension. *Brain Research, 1446*, 127-143.
 Bruner, J. S. (1990). *Acts of meaning*: Harvard University Press.
 Jouen, A., Elmore, T., Madden, C., Paller, C., Dominey, P., & Ventre-Dominey, J. (2015). Beyond the word and image: characteristics of a common meaning system for language and vision revealed by functional and structural imaging. *NeuroImage, 106*, 72-85.

Acknowledgements

Supported by NSF-ANR CRCNS Project Spaquene & EU FP7 Project WYSIWYD



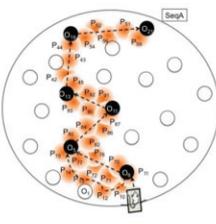
Prefrontal cortex reservoir network learns to create novel efficient navigation sequences by concatenating place-cell snippets replayed in hippocampus

N.CAZIN^{1,2}, M.LLOFRIU ALONSO³, P.SCLEIDOROVICH CHIODI³, T.PELC⁴, A.WEITZENFELD³, J-M.FELLOUS⁴, P.F.DOMINEY^{1,2}

¹INSERM U1208/SBRI, Bron, France; ²Univ. de Lyon, Univ. Lyon 1, Lyon, France; ³Computer Sci. and Engin., Univ. of South Florida, Tampa, FL; ⁴Psychology, Univ. of Arizona, Tucson, AZ

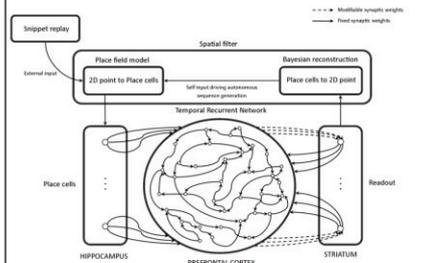
Introduction

Spatial navigation in the rat involves the **replay** of place cell subsequences (snippets) during awake and sleep states in the **hippocampus** during sharp-wave-ripple (SPWR) events ([1], [2]). We focus on the role of replay during the **awake** state, as the animal generates **increasingly efficient trajectories** between reward sites, across multiple trials. This trend toward optimal solutions is referred to as the **Traveling Salesman Problem (TSP)**. We hypothesize that **snippet replay** allows recurrent dynamics in prefrontal cortex (PFC) to **consolidate** snippet representations to create novel efficient sequences in TSP, by **rejecting** training sequences that are **less robustly coded** in the input.



Temporal Recurrent Network for sequence learning

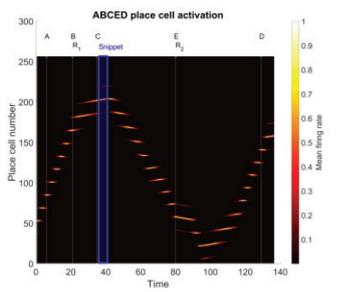
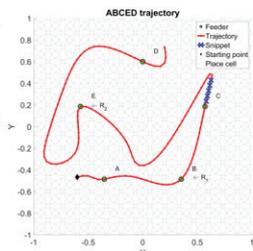
Prefrontal cortex is modeled as an instantiation of the **reservoir computing** paradigm called the Temporal Recurrent Network (TRN) [3], a computational metaphor of the **cortico-cortical** (reservoir) and **corticostriatal** (readout) circuits. Inputs excite a pool of neurons having fixed **recurrent** connections, generating high dimensional dynamics through **reverberation**. The principle is straightforward:



1. Input neurons (hippocampal place cells - HPCs) fire and induce a **high dimensional time dependent activity** pattern on the PFC reservoir's neurons
2. The desired output (next location coded HPCs) is **reconstructed** in readout neurons (striatum) by reading the reservoir's neurons and correcting the synaptic weights in order to **minimize the quadratic error** between the actual output and the expected output.
3. The predicted output feeds back (through a spatial filter) as self input driving autonomous sequence generation.

Snippet replay generates training input

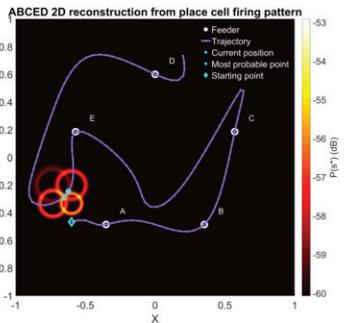
Snippet replay is modeled as generation of **random subsequences of place cell firing patterns** that occurred during previous trials. **Snippet length** (i.e. duration) is **fixed** and a **limited time budget** is allocated to snippet replay. Snippets are generated between random pairs of reward sites in previous trajectories. **Shorter paths** between rewarded sites are **more robustly coded** due to greater snippet overlap. Here is an example of a snippet of the ABCDE sequence. It lasts 5 simulation ticks and both spatial and place-cell representation are shown.



Spatial filter (reconstruction) during autonomous sequence generation

During autonomous sequence generation, HPC readouts code the next location in the sequence and feed-back to the input. To reduce noise in this feedback signal, we **reconstruct** the 2D location from a weighted average of **location hypothesis** carried by HPC [4] and then generate a corresponding HPC activation pattern coding this location to be **reinject**ed as input.

We propose the following method: Given the firing rate maps of 256 place-cells $\{A_i\}_{i=1 \dots 256}$ and a location s , place-cell responses $a_i=A_i(s)$ encode partial information about location s . Applying a **gaussian kernel** to A_i , centered on a_i will result in a map representing the probability of a 2D point s^* to generate the activation a_i .



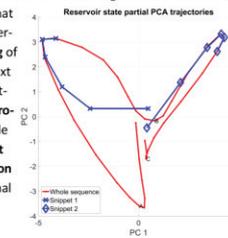
This map is noted $P(s^*|a_i)$ and will take values close to 1 for a_i values close to $A_i(s)$. Values close to 0 will be associated with points not related to $A_i(s)$. Finally, the map that **estimates the position** of the possible points s^* that generated the place-cell activation pattern (a_i) is given by:

$$P(s^*) = \sum_{i=1}^N P(s^* | a_i) P(a_i) \quad P(s^* | a_i) = K \cdot e^{-\frac{\|A_i(s^*) - a_i\|^2}{\sigma^2}}$$

Given a **prediction of a place cell activation pattern** and its associated **location probability map**, **computing** the most probable location and generating an HPC activation pattern coding this location results in significant **reduction** of accumulated **noise** in the HPC **feedback** signal.

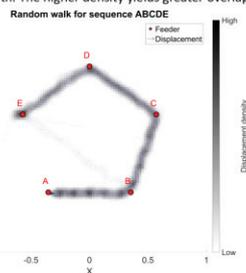
Trajectories are consolidated from random replay

TRN reservoir activation values over time can be represented as **trajectories** by using the principal component analysis (PCA). **Similar stimuli** are coded in **similar trajectories** in the reservoir's state space. The context provided by the recurrent connections induces slight differences in the encoding of similar sequences that have different preambles. A certain **time** is required for **fading** of this context. For a given context length, a **sufficient overlap** between snippets will ensure a **proper consolidation** of the whole trajectory. **Insufficient snippet overlap** results in **consolidation failure** (desired for non-optimal sequences).



Longer paths are rejected

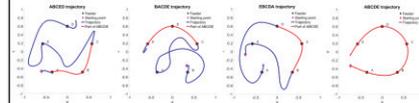
Two sequences joining sites ABE and ABCDE are replayed. Spatial distribution will be **sparse** between two rewarded sites linked by a **long path** and will be **denser** for a **shorter path**. The higher density yields greater overlap between snippets which favors them in competition during the learning process. Thus, for the same time of exposure to random replay, short paths between rewards will tend to be favored by greater snippet overlap.



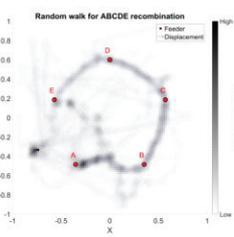
Condition	Test	Primed with ABE	Primed with ABCDE
No intermediate reward	ABE generated	ABE generated	ABE generated
Intermediate Reward	ABCDE generated	ABCDE generated	ABCDE generated

Novel efficient navigation sequence is created

Three different trajectories (ABCDE, BACDE, EBCDA) linking five reward sites contain parts of the **target sequence** ABCDE.

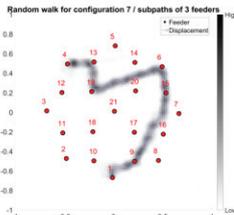


10 instances of the model are trained only with random snippets of sequences ABCDE, BACDE and EBCDA. After being primed with 10 samples of the ABCDE sequence, the model is able to **reject long subpaths** (CED, BAC, EB, DA) while **consolidating the short subpaths** (ABC, BCD, CDE) into the ABCDE sequence.



Spatial overlap is required

For a given feeder configuration, 9 different navigation trajectories contain subpaths of the target sequence. **Subpaths** are constrained to span over **2 feeders** in one condition and **3 feeders** in the other condition. Between each of the 9 first trials, the model is trained with snippets of the corresponding sequence. At trial 10, the model is exposed to snippets of the target sequence. > The model is able to **discover** the target sequence before trial 10 only when snippets span subpaths of 3 feeders.



Condition	2 feeders subpaths	3 feeders subpaths
Config 6	Trial 10	Trial 4
Config 7	Trial 10	Trial 7

Conclusion

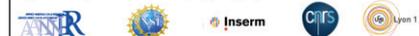
We demonstrated that the TRN reservoir model is able to **consolidate** randomly replayed snippets into efficient **navigation sequence**. It naturally **aligns** overlapping subsequences of stimuli in the reservoir **state space**. The limited **time budget** allocated to the random replay of snippets gives to the model the ability to **select** shortest paths between **reward sites**. Based on these two dynamical properties, we showed that the model is able to **consolidate** snippets of partially overlapping **multiple navigation sequences** in order to autonomously **generate novel efficient navigation sequences**. The current navigation model exploits a **spatial filter** to reduce noise in place cell predictions. It is based on place cells spatial response and it simulates a **perception/action** paradigm. The next version of the model will be implemented in a simulated and physical robotic **animat** that will take into account **embodiment** constraints. The online snippet replay model will be based on **electrophysiological** measurement of a rat **hippocampus**.

References

- [1] Alejandra Barrera, Gonzalo Tejera, Martin Llofriu, and Alfredo Weitzenfeld. Learning spatial localization: from rat studies to computational models of the hippocampus. *Spatial Cognition & Computation*, 15(1):27-59, 2015.
- [2] Laurel Watkins de Jong, Brian Gereke, Gerard M Martin, and Jean-Marc Fellous. The traveling salesrat: insights into the dynamics of efficient spatial navigation in the rodent. *Journal of Neural Engineering*, 8(6):065010, 2011.
- [3] Peter Ford Dominey. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological cybernetics*, (73):265-274, 1995.
- [4] Zhang, K., Ginzburg, I., McNaughton, B.L. & Sejnowski, T.J. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *Journal of neurophysiology* 79, 1017-44 (1998).

Acknowledgements

Supported by NSF-ANR CRCNS project **Spaquence** ANR-14-NEUC-0005-1





Prefrontal cortex reservoir network learns to create novel efficient navigation sequences by concatenating place-cell snippets replayed in hippocampus

N.CAZIN^{1,2}, M.LLOFRIU ALONSO³, P.SCLEIDOROVICH CHIODI³, T.PELC⁴, A.WEITZENFELD³, J.-M.FELLOUS⁴, P.F.DOMINEY^{1,2}

¹INSERM U1208/SBRI, Bron, France; ²Univ. de Lyon, Univ. Lyon 1, Lyon, France; ³Computer Scie. and Engin., Univ. of South Florida, Tampa, FL; ⁴Psychology, Univ. of Arizona, Tucson, AZ

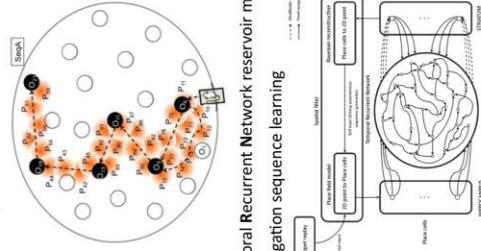
Introduction

Spatial navigation in the rat involves the **replay** of **place cells** subsequences (snippets) during awake and sleep states in the **hippocampus** during sharp-wave-ripple (SPWR) events ([1], [2]). We focus on the role of replay during the **awake** state, as the animal generates **increasingly efficient trajectories** between reward sites, across multiple trials. This trend toward optimal solutions is referred to as the **Traveling Salesman Problem (TSP)**. We hypothesize that **snippet replay** allows recurrent dynamics in prefrontal cortex (PFC) to **consolidate** snippet representations to create novel efficient sequences in TSP, by **rejecting** training sequences that are **less represented**.

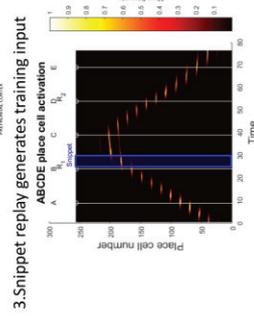
Temporal Recurrent Network Reservoir for sequence learning

Prefrontal cortex is modeled as an instantiation of the **reservoir computing** paradigm called the Temporal Recurrent Network (TRN) [3], a computational metaphor of the **cortico-cortical** (reservoir) and **corticostriatal** (readout) circuits. Inputs excite a pool of neurons having **fixed recurrent** connections, generating high dimensional dynamics through **reverberation**.

1. The traveling salesman

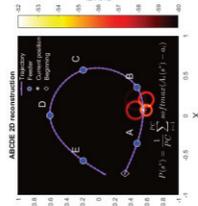


2. Temporal Recurrent Network reservoir model for navigation sequence learning

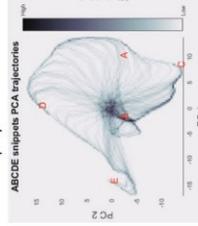


3. Snippet replay generates training input

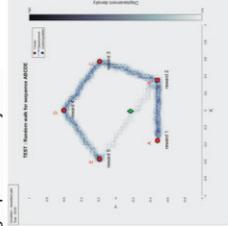
4. Spatial reconstruction during autonomous sequence generation



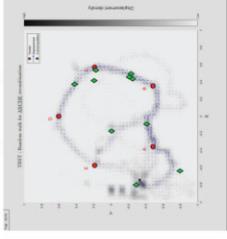
5. Trajectories are consolidated from random replay



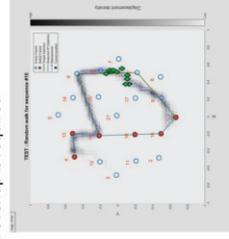
6. Longer paths are rejected



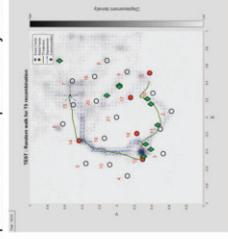
7. Novel efficient navigation sequence is created



8. Spatial overlap is required



9. Model predicts rat compatible trajectory



Conclusion

We demonstrated that the **TRN** reservoir model is able to **consolidate** randomly replayed snippets into efficient **navigation sequence**. It naturally **aligns** overlapping subsequences of stimuli in the reservoir **state space**. The limited **time budget** allocated to the random replay of snippets gives to the model the ability to **select** shortest paths between **reward sites**. Based on these two dynamical properties, we showed that the model is able to **consolidate** snippets of partially overlapping **multiple navigation sequences** in order to autonomously **generate novel efficient navigation sequences**. The current navigation model exploits a spatial filter to reduce noise in place cell predictions. It is based on place cells spatial response and it simulates a **perception/action** paradigm. The next version of the model will be implemented in a simulated and physical **robotic animal** that will take into account **embodiment** constraints. The online snippet replay model will be based on **electrophysiological** measurement of a rat **hippocampus**.

Acknowledgements

Supported by NSF-ANR CRCNS project **Spaquence ANR-14-NEUC-0005-1**



References

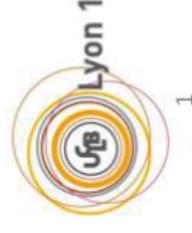
[1] Alejandra Barrera, Gonzalo Tejera, Martin Llofriu, and Alfredo Weitzenfeld. Learning spatial localization: from rat studies to computational models of the hippocampus. *Spatial Cognition & Computation*, 15(1):27-59, 2015.
 [2] Laurel Watkins de Jong, Brian Genke, Gerard M Martin, and Jean-Marc Fellous. The traveling salesman: insights into the dynamics of efficient spatial navigation in the rodent. *Journal of Neural Engineering*, 8(6):065010, 2011.
 [3] Peter Ford Dominey. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological Cybernetics*, (73):265-274, 1995.
 [4] Zhang, K., Ginzburg, I., McNaughton, B.L. & Sejnowski, T.J. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *Journal of neurophysiology*, 79, 1017-44 (1998).

Reservoir computing : A possible implementation for active inference ?

Nicolas CAZIN

ISTC-CNR

3/3/2017



Summary

- Problem statement
- Temporal recurrent network (TRN) model description
- Results
 - Selection and consolidation from random replay
 - Prediction of rat compatible trajectories
- Mapping TRN to active inference

A global optimization problem

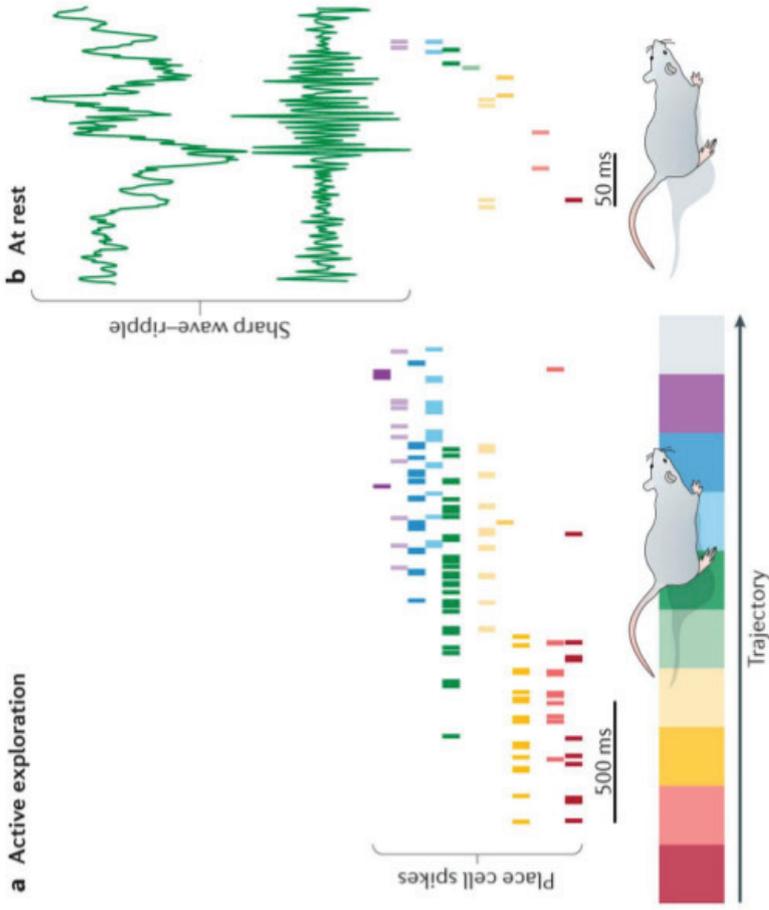
Traveling SalesRat

SG, TP, JMF - 2015

**Computational and Experimental
Neuroscience Laboratory
University of Arizona**

3

Replay of hippocampal place cells during SWR events



- Hippocampal replay is not a simple function of experience [2]
- We focus on forward replay
 - Random subsequences of recent experiences
 - Rewarded subsequences are replayed more often

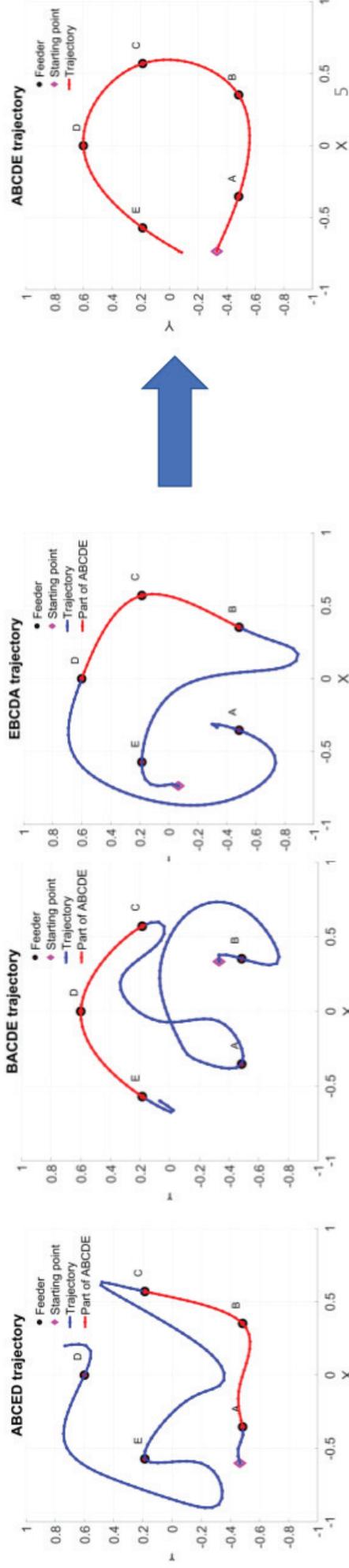
Nature Reviews | **Neuroscience**

Place cell firing pattern subsequences are replayed during rest state. Figure extracted from [1]

[1] Colgin, L. L. (2016). Rhythms of the hippocampal network. *Nature Reviews Neuroscience*.
[2] Gupta, A. S., van der Meer, M. A., Touretzky, D. S., & Redish, A. D. (2010). Hippocampal replay is not a simple function of experience. *Neuron*, 65(5), 695-705.

Consolidation from multiple trajectories

- Material and Methods : Random replay of 3 suboptimal sequences. Each contains
 - a part of the target sequence (ABCDE)
 - A prefix/suffix long sequence that links the remaining feeders together
- Results :
 - Concatenation of a new navigation sequence from random replay
 - Long paths rejection

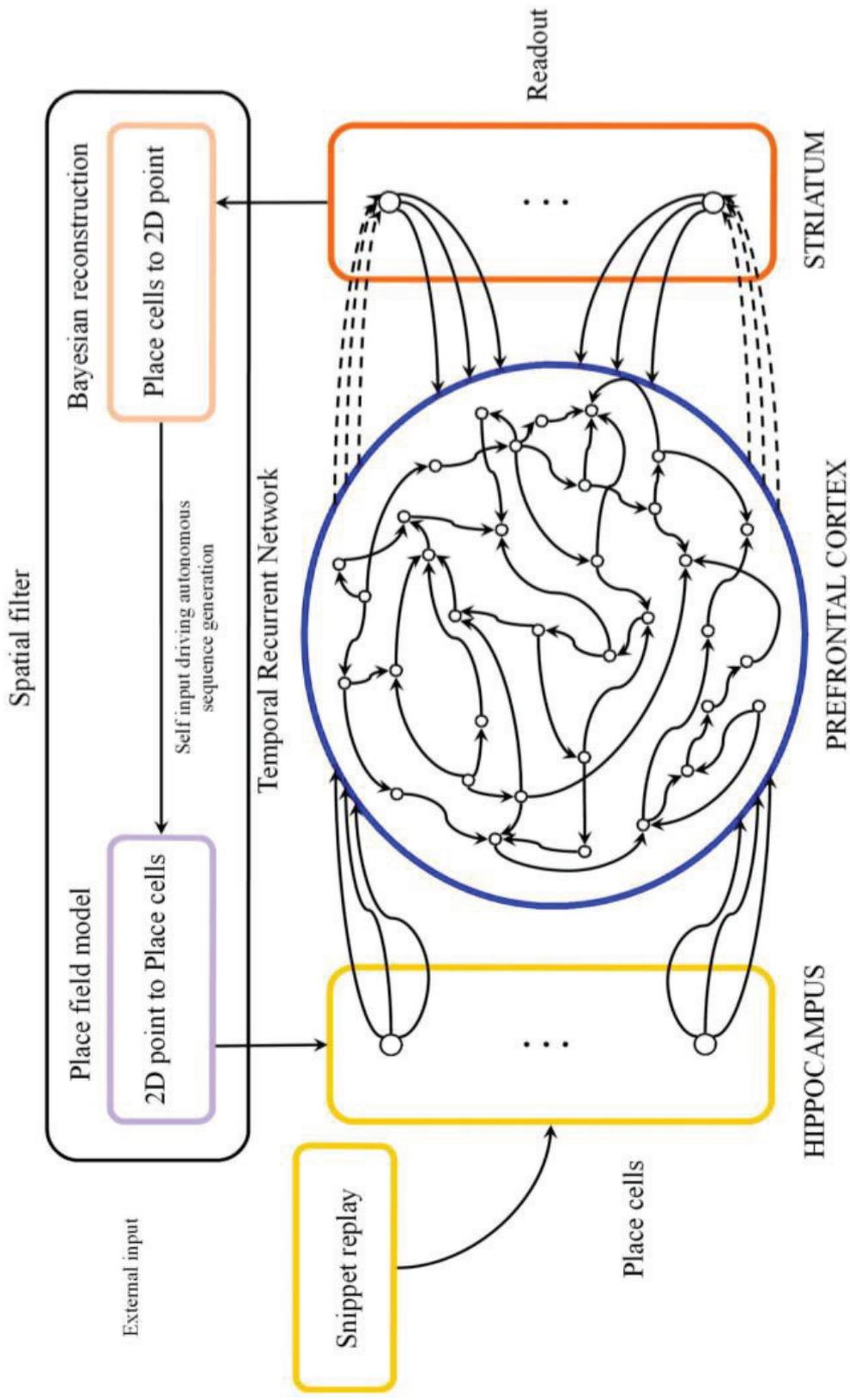


Summary

- Problem statement
- Temporal recurrent network (TRN) model description
- Results
 - Selection and consolidation from random replay
 - Prediction of rat compatible trajectories
- Mapping TRN to active inference

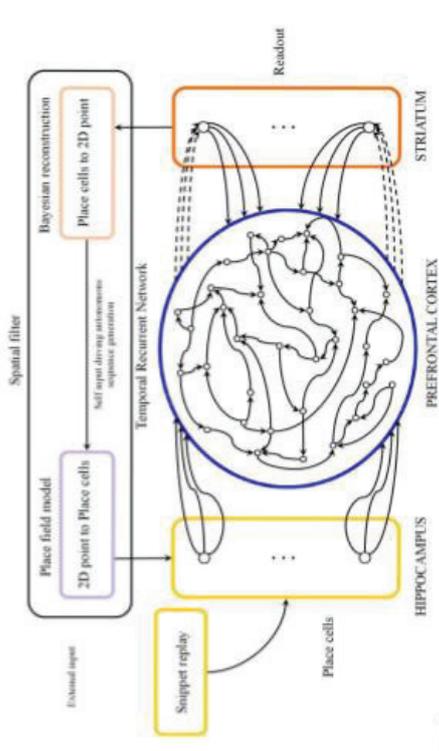
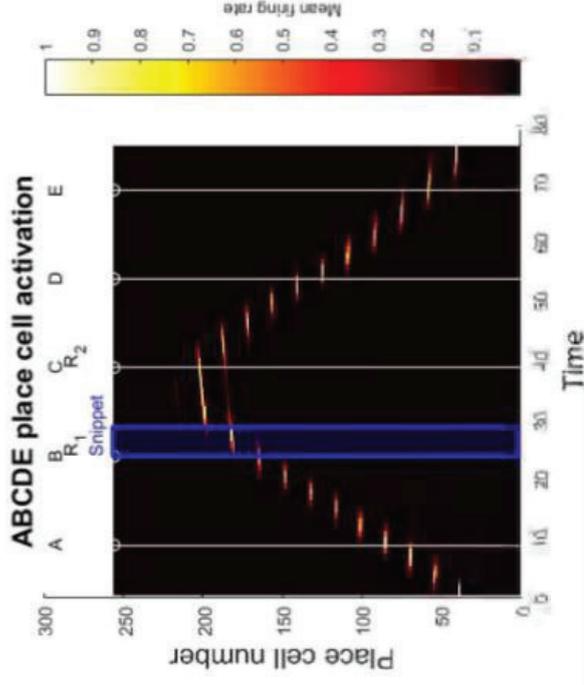
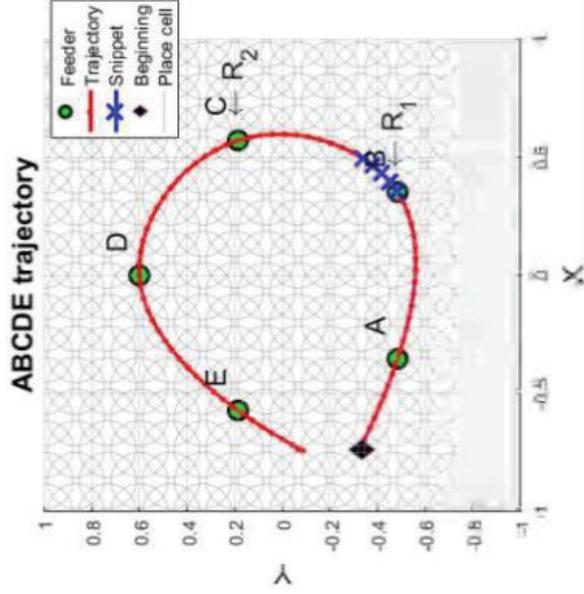
Reservoir model

- - - - -> Modifiable synaptic weights
 - - - - -> Fixed synaptic weights



Internally generated sequences

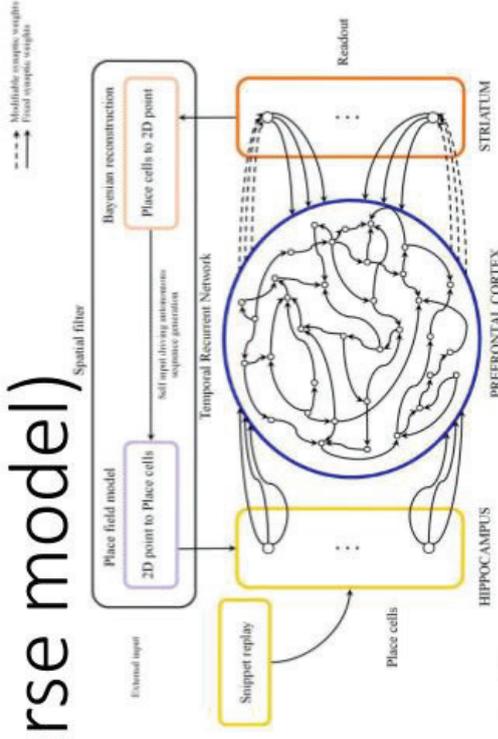
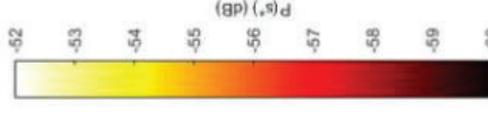
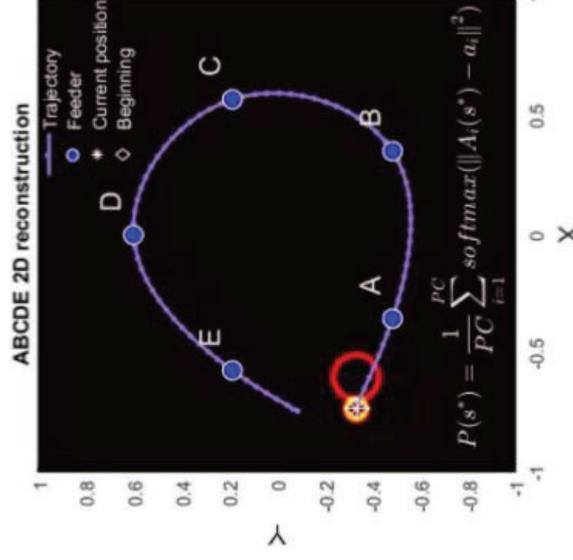
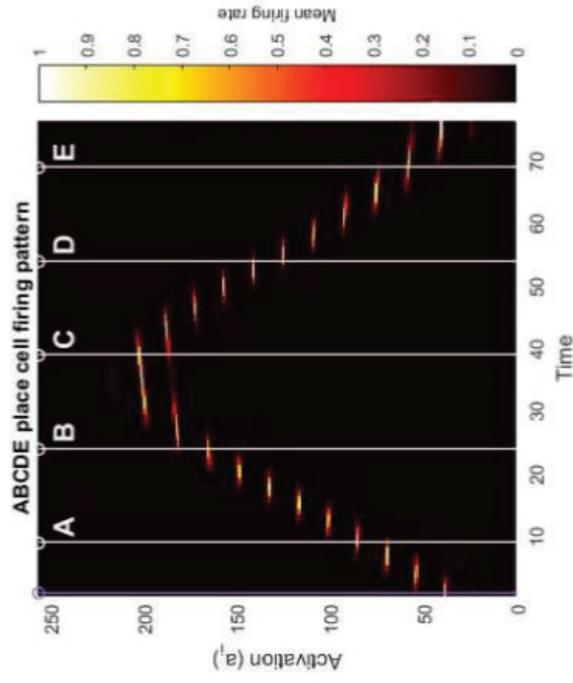
- Snippet generation method
 1. Draw 2 random reward sites
 2. Draw a random subsequence within this 2 reward sites



Spatial reconstruction during autonomous sequence generation (reverse model)

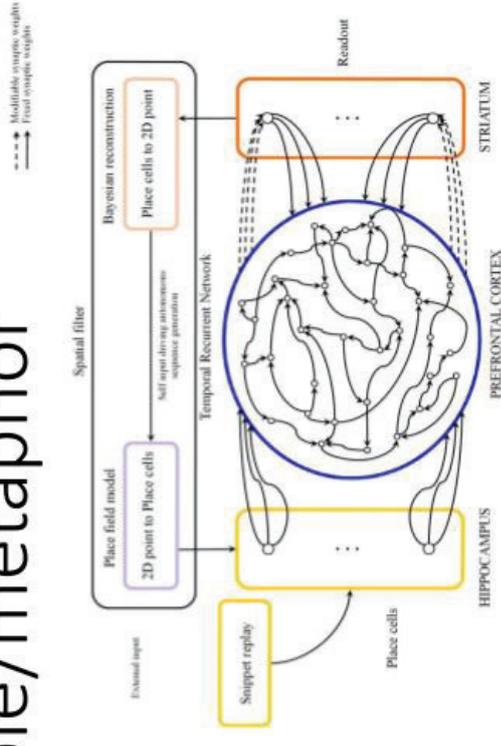
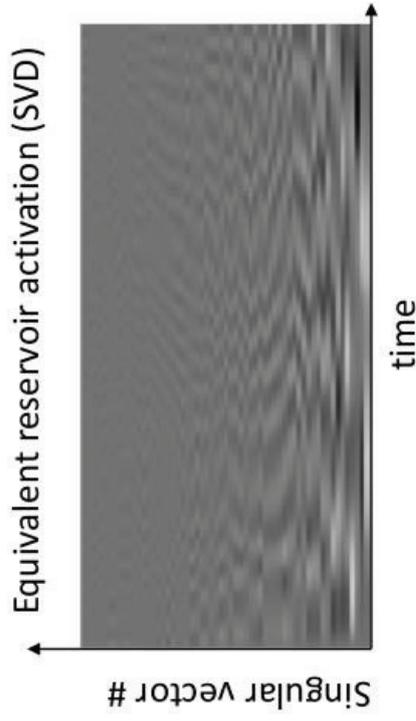
- Bayesian 2D location reconstruction from place cell firing pattern

1. Retrieve probable locations that generated a particular firing rate for a given place cell
2. Stack probable points hypothesis



Reservoir computing principle/metaphor

- Inputs excite a pool of leaky integrator neurons
 - Having **fixed recurrent** connections
 - Generating high dimensional dynamics through **reverberation**
- Reservoir activation pattern :
 - Interferencies of multiple sources reverberated through recurrent connections
- Readout learns to decode time congruent (correlated) expected signals



Summary

- Problem statement
- Temporal recurrent network (TRN) model description
- Results
 - Selection and consolidation from random replay
 - Prediction of rat compatible trajectories
- Mapping TRN to active inference

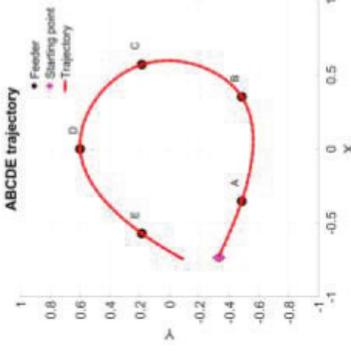
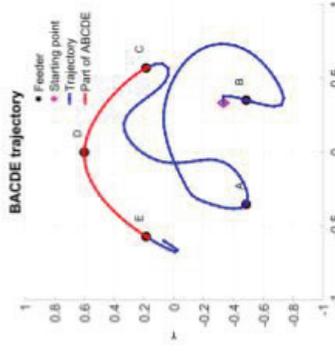
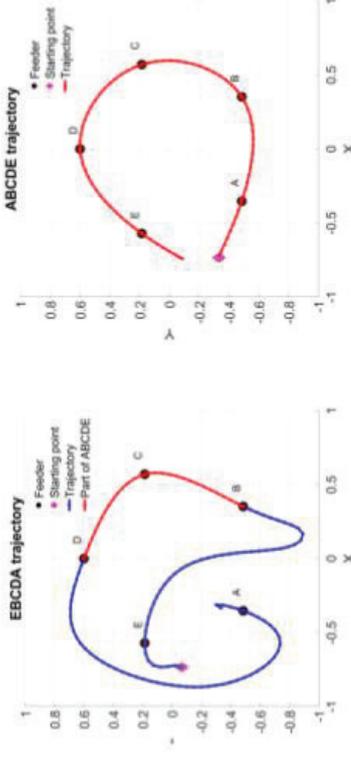
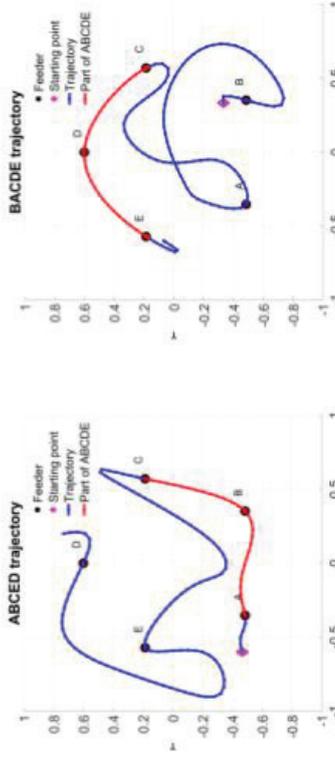
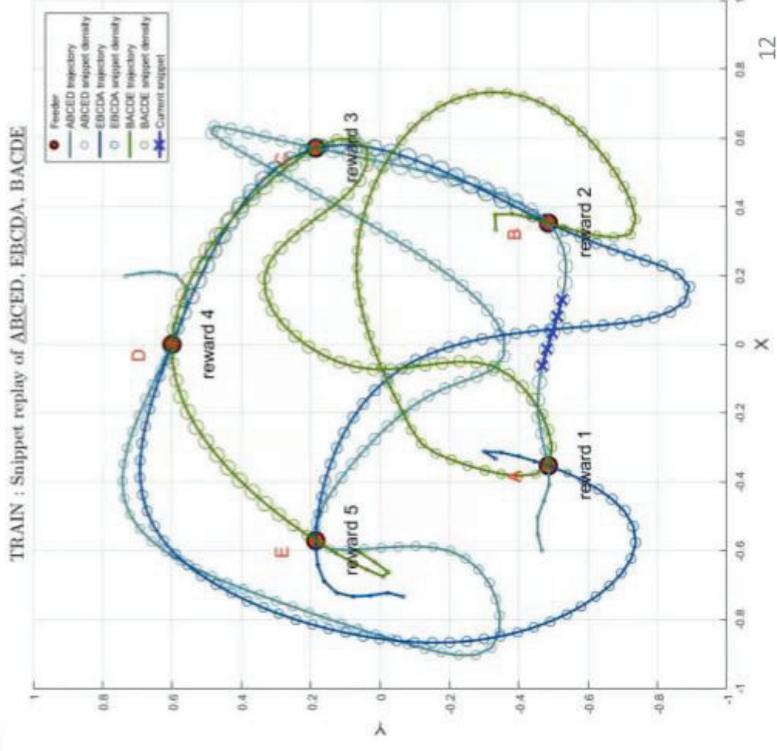
Novel efficient navigation sequence is created

Reservoir model is exposed to

- A random replay of three trajectories (recent experience)
- Derived from a target trajectory

Environment is sampled according to the agents updated beliefs

Trial 10/10



Model predicts rat compatible trajectory

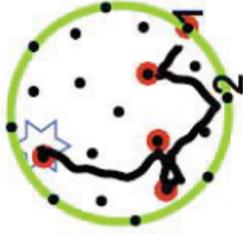
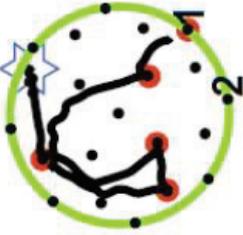
Trajectories extracted from a behavioral experiment with a rat
(CENL Lab, Fellous, Tucson AZ)

Conflicting trajectories result in an ambiguous decision

Trial: 10/10

T2: 3.90 m traveled, 1.59 (1.45, 1.72) ratio, **T3:** 2.58 m traveled, 1.31 (1.18, 1.69) ratio,
26.42 tot-time, 19.04 mov-time

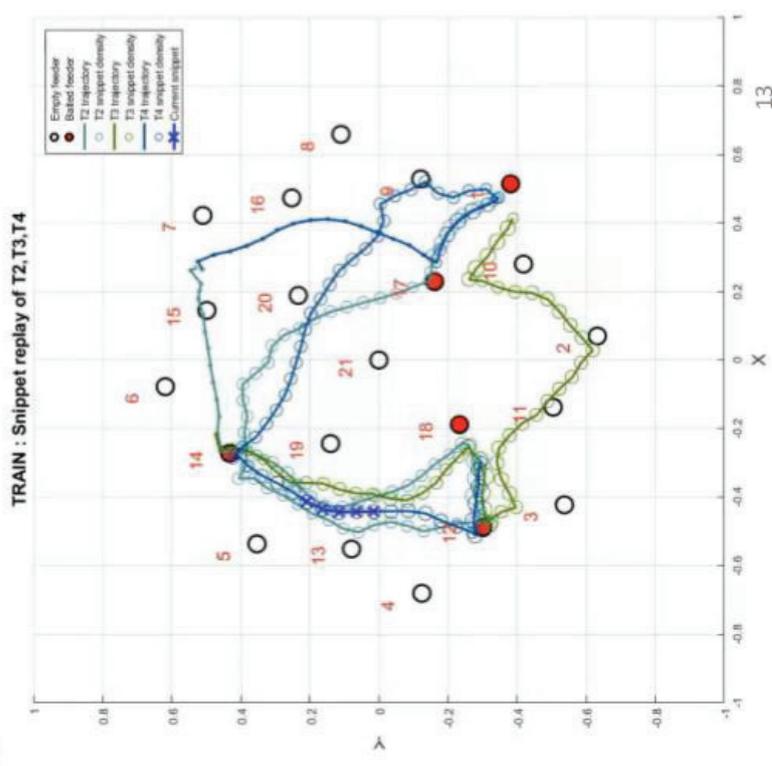
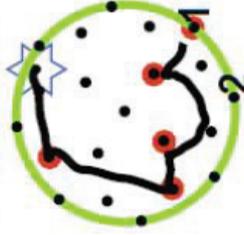
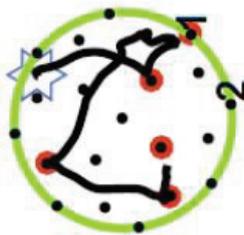
14 18 12 14 17 1 (5/5, 3) - 14 12 18 17 1 14 18 12 18 17 1 (5/5, 4) - 14 12 18 17 1



T4: 3.52 m traveled, 1.44 (1.29, 1.25) ratio, **T5:** 2.85 m traveled, 1.16 (1.04, 1.27) ratio,
29.58 tot-time, 15.08 mov-time

17 1 14 12 18 (5/5, 3) - 14 12 18 17 1

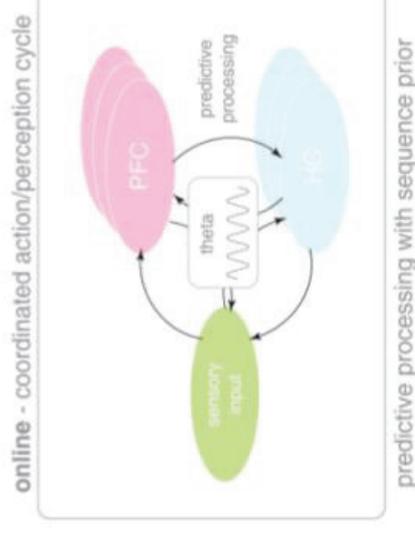
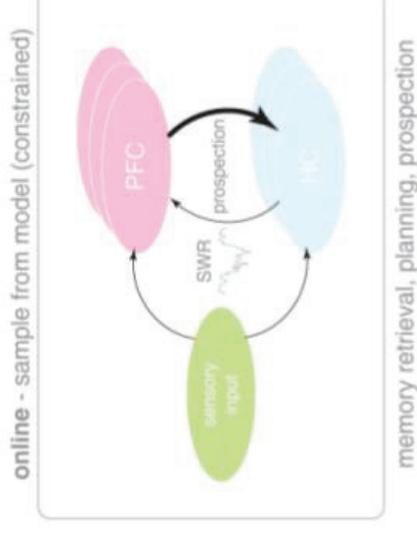
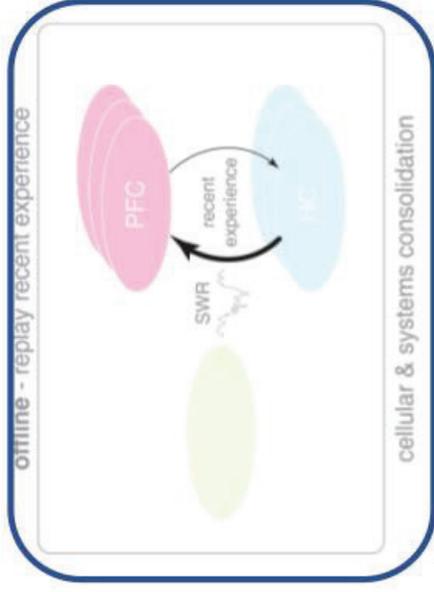
14 12 18 17 1 (5/5, 4) - 14 12 18 17 1



Summary

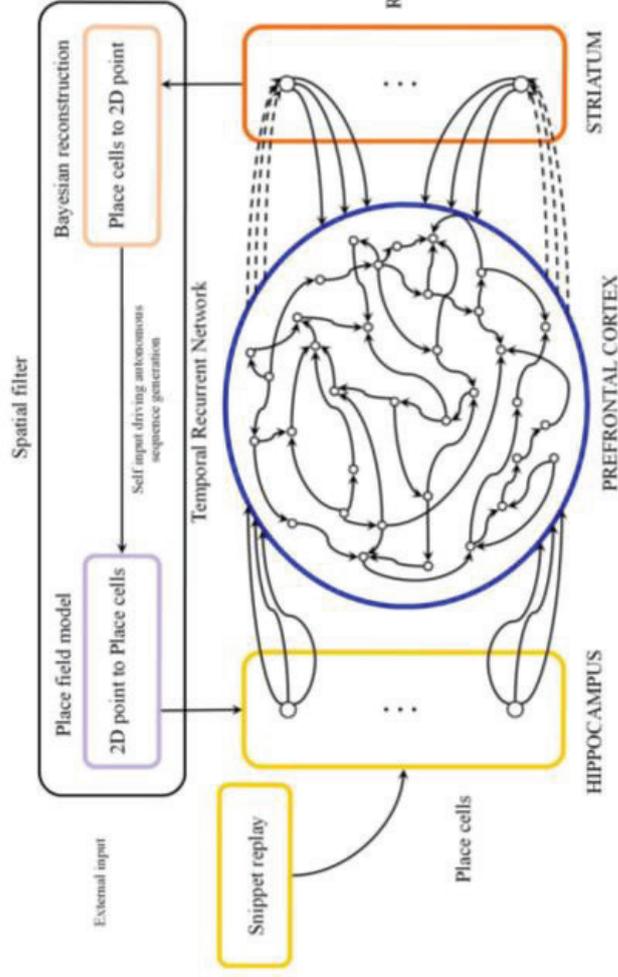
- Problem statement
- Temporal recurrent network (TRN) model description
- Results
 - Selection and consolidation from random replay
 - Prediction of rat compatible trajectories
- Mapping TRN to active inference

Mapping TRN to active inference



Mapping TRN to active inference

---> Modifiable synaptic weights
 -> Fixed synaptic weights



(solutions to) Belief updating

Action selection (and Bayesian model averaging)

$$U_t = \min_{\theta_t} \theta_t \cdot \zeta_{t+1}^H$$

$$\zeta_{t+1}^H = \ln \mathbf{A} \mathbf{s}_{t+1} - \ln \mathbf{A} \mathbf{B}(\mu \mathbf{B}_t)$$

$$\mathbf{s}_t = \sum_{\pi} \pi_t \cdot \mathbf{s}_t^{\pi}$$

State estimation (learning as inference)

$$\mathbf{s}_t^{\pi} = \sigma(\mathbf{A} \cdot \mathbf{o}_t + \mathbf{B}_{t-1}^{\pi} \mathbf{s}_{t-1}^{\pi} + \mathbf{B}_t^{\pi} \cdot \mathbf{s}_{t+1}^{\pi})$$

Policy selection

$$\pi_t = \sigma(\mathbf{F} - \gamma \cdot \mathbf{G})$$

$$F(\pi, \tau) = \mathbf{s}_t^{\pi} \cdot (\mathbf{s}_t^{\tau} \cdot \mathbf{A} \cdot \mathbf{o}_t - \mathbf{B}_{t-1}^{\tau} \mathbf{s}_{t-1}^{\tau})$$

$$G(\pi, \tau) = \mathbf{o}_t^{\tau} \cdot (\mathbf{o}_t^{\tau} - U_t) + \mathbf{s}_t^{\tau} \cdot \mathbf{H}$$

Precision (incentive balance)

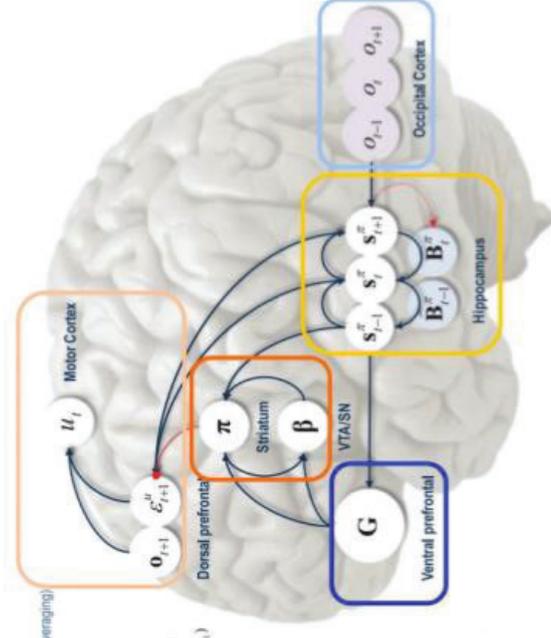
$$\beta = \beta + (\pi - \pi_0) \cdot \mathbf{G}$$

Learning

$$\mathbf{b}(t) = \mathbf{b}(t) + \sum_{\pi} \pi_t \cdot \mathbf{s}_t^{\pi} \odot \mathbf{s}_{t-1}^{\pi}$$

$$\mathbf{B} = \psi(\mathbf{b}) = \psi(\mathbf{b}^H)$$

Functional anatomy

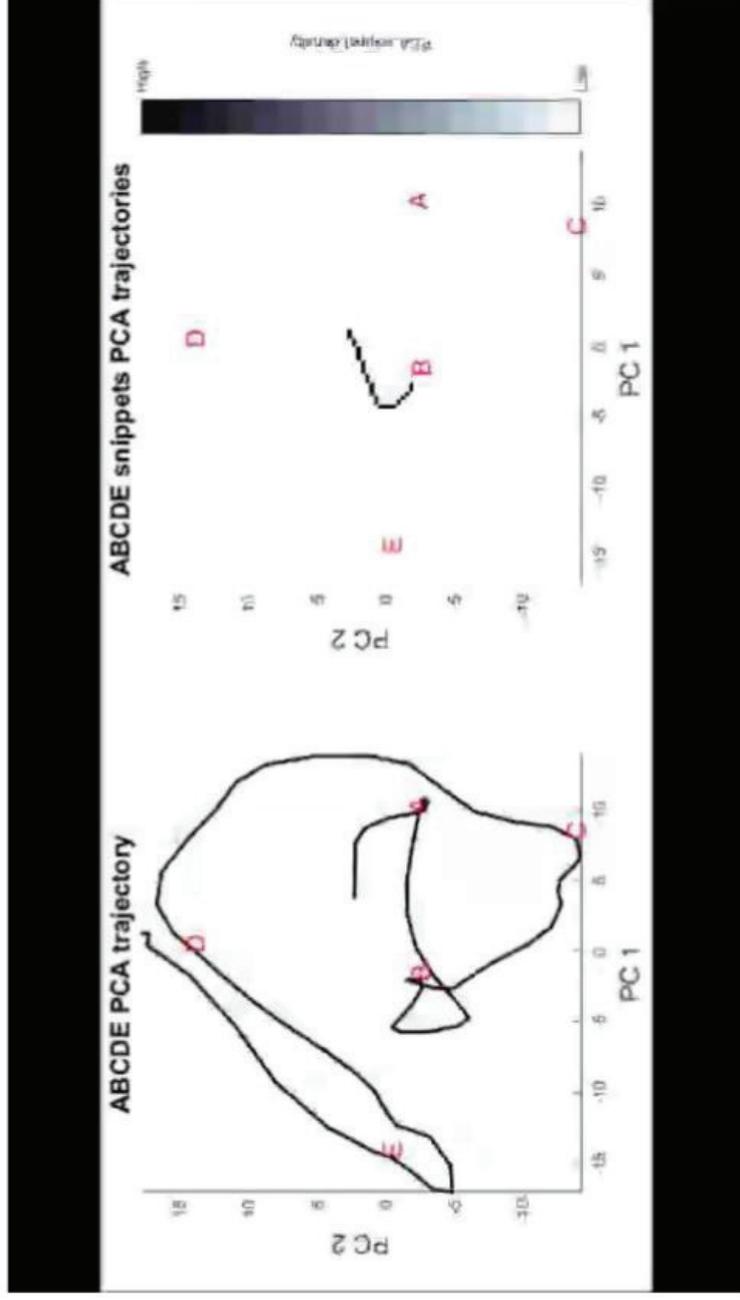


Mapping TRN to active inference

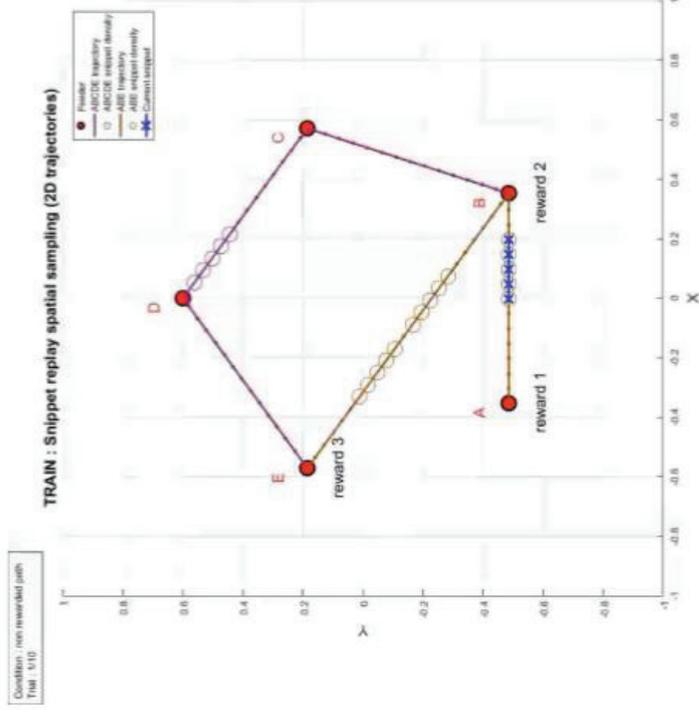
Modeled Area	Temporal recurrent network	Function	Generative model (consequences)	Perception (cause inferring)	Learning	Action
Occipital cortex	$s(t_n) \in S = \{(x(t_n), y(t_n)) \mid n \in [1, T]\}$	Sensory input				
Hippocampus	$x_{hippp}(t_n) = (A_i(s(t_n)))_{i \in [1, PC]} = (e^{-\frac{\ s(t_n) - \mu_i\ ^2}{\sigma_i^2}})_{i \in [1, PC]}$	State estimation under plausible policies	$y^{(2)} = g(v^{(2)}, \lambda^2)$ $v^{(2)} = s(t_n)$ $\lambda^{(2)} = (\mu_i, \sigma_i)_{i \in [1, PC]}$ $y^{(2)} = (A_i(s(t_n)))_{i \in [1, PC]}$			
Ventral PFC	$p(t_{n+1}) = p(t_n) + h * \frac{\partial p}{\partial t}$ $\frac{\partial p}{\partial t} = -p(t_n) + W_{fwd} * x_{hippp}(t_n) + W_{rec} * x_{res}(t_n) + W_{fwd} * x_{ro}(t)$ $x_{res}(t_n) = \tanh(p(t_n) - \theta)$	Evaluation of policies	$\dot{x}^{(3)} = f(x^{(3)}, \theta^{(3)}, v^{(3)}, \lambda^{(3)})$ $x^{(3)} = p(t_n)$ $v^{(3)} = [x_{hippp}(t_n), x_{res}(t_n), x_{stri}(t)] = [y^{(2)}, y^{(3)}, y^{(4)}]$ $\theta^{(3)} = [W_{fwd}, W_{rec}, W_{fback}]$ $\dot{x}^{(3)}$ $= -p(t_n) + W_{fwd} * x_{hippp}(t_n) + W_{rec} * x_{res}(t_n) + W_{fback}$ $* x_{ro}(t)$ $y^{(3)} = \dots$			
VTA/SN	$\varepsilon \in [0, 1]$	Precision				
Striatum	$x_{stri}(t_n) = W_{ro}(t_n) * x_{res}(t_n)$ $\frac{\partial W_{ro}}{\partial t} = (x_{des}(t_n) - x_{stri}(t_n)) * x_{res}(t_n)$ $W_{ro}(t_{n+1}) = W_{ro}(t_n) + \varepsilon * \frac{\partial W_{ro}}{\partial t}$	Policy selection				
Dorsal PFC	$P_{sensor}(s^*; t_n) = \frac{1}{PC} * \sum_{i=1}^{PC} softmax(\ x_{stri}(t_n) - A_i(s^*)\ ^2)$ $P_{next\ location}(s^* s; t_n) = \alpha * P_{transition}(s^* s; t_n) * P_{sensor}(s^*; t_n)$	Bayesian model average to next outcome				
Motor cortex	$x_{hippp}(t_{n+1}) = (A_i(\text{argmax}_{s^*}(P_{next\ location}(s^* s; t_n))))_{i \in [1, PC]}$	Predicted action				17

Thank you for your attention

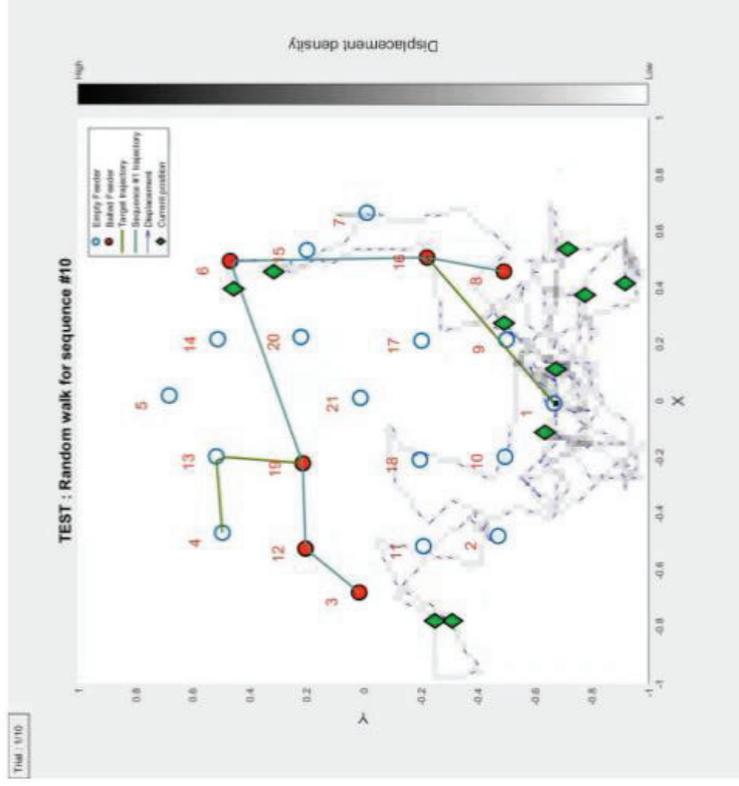
Trajectories are consolidated from random replay



Longer paths are rejected



Spatial overlap is required



Related work

Study on chaos in continuous time neural networks and application to motor control of a robot

Nicolas CAZIN, Mathias QUOY
 ETIS Laboratory, UCP/ENSEA/CNRS (UMR 8051), Cergy-Pontoise

18 février 2016

Résumé

1 Introduction

Artificial neural networks models are often divided among feed-forward networks (multi-layer perceptrons for instance) and recurrent neural networks (RNN), like Hopfield networks for instance [17]. The main property of RNN is their definition of *attractors*. Indeed, a RNN may be considered as dynamical system, and as such its dynamics exhibits various dynamical patterns, and converges towards different kind of attractors : stable fixed point, limit cycles, or even chaotic attractors [35, 40]. Therefore, since the seminal work by Sompolinsky et al. [39], work in the field of chaotic neural networks has been developing. In particular recurrent neural network model were proposed [?, 7, ?, 36]. In these models, the RNN is converging towards an attractor. Depending on their convergence scheme and attractors, we can define two different types of such RNN : 1- RNN where the parameters are fixed, and 2- RNN where the parameters may evolve through internal constraints (eg. learning) or external influence (eg. visual inputs). In the first case, parameters set *one* dynamical system. This dynamical system may have one or several attractors [17, 42, 36] **Samso à vérifier**. One of the drawbacks is the fact that once on the attractor, the systems stays on it. One way to avoid it, is to change the state of one or several neuron. This leads to change the initial condition of the dynamical system, and therefore gives the possibility to converge to another attractor if the initial condition is outside the attraction basin [17, 31]. One other way is to use the dynamical system as a *reservoir* of attractors, and learn of the links between the RRN and an output layer. This is the framework of *reservoir computing* [12, 28, 18]. In 2004 Echo State Networks proposed by Jaeger and Haas [18] gave a theoretical and practical framework for the learning of sequences using recurrent neural networks. In 2009, Sussillo and Abbott [41] proposed an extended learning scheme for ESN called FORCE learning. The last possibility is to use a chaotic itinerancy [45] through Milnor attractors, where the dynamics may escape the attractor ruin [46]. The difficulty is to define the attractors depending on what the RNN is used for. In Hopfield networks, the weights are computed beforehand by a combination of all pattern to be stored. The problem of learning patterns on-line is that the weights are parameters of the dynamical system. Thus if they are modified, the dynamical system is not the same anymore, and hence the attractors are different. Therefore, the system falls into the second category

where the parameters are modified. Changing the parameters of a dynamical system may lead to *bifurcations*. These bifurcations change the number and the nature of the dynamical system's attractors. Therefore, in the framework of ESN, Laje and Buonomano proposed to tune the recurrent connections for learning timing and moto patterns [?]. In an earlier work, Daucé et al. [9] proposed a discrete time recurrent neural network learning on the links between the recurrent net and the output, and within the recurrent net. It was based on the theoretical study of this kind of networks [7, ?]. This model was limited to discrete time neural networks. This article investigates how it may be extended to continuous time neural networks. We will focus here on firing rate models. Some work has been done on spiking models, although not formerly presented as chaotic [6, 48, 47, 21, 5].

2 Continuous time update

The definition of an asymetrical random recurrent neural network is given in [11] and its short name is *Resonnant Spatio-Temporal architecture (ReST)*. Let R be one instance of this neural network structured in K layers. Each $p \in K$ layer contains $N^{(p)}$ neurons. The $J^{(pq)}(t)$ matrix contains synaptic weights between the $q \in K$ layer and the $p \in K$ layer. $J_{ij}^{(pq)}$ is the synaptic weight on the link between the $j \in N^{(q)}$ neuron of layer $q \in K$ and the $i \in N^{(p)}$ neuron of layer $p \in K$. $J^{(pq)} = (J_{ij}^{(pq)})$ is a $N^{(p)} \times N^{(q)}$ matrix. A link is said to be modifiable if the weight may change. The figure 1 shows an instance of a two layer *ReST* network.

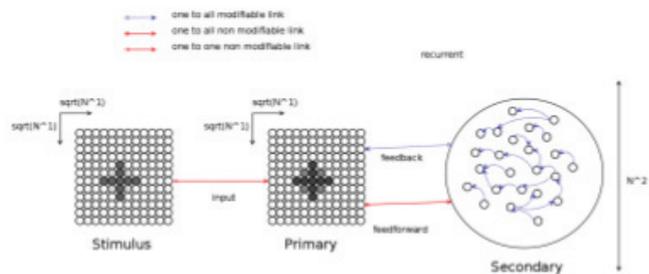


FIGURE 1 – *ReST* architecture

The incoming *stimulus* feeds the *primary layer* through a *one to one non modifiable link*. The *primary layer* is connected to the *secondary layer* through a *one to all non modifiable feedforward link*. The *recurrent link* in the *secondary layer* is

intended at capturing a spatio-temporal dependency of the *secondary layer's* states. The *feedback link* is a *one to all modifiable* link and is intended for associating a spatio-temporal state of the *secondary layer* to the current *stimulus* on the *primary layer*. The *feedforward* and *recurrent* synaptic weights are randomly set according to a gaussian distribution such as :

$$(1) \quad \mathcal{N}(\mu_j^{(pq)}, \sigma_j^{(pq)}) = \mathcal{N}(0, \frac{1}{\sqrt{N^{(p)}}})$$

With the condition :

$$- (J_{ij}^{(pq)} = 0)_{\forall i \in N^{(p)}, \forall j \in N^{(q)} | i=j}$$

Each weights of the *recurrent link* samples this distribution and is connected to others neurons but there is no recurrent connection for an individual neuron. The feedback link is initially set to 0 for each weights :

$$(2) \quad (J_{ij}^{(pq)} = 0)_{\forall i \in N^{(p)}, \forall j \in N^{(q)}}$$

The *feedforward link* connects each neuron of the *secondary layer* to every neurons of the *primary layer* according to random weights. Since the primary layer is spatially coherent¹, a neuron of the *secondary layer* may view the stimulus modified by the random weights as a random spatial feature extracted from the *primary layer*. Each neuron of the secondary layer is coding for a random feature of the primary layer and the *recurrent link* introduces a time dependency amongst the previous states of the *secondary layer*. Thus, the states of the secondary layer constitutes at each time a high dimensional spatio-temporal encoding of the low dimensional incoming stimulus. Each stimulus is associated by reinforcement² to the its previous encoding. This is clearly an on-line reservoir computing approach.

2.1 Local field

The variation of the *local field* at time t_n of the i neuron on the p layer when considering the incoming q layer is given by :

$$(3) \quad \tau_u^{(p)} \cdot \frac{\partial u_i^{(pq)}}{\partial t} = -u_i^{(pq)}(t_n) + \underbrace{\sum_{j=1}^{N^{(q)}} J_{ij}^{(pq)}(t_n) \cdot x_j^{(q)}(t_n)}_{c_i^{(p)}(t_n)}$$

Where :

- $u_i^{(pq)}(t_n)$: is the value of the local field at time t_n .
- $\tau_u^{(p)}$: is the time constant all local fields of supporting layer p .
- $N^{(q)}$: is the number of neurons contained in the q incoming layer.
- $J_{ij}^{(pq)}(t_n)$: is the synaptic weight between the i neuron of the p layer and the j neuron of the q layer.
- $x_j^{(q)}(t_n)$: is the state of the j neuron of the q layer at time t_n .
- $c_i^{(p)}(t_n)$: is the instantaneous contribution of the incoming synaptic weights $(J_{ij}^{(pq)}(t_n))_{j \in N^{(q)}}$

1. the stimulus is a picture

2. modified / time delayed hebbian learning rule

Basically, it is a generalization of the leaky integrator neuron model described in [38]. Biologically speaking, the $\tau_u^{(p)}$ time constant is equivalent to the product of the resistivity and capacity properties of the membrane. The neurons time constant is the same for all units. We use the Euler's method for solving (3). Let $f^{u_i^{(pq)}}$ be the auxilliary function evaluating the first order derivative of the local field $u_i^{(pq)}$ $i^t h$ component at time t and point l :

$$(4) \quad f^{u_i^{(pq)}}(t, l) = \frac{\partial u_i^{(pq)}}{\partial t} |_{t=\tau_n, l=u_i^{(pq)}(\tau_n)}$$

Let $h_u^{(p)}$ be the infinitesimal time increment between two evaluations of the auxilliary function (4)

$$(5) \quad t_{n+1} = t_n + h_u^{(p)}$$

Where :

$$- h_u^{(p)} = \frac{1}{\tau_u^{(p)} \cdot N_u^{(p)}}$$

- $N_u^{(p)}$ is the number of points contained in a sampling period³

The time update equation of the local field solved by the Euler's method is given by :

$$(6) \quad \begin{aligned} u_i^{(pq)}(t_{n+1}) &= u_i^{(pq)}(t_n + h_u^{(p)}) \\ &= u_i^{(pq)}(t_n + h_u^{(p)}) \cdot f^{u_i^{(pq)}}(t_n, u_i^{(pq)}(t_n)) + O(h_u^2) \\ &\approx u_i^{(pq)}(t_n) + \frac{1}{N_u^{(p)}} \cdot u_i^{(pq)}(t_n) + c_i^{(p)}(t_n) \\ &\approx \underbrace{\left(1 - \frac{1}{N_u^{(p)}}\right)}_{h_u^{(p)}, \text{ past}} \cdot \underbrace{u_i^{(pq)}(t_n)}_{\text{past}} + \frac{1}{N_u^{(p)}} \cdot \underbrace{c_i^{(p)}(t_n)}_{\text{instantaneous}} \end{aligned}$$

In particular, $N_p = \tau_u^{(p)} = 1$ leads to the discrete update scheme. Clearly, it is a convex combination between the past and the instantaneous values.

2.2 Membrane potential

The *membrane potential* at time t_n of a i neuron on the p layer is the sum of its local field contributions :

$$(7) \quad p_i^{(p)}(t_{n+1}) = \sum_{q=1}^P u_i^{(pq)}(t_{n+1}) - \theta_i^{(p)}$$

Where :

- $\theta_i^{(p)}$: is the threshold of the i neuron of the p layer

2.3 Activation function

The activation function of each neuron is a sigmoid function given by :

$$(8) \quad f_a(x; g) = \frac{1}{1 + e^{-2 \cdot g \cdot x}}$$

Where :

- x : is a value in \mathbb{R}
- g : is the slope parameter of the function

3. i.e. the $\tau_u^{(p)}$ time constant

2.4 State

Each neuron i of layer p computes its *activation state* according to its membrane potential

$$(9) \quad x_i^{(p)}(t_{n+1}) = f_a(p_i^{(p)}(t_{n+1}) + I_i^{(p)}; g)$$

Where :

- $I_i^{(p)}$: is the i^{th} coordinate of the I stimulus vector feeding the p layer

2.5 Synaptic weights

The learning rule is a temporal version of the hebbian learning rule. Instead of learning co-occurency between presynaptic neurons⁴ and postsynaptic neurons⁵, this rule is base on the covariance of signals propagating through layers. The equation is the following :

$$(10) \quad \tau_J^{(pq)} \cdot \frac{\partial J_{ij}^{(pq)}}{\partial t} = \underbrace{\frac{\Phi_g(p_i^{(p)}(t_n))}{N^{(q)}}}_{modulation} \cdot \underbrace{\epsilon^{(pq)} \cdot d_i^{(p)}(t_n) \cdot d_j^{(q)}(t_{n-1})}_{hebbian}$$

Where

- $d_i^{(p)}(t_n) = x_i^{(p)}(t_n) - \hat{x}_i^{(p)}(t_n)$
- $d_j^{(q)}(t_{n-1}) = x_j^{(q)}(t_{n-1}) - \hat{x}_j^{(q)}(t_{n-1})$
- $\hat{x}_i^{(p)}(t_n)$: is the running average activation value i estimated at time t_n
- $\hat{x}_j^{(q)}(t_{n-1})$: is the running average activation value j estimated at time t_{n-1}
- $\Phi_g(p_i^{(p)}(t_n)) = 1 - f_a(p_i^{(p)}(t_n); g)$
- $\tau_J^{(pq)}$: is the sampling period of the evolution of the synaptic weights between layers p and q .
- $\epsilon^{(pq)}$: is a small positive constant called the *learning rate*.

$\Phi_g(p_i^{(p)}(t_n))$ is a function intended to allow learning to occur only when the postsynaptic neuron i is not saturated⁶. Only the so called **dynamic/fluctuating** neurons are interesting during learning. $d_i^{(p)}(t_n)$ is in fact the dynamical part of the signal of neuron i . It is related to the part of the signal fluctuating around the running average activation value $\hat{x}_i^{(p)}(t_n)$ by :

$$(11) \quad \hat{x}_i^{(p)}(t_{n+1}) = \beta \cdot \hat{x}_i^{(p)}(t_n) + (1 - \beta) \cdot x_i^{(p)}(t_n)$$

Where :

- β : is the habituation parameter and is chosen in the $[0, 1]$ interval

It is again a convex combination between novelty and past. The higher β is, the less the average estimation will take into account the the current state $x_i^{(p)}(t_n)$. The same explanation holds for $d_j^{(q)}(t_{n-1})$.

This learning rule will reinforce synaptic connections between each j neuron having a signal correlated at time t_{n-1} with each neuron i at time t_n [9]. The introduction of the synaptic propagation delay⁷ in the covariance of the activation signals is mandatory for learning a sequence of stimuli

4. neurons j of layer q

5. neurons i of layer p

6. activity close to 1

7. the time difference between t_n and t_{n-1}

varying from one step to another. The whole equation (10) is normalized by the size of the incoming layer $N^{(q)}$. This trick is intended to have an equivalent influency on the synaptic weights of the local field $u^{(pq)}$, for every value of $N^{(q)}$. The amount of modification of synaptic weights is kept the same. The drawback of this trick is that the learning in wide networks tends to be slower. Theoretical results are encountered in wide networks and the computational load is increased with realistic sizes. For simplicity purpose, synaptic weights time constant $\tau_J^{(pq)}$ is the same across the whole local field (pq) .

Let $f_{ij}^{J^{(pq)}}$ be the auxilliary function evaluating the first order derivative of the synaptic weight $J_{ij}^{(pq)}$ at time t and point l :

$$(12) \quad f_{ij}^{J^{(pq)}}(t, a) = \frac{\partial J_{ij}^{(pq)}}{\partial t} \Big|_{t=t_n, a=J_{ij}^{(pq)}}$$

Let $h_J^{(pq)}$ be the infinitesimal time increment between two evaluation of the auxilliary function (4)

$$(13) \quad t_{n+1} = t_n + h_J^{(pq)}$$

Where :

$$- h_J^{(pq)} = \frac{1}{\tau_J^{(pq)} \cdot N_J^{(pq)}}$$

- $N_J^{(pq)}$ is the number of points contained in a sampling period⁸

We use the Euler's method for solving (10) :

$$(14) \quad \begin{aligned} \Delta J_{ij}^{(pq)} &= J_{ij}^{(pq)}(t_n + h_J^{(pq)}) - J_{ij}^{(pq)}(t_n) \\ &= h_J^{(pq)} \cdot f_{ij}^{J^{(pq)}}(t_n, J_{ij}^{(pq)}(t_n)) + O(h_J^{(pq)2}) \\ &\approx \epsilon^{(pq)} \cdot \frac{\Phi_g(p_i^{(p)}(t_n))}{N^{(q)} \cdot N_J^{(pq)}} \cdot d_i^{(p)}(t_n) \cdot d_j^{(q)}(t_{n-1}) \end{aligned}$$

2.6 Error measure

The error measure used in this article is the instantaneous mean square error :

$$(15) \quad E_r^{(pq)}(t_n) = \frac{1}{N^{(p)}} \cdot \sum_i^{N^{(p)}} (I_i^{(q)}(t_n) - F_i^{(pq)}(t_n))^2$$

Where :

- $F_i^{(pq)}(t_n) = f_a(u_i^{(pq)}(t_n); g)$ is the i^{th} coordinate of the predicted signal.

- $I_i^{(q)}(t_n)$ is the i^{th} coordinate of the input stimulus.

The prediction is extracted from the local fields connecting the secondary layer q to the output layer. It may be seen as what the output would be without the incoming stimulus used for hebbian learning.

A network r depends on a random seed. The error is measured among several networks in order to have a statistically robust learning metric. The mean and standard deviation of this instantaneous error is extracted from these networks and reported on a graph such as the one in figure 2. This measure

8. i.e. the $\tau_J^{(pq)}$ time constant

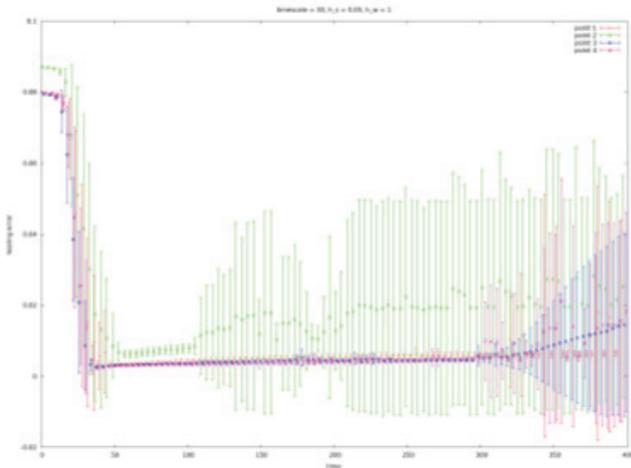
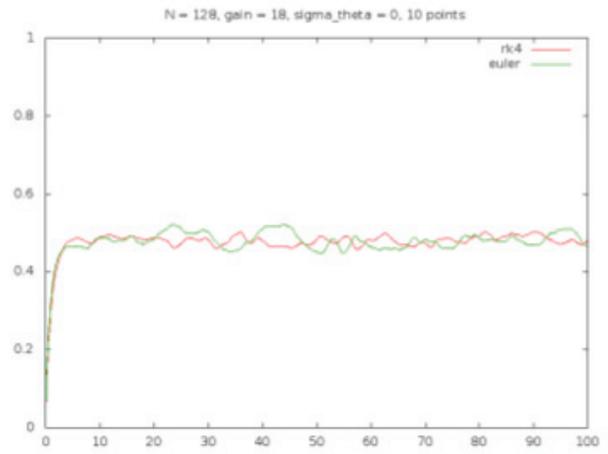


FIGURE 2 – Mean/Standard deviation of the instantaneous learning error

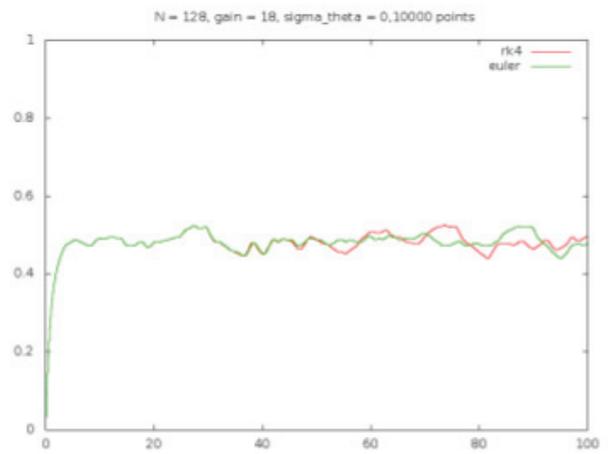
is done after the states are updated and before the weights are adapted. Thus, when a stimulus is driving the network, the error is evaluating the learning done during the previous cycle. Without this extra care, it would be very easy to report good learning error with an arbitrarily important learning rate. In this particular case, the network will overadapt its weights in order to fit the current stimulus. The error will not be zero even if the learning process is held for a long time. The sigmoid function is often very stiff because of the chaotic dynamics requirements and the image of the stimulus on the primary layer may be saturated. In the case where the learning leads to a perfect match between the predicted stimulus and the next stimulus projected onto the input layer, the error will not be zero when comparing the prediction against the non saturated stimulus.

2.7 Local truncation error

Every numerical integration scheme involve a local truncation error. Euler's method is about $O(h_u^2)$. Due to the recurrent nature of RRNN, the error committed at a given time step is reinjected at the next time step. The figure 3 depicts the effect of the reinjection of local truncation error.



(a) $h_u = 0.1$



(b) $h_u = 0.0001$

FIGURE 3 – Comparaison between Euler explicit et Runge-Kutta 4th order methods with different h_u values.

The Euler's method is compared to the classical RK4 method for each subfigure. The network is exactly the same, including initial conditions and weighting. The subfigure 3a reports the mean activation values of the secondary layer over time without stimulus and $h_u = 0.1$. The subfigure 3b shows the effect of increasing the time resolution⁹ : both update scheme bifurcate after a given time. This time is increasing when parameter g is increased. Fast variation of neural activity arise and the resulting drift overcomes faster. Decreasing h_u will delay this drift phenomenon but it cannot be avoided. The wise way to handle this problem is to consider both h_s and h_w as parameters of the RRNN. The numerical integration scheme is a part of the parameter set too.

2.8 Population update policy

For simplicity purposes, this study is restricted to the synchronous update case in a pseudo hierarchical fashion. Each layer is updated from the stimulus layer to the secondary layer.

9. practically by decreasing h_u

2.9 Stiffness

A stiff system may be described by several coupled partial differentiate equation evolving at different timescales. In this study, we will consider the update equations of the weights (10) and states (9). The problem arises when an important variation of a weight occurs at a fast time scale. The state variation induced by this fast weight variation may not be perceived by the network state equation. The time increment is too coarse and the leaky integrator neuron is damping this quick variation. In this study, we do not want to take into account the stiffness of the equation system by using a state of the art integration scheme with variable step. The damping effect may be interesting and there is no restriction for a fast time scaled network to drive a slower time scaled network. As stated in section 2.7, we will use the forward Euler's method for the leaky integrator model.

3 Simulation

3.1 Limitations of the original model

3.1.1 Arbitrary timescale

The original *ReST* model is described in [11] with a very specific scheduling when updating states, weights and stimuli. One stimulus update induce one state update for all neurons group in a feedforward way and this state update is followed by an update of the modifiable synaptic weights. Let's note it $stimulus_1 \Rightarrow states_1 \Rightarrow weights_1$. This scheduling is very restrictive and does restrict any physical world experiment since stimuli and network do not necessary evolve along the same timescale. Each element of the update scheme may be refreshed at a different rate. A more general scheduling noted $stimulus_A \Rightarrow states_B \Rightarrow weights_C$ would allow us to be more flexible in order to embed the *ReST* architecture into a robot. A simple try of the original model with a stimulus remaining the same during 10 cycles shows the limitations of the original *ReST* network at handling different timescales. The figure 4 shows an increasing learning error :

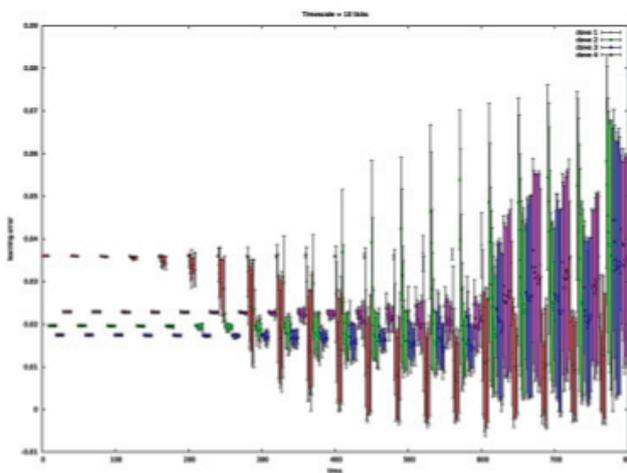


FIGURE 4 – Learning in original *ReST* architecture with a stimulus remaining 10 cycles

It may be explained by the fact that in this particular implementation and for a given stimulus timescale $T = 10$, the network will learn 9 times the transition ($spot1 \rightarrow spot1$) and one time the expected transition ($spot1 \rightarrow spot2$). Figure 4 depicts the networks inability to learn the ($a \rightarrow b$) transition while the sequence ($a \rightarrow a \rightarrow b \rightarrow b$) is driving the network. A simple solution to deal with arbitrary stimulus timescale is to introduce a transition detector based on the first order derivative. The network learning is triggered only when a rising edge is detecting a stimulus change. This is a ad-hoc solution but not so meaningless from a biological point of view. This is a rough novelty detection in the incoming stimulus and the enthorinal cortex is involved in the novelty detection. At least one criticism may be emitted : A local modification in the stimulus will trigger learning to the whole set of neurons. From one synapse point of view, it implies the knowledge of all variation of activities of the others neurons and this is not correct.

In fact, this triggering is built in the learning equation (10). Each dynamical part of the signal coordinate¹⁰ is in fact a novelty detector. For instance, lets consider the layer p and its i^{th} neurons state at time t_n , noted $x_i^{(p)}(t_n)$. When the state of this neuron differs from its running average $\hat{x}_i^{(p)}(t_n)$, $d_i^{(p)}(t_n)$ is not zero. In other words, the neuron's state is different from its habituation value. According to (10), learning occurs only when a difference between habituation and instantaneous values occurs at a given timescale. Since neurons from the secondary layer are recurrent, several paths of different length may exist and the dynamical part of the signal of a neuron won't fade at the same rate as other neurons dynamical part. That's why silent or saturated neurons time associations won't be learned and fixed points dynamics will restrict the learning timelapse. The network will learn until postsynaptics or presynaptics neurons reach their average value. This average value may be reached slowly or quickly, according to the value of the β parameter. Figure 5 depicts this phenomenon.

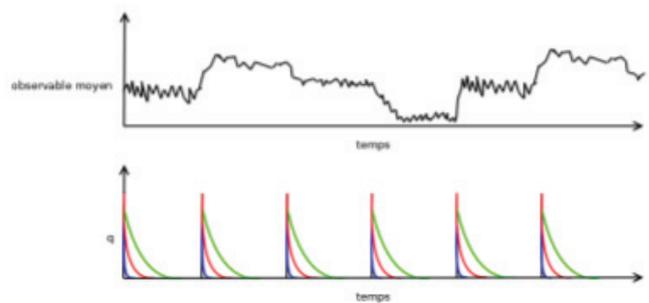


FIGURE 5 – Influence of β on the amount of modified weights during a learning cycle

The upper subfigure shows the mean observable measures¹¹ across time. A different attractor is reached when the corresponding stimulus drives the network. The subfigure

10. $d_i^{(p)}(t_n)$ and $d_j^{(q)}(t_{n-1})$

11. $\frac{1}{N^{(p)}} \cdot \sum_{i=0}^{N^{(p)}} x_i^{(p)}(t_n)$

below represents the total amount of modified weights during a learning cycle¹². Learning is effective only when the perception differs from the expectation of the neurons state. Large β values allow the neurons to capture slow transitions between stimulus/states of the secondary layer. Small β values will tend to focus the neurons learning on the beginning of the transitions.

3.1.2 Merged evocation and perception

The two layers model is built in a such manner that the stimulus $I^{(1)}(t_n)$ (Figure 6a) will be blended with the contribution of the local field $u_i^{(12)}(t_n)$ (Figure 6b). This may have annoying consequences on the expected output. When a sequence is learned during a significant amount of time, the local field contribution begins to hold the prediction of the next stimulus so strong that the it will evocate the prediction of the stimulus.

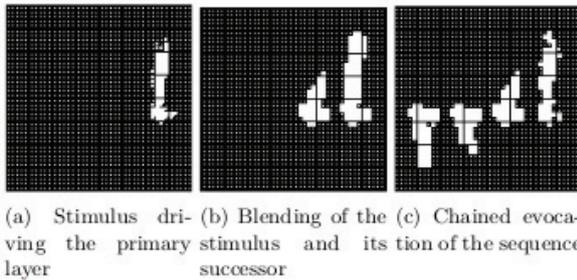


FIGURE 6 – Blending on the primary layer

If the network is exposed to learning for a longer period, the evocation overlaid on the primary layer will evocate the next prediction and so on (Figure 6c). In order to avoid this problem, one have to unfold the network by duplicating the primary layer and thus separate input from output.

3.2 Towards an independant dynamics model

In order to overcome the limitations of the original model and to give the network the ability to have an update frequency independant from the stimuli frequencies, the *ReST* architecture has been unfolded in the following way (Figure 7) :

12. $\sum_{i=0}^{N^{(p)}} \sum_{j=0}^{N^{(q)}} |J_{ij}^{(pq)}(t_{n+1}) - J_{ij}^{(pq)}(t_n)|$

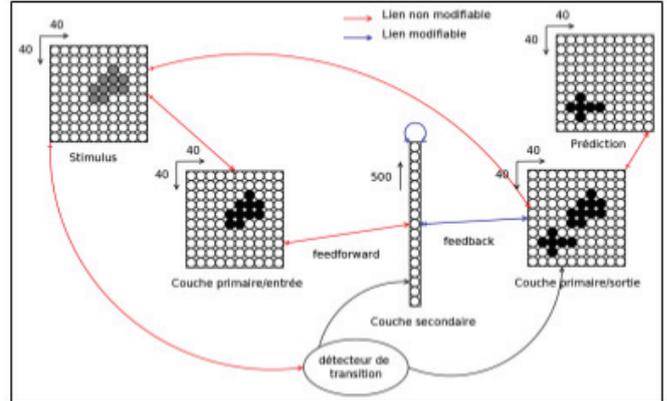


FIGURE 7 – The modified *ReST* architecture

The primary layer is now implemented by two layers. The output layer is the exact copy of the input layer. The initial conditions are the same. The feedback is extracted in the same way. The stimulus signal is applied to the output layer because :

- the feedback synaptic weights are initially set to zero
- the learning equation (10) implies hebbian style learning.

At the begining of the learning process, the post-synaptic activity is zero since the synaptic weights are zero. The hebbian style learning rule might be viewed as o product of pre-synaptic and post-synaptic activation states. A zero post synaptic term would obviously drive the equation (10) to zero. Since non zero post-synaptic activation states are required and the synaptic weights between pre-synaptic neurons and post-synaptic neurons are zero, the only way to induce non zero post-synaptic activation accordingly to the (9) equation is to drive the output layer with the input stimulus. Learning can occur in this way and the local field contribution will grow as the learning process is sustained. When the feedback local field will have a sufficient contribution, it may be able to overcome the neuron's threshold value and have a significant influence on the output layer's state. At this point of the learning process, the output layer will relate informations from the current input¹³ and the prediction of the next input¹⁴. The prediction is simply extracted by computing (10) without the contribution of $I^{(p)}$. Having non zero synaptic weights at the begining of the simulation is possible and in this case, driving the output layer with the input stimulus is no longer required. The activation values of the output layer will be driven by the secondary layer's states and will heavily rely on the initial weighting scheme. This architecture is not restricted to have an output layer similar to its input layer. It could be set to a layer with a different number of neurons. The only condition is to have a postsynaptic activity driven by the expected output. The learning process will capture any delayed co-activation between input stimulus and expected output. This network might be seen as a delayed associative memory.

The following experiments will use the same stimuli sequence depicted in Figure 8.

13. $I^{(p)}$

14. $p_i^{(p)}(t_{n+1})$

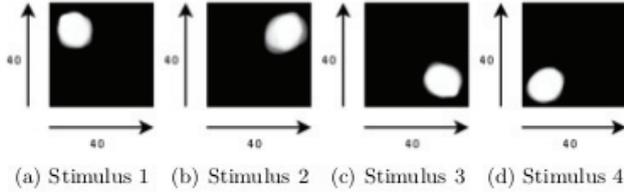


FIGURE 8 – Stimuli of the 'points' sequence

Note that the stimulus presentation frequency is steady and do not contains any jitter component. The robot experiment will allow to test the network ability to capture sequences with small phase fluctuating.

3.3 Effect of h_u

Theoretically speaking, decreasing h_u is equivalent to increasing the dominance of the local fields histories amongst their current values in (6). The right operand¹⁵ weighted by h_u have lesser importance. The equation is evolving as if the time constant $\tau_u^{(p)}$ in the equation (3) were raised. The network evolution is slowed down if we consider the update cycle time as constant. For the same cycle time, two networks set with a different h_s will compute a different amount of state variation. From a more experimental point of view, decreasing h_u will slow down the evolution of the network while the learning is still possible. In particular, learning will continue even if h_u is modified when learning. The only condition is that the network continues to fluctuate with a speed related to the stimuli sequence update speed. In this way, the corresponding attractor is reached before the next stimulus change and the learning process may take place and converges towards the correct prediction. In order to understand the effect of the h_u parameter on the learning process for a given stimulus timescale T , several learning were performed with different values of (h_u, T) . The figure 9 relates some of those cases.

15. $c_i^{(p)}(t_n) = \sum_{j=1}^{N^{(q)}} J_{ij}^{(pq)}(t_n) \cdot x_j^{(q)}(t_n)$

$f_s = \frac{1}{T}$ \ h_s	0.01	0.05	0.1	0.5	1
1 ($T = 1$)				×	
0.5 ($T = 2$)				×	
0.2 ($T = 5$)			×		
0.1 ($T = 10$)			×		
0.05 ($T = 20$)		×	×		
0.033 ($T = 30$)		×	×		
0.025 ($T = 40$)		×			

TABLE 1 – Selectivity of the network with different network/stimuli timescale. A cross \times means that the network is sensitive to the stimulus timescale for the network timescale range

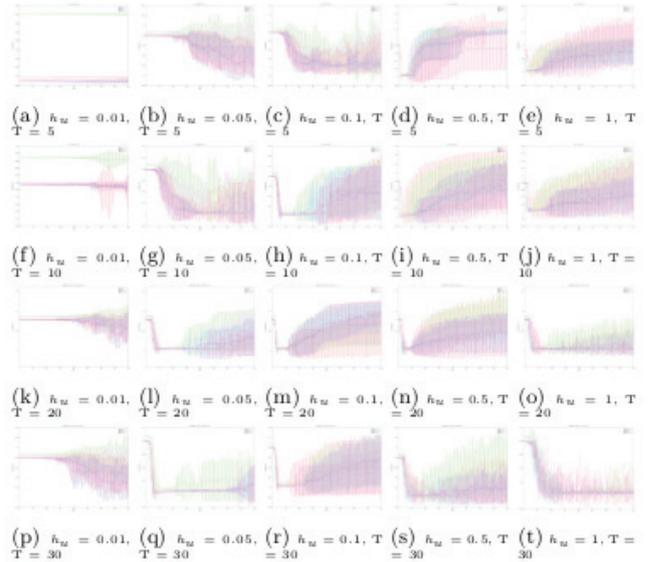


FIGURE 9 – Learning curves according to different network/stimuli timescales

The learning error curves are very different according to the networks¹⁶ and stimulus¹⁷ timescales. For instance, $T = 20$ is suitable for learning only with $h_u \in \{0.5, 0.1, 0.05\}$ and $T = 30$ for $h_u \in \{0.1, 0.05\}$. From a global point of view, stimuli sequences are properly learned in a **limited bandwidth** of the network update frequency. This is clearly a stimulus frequency selectivity property of the network. Such a network may be **tuned** to the correct update frequency in order to capture the required stimuli update frequency. Table 1 summarizes the networks selectivity property. Figures 9t and 9o relates a problem of ambiguous learning. It could be explained by the use of an inappropriate timescale. The network is updating too fast compared to the slow update rate of the stimulus. The time dependency no longer holds a sufficient amount of stimulus related informations. The dynamics of the reservoir is not input driven. Points on the attractors trajectory are no longer related to a stimulus. The error rate is increasing with time after reaching a minimum. This is due to the fact that at the very beginning of the learning process, when the

16. h_u
17. $\frac{1}{T}$

instantaneous error is diminishing, the membrane potential is not strong enough at this particular moment and is amplified by the sigmoid's gain¹⁸. It appears like a strong response with some gradients, matching the shape of the next incoming stimulus. Learning strengthens the synaptic connections involved in the coding of the prediction and the resulting activity becomes more and more saturated. At this point, the instantaneous error leaves its minimum and rises, until a remanence phenomenon occurs. This is explained in section 3.5.

3.4 Effect of h_w

Changing the value of h_w will have an obvious effect on the learning process since h_w is directly related to the learning equation. Rewriting equation (14) leads to :

$$(16) \quad J_{ij}^{(pq)}(t_{n+1}) = J_{ij}^{(pq)}(t_n) + (h_{ij}^{(pq)} \cdot \epsilon^{(pq)}) \cdot \frac{\Phi_g(p_i^{(p)}(t_n))}{N^{(q)}} \cdot d_i^{(p)}(t_n) \cdot d_j^{(q)}(t_{n-1}) + O(h_{ij}^{(pq)2})$$

By neglecting the $O(h_{ij}^{(pq)2})$ term, modifying $h_w = h_{ij}^{(pq)}$ is equivalent to choosing another learning rate parameter such that $\epsilon^{(pq)} = h_{ij}^{(pq)} \cdot \epsilon^{(pq)}$. The learning process will be slower but the local error will have an influence on the reservoir's dynamics¹⁹. Injecting a local truncation error of $O(h_{ij}^{(pq)2})$ at each learning cycle on each synaptic weight will induce a different activation value of the outgoing neurons since the activation function is non-linear and stiff. Small changes in the membrane potential could lead to noticeable changes in the activation value. Learning during T steps with a $\epsilon^{(pq)}$ learning rate is not strictly equivalent to learning during $\frac{T}{h_w}$ steps with a $\epsilon^{(pq)} = h_{ij}^{(pq)} \cdot \epsilon^{(pq)}$ learning rate. The figure 10

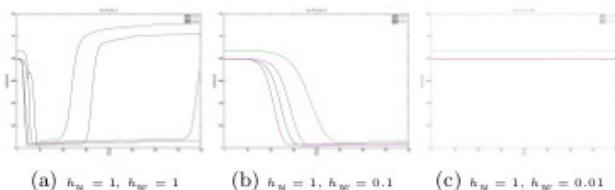


FIGURE 10 – Learning curves for different values of h_w

shows that the learning process is effectively slowed down. The timescale relative to the parameter $h_w = 1$ ²⁰ allows the network to reach a satisfactory learning process faster than a network set with a parameter $h_w = 0.1$ ²¹. The learning process is not only slowed down but the dynamics of the network are altered. The learning curves of each pattern do not reach their minimum at the same time in figure 10a and figure 10c. As stated in section 2.9, the evolution of equations (3) and (10) are strongly dependent and a fast variation of (10) will possibly trigger a fast variation in (3).

18. the g sigmoid slope parameter

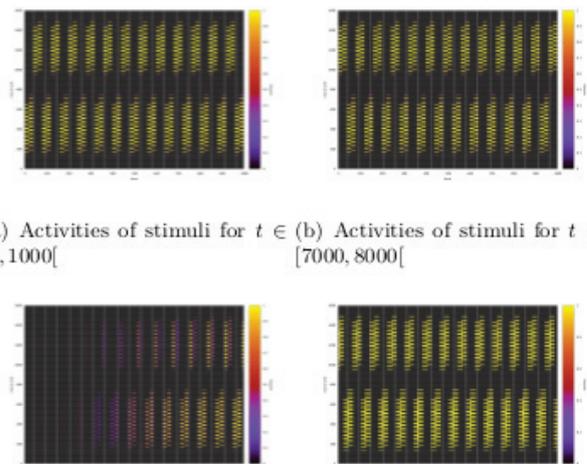
19. See section 2.7

20. Figure 10b

21. Figure 10b

3.5 Response latency reduction

The response latency of the network for giving a suitable prediction may vary according to the network's timescale²². A slow dynamic will induce a late response. This is not annoying and the network remains suitable for realtime purposes if the prediction is provided always before the next stimulus of the sequence. Figure 11 depicts the 'Points' sequence driving the network with a period $T = 20$ and a leak rate of $h_u = 0.05$. In this way, the network is tuned to the stimulus period²³. Figures 11a and 11b display the activities of the stimulus layer, driving the input layer. Each diagram is split in several tiles for readability purposes. Figures 11c and 11d represent the predictions of the output layer. The corresponding lear-



(a) Activities of stimuli for $t \in [0, 1000[$ (b) Activities of stimuli for $t \in [7000, 8000[$

(c) Predictions for $t \in [0, 1000[$ (d) Predictions for $t \in [7000, 8000[$

FIGURE 11 – Stimuli and predictions for $t \in [0, 1000[$ and $t \in [7000, 8000[$

ning curve²⁴ has been established by sampling the instantaneous error before the stimulus change. The error curve is rising after reaching its minimum. This is related to the error measure²⁵. According to the learning process, this prediction tends to minimize the error between the input layer and the next stimulus applied on this input layer. This input layer may add some distortion to the input stimulus because of the gain of the sigmoid function and the threshold of each neuron. The difference between a non saturated stimulus pattern and its distorted prediction will always be non zero. When the learning process is in its early stage, the readout weights applied to the secondary layer states gives a non saturated signal which is closer to the original/non saturated stimulus. Once the synaptic weights are reinforced, the incoming prediction signal is strong enough to trigger a saturated response of the output neurons. The comparison of the activities shows us that the response latency is diminishing as the learning duration is increasing. By looking closer at the diagram 11c, one

22. h_u , the leak rate parameter

23. $T * h_s = 1 \Leftrightarrow \frac{1}{T} = h_s$

24. Figure 12

25. Section 2.6

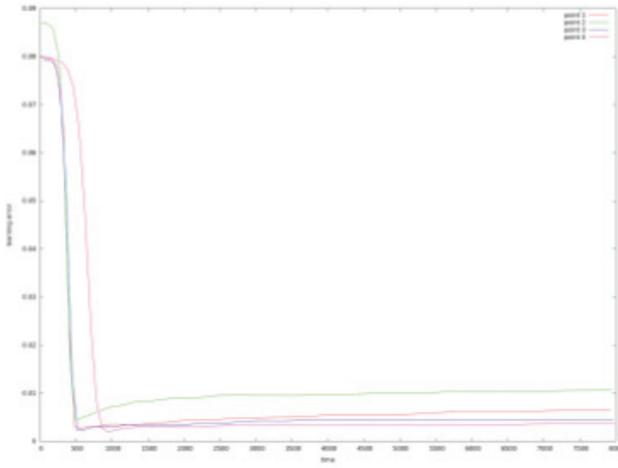


FIGURE 12 – Prediction error for $T = 20$ and $h_u = 0.05$

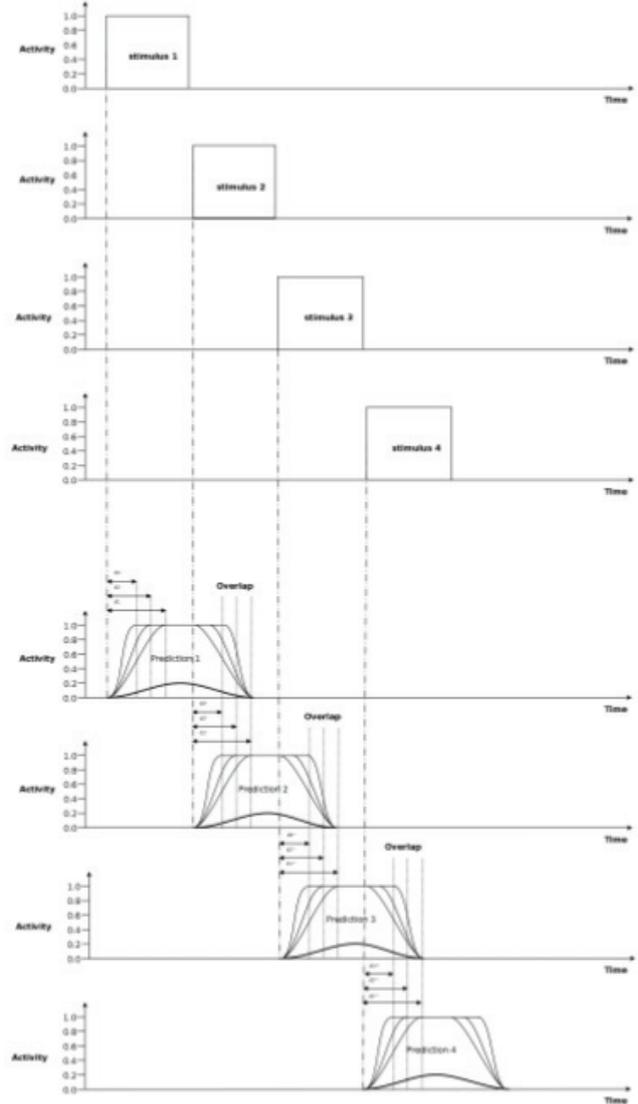


FIGURE 13 – Chronograms of the sequence prediction

Three snapshots of a prediction activation pattern for a given stimulus are reported on the same prediction graph. The network starts to predict the next stimulus with a weak prediction activation pattern²⁶. The amplitude is low and the slope is soft. Then, due to the learning effect, the predictions are strengthened and the activation pattern surface is wider. As learning occurs, the prediction pattern surfaces are spreading. The amplitude is rising and is of course saturated by the sigmoid function. The slope of each pattern is getting stiffer and the side effect is a reduction of the latency between the stimulus and its prediction²⁷. Even if the prediction response delay is decreasing, the prediction pattern remains a longer extent of time and may interfere with other prediction patterns. In fact, the network does not produce its predictions faster as the learning time is extended. It would be more ac-

may observe that the activation values are getting stronger as the learning time increases. Figure 13 explains from a global point of view the response latency reduction.

26. bold curve

27. the delays d_1 , d_2 , d_3

curate to say that the network computes strong predictions²⁸ faster as the learning time is extended. Figure 14 depicts an attempt to measure this phenomenon.



FIGURE 14 – Latency of prediction versus learning time

As shown in figure 11, the prediction are still constant, even if they overlap. The error function used in this study would measure an incorrect prediction since the overlapped prediction pattern is different from the expected pattern. An error measure including a novelty detector feature²⁹ will allow an agent implementing a *ReST* architecture to benefit from a reduced prediction latency over time.

4 Robot

The modified *ReST* architecture is able to learn stimuli sequences having a frequency³⁰ different from the network frequency³¹. Running experiments on a robot are more difficult. Sensors are producing noise and the robot may experience unforeseen phenomena from the environment. Stimulus will not be presented with exactly the same frequency. From the *ReST* network point of view, this jitter will enable the learning process on different points of the attractor trajectories.

28. saturated

29. like the learning rule (10)

30. $f_{stimulus} = \frac{1}{T}$

31. $h_u^{(p)} = \frac{1}{\tau_u^{(p)} \cdot N_u^{(p)}}$



FIGURE 15 – *Robulab 10*® used in this study

Figure 15 shows the *Robulab 10*® robot from ©*Robosoft* used in this study. A part of the software is embedded in the robot internal computer and the computational part is left to a workstation. The robot has two driving wheel mounted with rotary coders, one panoramic camera and an electronic compass. The wheels of the robot are submitted to frictional forces and the limited precision of rotary encoders and gears. This implies that a motor sequence may set the robot with an azimuth not previously perceived. The network state may be close to a known attractor and converge towards the expected state or far enough from a known situation and exhibit a chaotic / exploratory behavior [11].

4.0.1 Compass

As stated before, the robot may experience unforeseen phenomena from the environment. The electronic compass relies on magnetic fields and they may be perturbed by steel embedded in the building structure. Figure 16 depicts the result of such a phenomenon. The robot is rotating 3 times and two discontinuities are encountered from azimuth 40° to 70° and 240° to 255°. Even if the hypothesis of magnetic field deflection may be partially false, this measure is a hard proof about the difficulty to rely on coherent sensors data in this experiment.

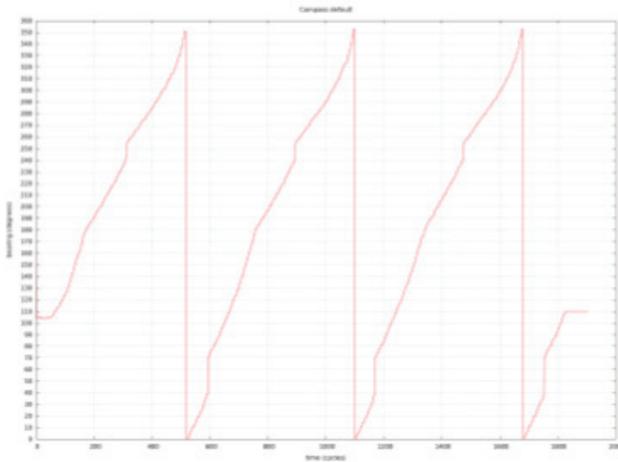


FIGURE 16 – Discontinuities of the compass measurements

4.0.2 Two sensory modalities network

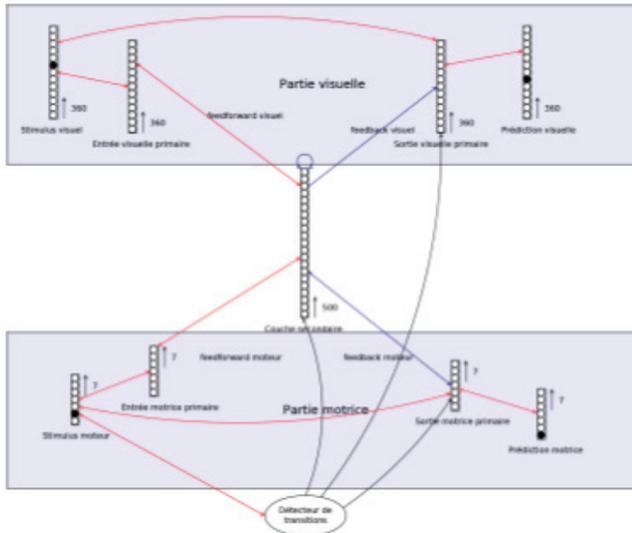


FIGURE 17 – *ReST* architecture with two sensorial modalities

The *ReST* architecture is modified for this experiment by extending it to a two sensory modalities network. Figure 17 shows this modification. Input and output layer are simply duplicated and adapted to the size of the new inputs. The learning rate parameters³² are set to a high value such that the learning process can occur in less than ten step. Of course, the network will be quickly pushed to overlearning and the restitution of spatio temporal pattern will be limited. This is a tradeoff for the experiment. Learning is triggered only when a motor event occurs. The robot is thus restricted to learn sensori-motor associations only when a new motor order is issued. This constraint has to be relaxed and no more transition detector should be used since novelty detection is built in the learning rule (10). Even if the presence of a such motor

32. $\epsilon^{(pq)}$ on the output layers only

event transition detector is artificial, this setup is sufficient for starting to study continuous time updates RRNN.

Motor In this study, the robot's moves are restricted to simple rotations. No translation occurs. Figure 15 depicts the workspace which is partitionned in absolute bearings identified with rubber tape stripes : 0° , 30° , 90° , 180° , 210° , 270° and 360° . From the robot point of view, those bearings are reachable by 3 successive rotations of : 30° , 60° , 90° if we consider a positive spinning angle. The motor part is a simple layer of 7 neurons encoding the 7 possible relative azimuths with a population coding : -90° , -60° , -30° , 0° , 30° , 60° , 90° . The current move corresponding activity is sustained until the rotation is successfully achieved³³. Such motor orders are forced via a joystick by the experimenter.

Visual As stated before, the compass measure is noisy and not suitable for building a more complex perception based on visual place cells [20]. Place cells are based on unsupervised classification and these landmarks are built according to the panoramic camera and the electronic compass. The absolute azimuth of a landmark is computed thanks to the absolute azimuth of the robot, the relative azimuth of the panoramic camera and camera specific geometrical clues. The resulting visual place cells are not properly categorized at the neighborhood of the discontinuities shown in 16. For simplicity and feasibility and since the robots moves are rotations only, visual place cells are implemented with a simple population coding of the compass angle to the magnetic north. A layer of 360 neurons is relating the current bearing of the robot with a 1° error.

4.1 Simple sequence

The experience from [11] is reproduced. The robot is positioned inside a circular workspace and the front of the robot is pointing to the beginning of the workspace³⁴. The motor sequenced learned by the robot is : 30° , 60° , 90° . At the end of one motor sequence, the robot will points toward the 180° azimuth and after the second motor sequence, the perceived visual pattern will be the same as at the beginning of the experiment. The period of the motor stimuli is 2 while the period of the visual stimuli is 1. When the predictions reach a satisfying level³⁵, the joystick is no longer used and the robot is driven only by its predictions. The sequence is replayed without any exterior intervention.

33. the robots angular speed command goes below a very low threshold

34. bearing 0°

35. saturated and sustained activation, nearly immediate response

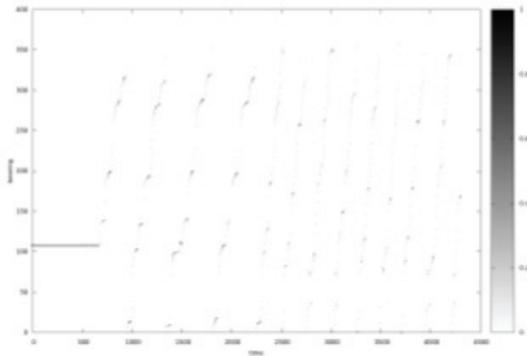
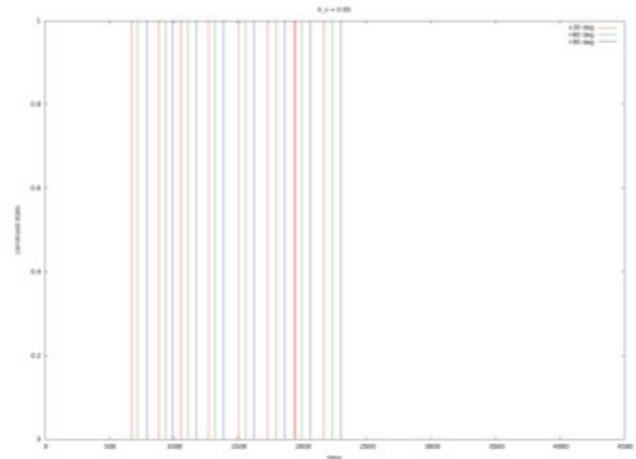


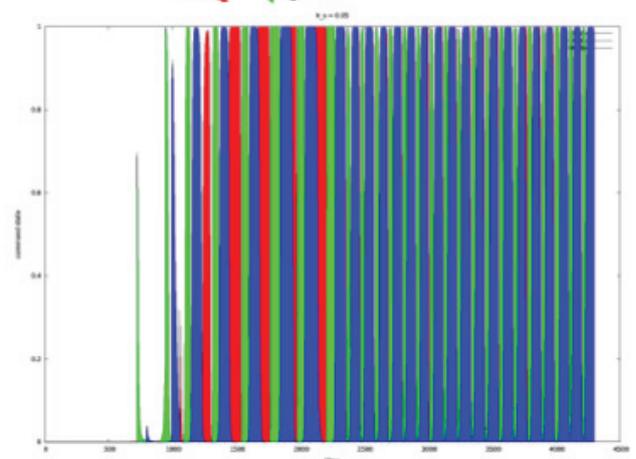
FIGURE 18 – Visual patterns vs time

Figure 18 shows visual patterns³⁶ through time. Each point represents the angle between the robot and the magnetic north. As expected, wide bearing moves are slower than the tiny ones. Friction forces and population coding of the azimuth explain the variation of the visual pattern. Thus, it is shifted around the 0° bearing. The starting angle after each double sequence is not exactly the same. Compass error observed in 16 are contributing too to these irregularities.

36. bearing of the robot in our case



(a) Motor orders sent to the robot ($h_s = 0.05$)



(b) Prediction of the motor orders ($h_s = 0.05$)

FIGURE 19 – Motor orders and their predictions

Figure 19a shows the motor order sent to the robot by the experimenter. Even if the imposed rhythm of motor stimuli tends to be as steady as possible, phase fluctuations are observed. Because of friction forces and bearing algorithm slightly fluctuating convergence time, the moment where the robot is stabilised might be slightly different at each different sequence. At time $t \approx 2300$, the motor sequence is considered to be learned and motor forcing is disabled. The robot is only driven by the prediction of its next moves and anticipated visual situation. Figure 22i shows that the motor sequence is replayed with the good serial structure³⁷ but with a different temporal structure³⁸. In fact, the robot is no longer constrained to wait the end of its current move. As soon as the visuomotor perception triggers the evocation of the next action and visual situation, the corresponding azimuth is sent as the target steering of the robot. Moreover, the motor input is sustained while the robot is moving. It means that the proprioception of the robot is forced to be the same as if it were at the beginning

37. $30^\circ \rightarrow 60^\circ \rightarrow 90^\circ$

38. shorter delays between motor orders

of the move. The proprioception is changing at every moment of the move and should be taken into account. In fact, the original experiment described in [11] contained a stronger working hypothesis : One move of the robot was spanned across one time step and during this step, the network was frozen. Stimulus, robot, network update and learning used to evolve within the same timescale. In this continuous time study each stimulus has its own frequency that might fluctuate because of the experimenter. The robot sensors and actuators are running at their own frequencies³⁹ and are constrained by their design. The network update frequency is influenced by a controlled h_u leak rate and the computing power of the supporting architecture and its cpu load. The discretization of motor orders into seven relative azimuth is inducing a time dependency against motion completion. The proprioception is frozen while the robot and the network are reacting to the environment. The motor neurons layers should be changed to a more continuous and responsive implementation in order to observe the correct temporal constraints reproduction during replay. A population coded layer representing the possible angular speeds would be more appropriate. Practically speaking, the current setup is sufficient for demonstrating the networks ability of serial replay of sensori motor associations with a continuous time update scheme of a RRNN.

4.2 Restitution without compass

The same experiment as in section 4.1 is realized but this time, the visual input⁴⁰ is cut just before replaying the sequence. This is the equivalent to occulting the panoramic camera with an opaque mask. Please note that in the original experiment, the camera is still plugged and thermal noise is in fact driving the associative network. Since the noise is not correlated to motor orders, no sensory motor pattern should be learned by the time delay reinforcement rule (10). Learning is done as previously and at $t = 1500$ the learning process is assessed to be satisfactory. The compass information is forced to zero and the replay of the visuo-motor sequence is triggered (Figure 20).

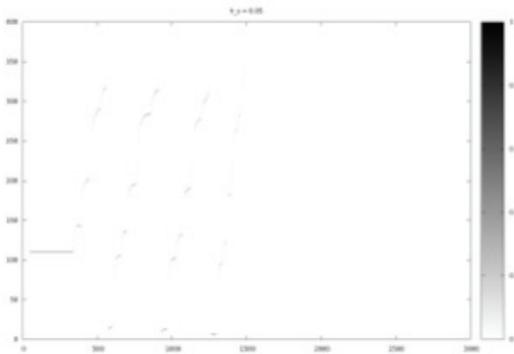


FIGURE 20 – Visual patterns during learning and replay

39. refresh rate

40. the population coded bearing angle given by the compass

Figure 21b shows that the replay without the compass is not the same.

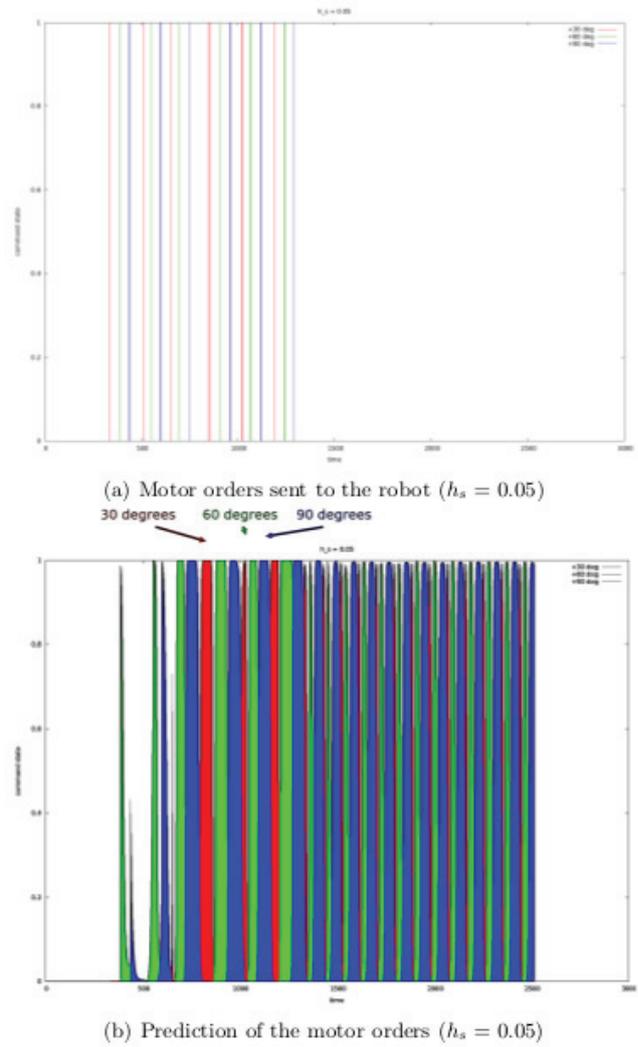


FIGURE 21 – Motor orders and their predictions

This time, the robot is no more driven by its visual input during replay. The next motor order prediction is computed only thanks to the current motor order. The robot is blind and the replay continues for a long time. A shorter replay time was expected. As stated before, $\epsilon^{(pq)}$ learning rate constants were arbitrarily set to unusual high values in order to achieve an experiment with a decent amount of trials. Learning is triggered only by motor events and visual transitions are no longer used as clues in the replay. The learning process captured only the serial structure of motor orders. A population coded motor layer representing the instantaneous possible angular velocities should solve the problem of having one sensorial modality and one target azimuth angle instead of two responsive sensorial modalities.

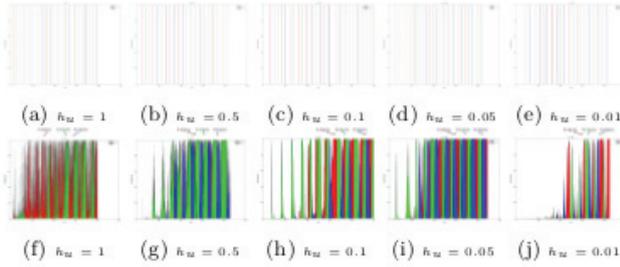


FIGURE 22 – Motor command signals and their predictions

4.3 Several time scales

The same sequence is learned for different values of h_u parameter. Figure 22f shows that the timescale $h_u = 1$ is not adapted to the timescale of the stimuli. The network is unable to capture the stimuli sequence because the history of its states is fading too fast compared to the slow stimuli frequency. Figure 22g shows more structured predictions. Predictions are contiguous and the serial order is correct. However, the 30° motor order prediction⁴¹ is missing too often to be considered as a good prediction. Timescales used in both figures 22h and 22i are adapted to the stimuli timescale. Motor orders are all predicted in the good serial order with a high intensity and sustained response. The timescale used in figure 22h is no longer adapted. The network takes too much time to accumulate activities and converge towards correct predictions and in this case, the experimenter tends to adapt its motor order frequency to the network prediction frequency. This is the same selectivity property observed in section 3.3.

5 Conclusion

We saw that it is possible both in theory (chapter ??) and in practice (chapitre ??) to use continuous time update RRNN in order to learn and replay sequences. On the contrary to [11], the predictions are no longer directly injected in the network. The latter have its own dynamics and timescales and is no longer constrained to be phase locked with the incoming stimuli (section ??). Stimuli occur at their own timescale and the h_s parameter is used in order to "tune" the time constant of the network to the timescale of the stimuli (section ??). This is the key in the successful implementation of a visuomotor sequences learning task on a mobile robot (chapitre ??).

The parameter β needs to be studied more in details and in particular, its relationship with the parameter h_s (section 3.1.1). This is required in order to get rid of the additional novelty detection mechanism based on the first order derivative. The continuous time update RRNN robustness against noise needs to be studied. Another interesting direction would be to study the network dynamical properties with a different scheduling. Indeed, each group of neurons are updated in a fixed order. Relaxing this constraint will have a major influence on the convergence of the network towards solutions

41. red color

Nous avons vu que les stimuli non orthogonaux engendraient des prédictions incomplètes qui ressemblaient à une combinaison des motifs non orthogonaux entre eux (section ??). C'est peut-être là un phénomène intéressant à expliquer et à exploiter pour prédire la prochaine zone d'intérêt en mouvement d'un stimulus dans sa projection sur la couche primaire.

Une des problèmes inhérents à l'apprentissage par renforcement est la question du moment à partir duquel le réseau ne doit plus apprendre. Un premier élément de réponse est obtenu en examinant la règle d'apprentissage (équation (10)) qui comporte une fonction permettant d'empêcher l'apprentissage d'une prédiction sur un champ local contenant suffisamment d'information. Un deuxième élément de réponse vient en exploitant une observation simple : l'erreur d'apprentissage qui est mesurée à chaque instant possède la caractéristique d'augmenter lentement une fois qu'elle a atteint son minimum, pour ensuite se stabiliser lorsque les champs locaux du lien de feedback sont saturés. On peut aussi faire le choix de continuer l'apprentissage jusqu'à l'obtention d'un temps de réponse satisfaisant au prix d'une rémanence de la prédiction précédente (section 3.5). Dans ce cas là, il est nécessaire d'ajouter un mécanisme permettant d'extraire la dernière prédiction avant de mesurer son erreur et de la réinjecter. Des mécanismes attentionnels peuvent bien sûr moduler la plasticité du réseau.

Les réseaux *ReST* ont des propriétés similaires aux Echo State Network et peuvent être utilisés en ligne avec beaucoup moins de contraintes. Les expériences menées dans cette étude contenaient uniquement des stimuli ne proposant qu'une seule fréquence. Les réseaux étudiés n'utilisent qu'une seule couche secondaire et moyennant un réglage adapté du paramètre h_s , il est possible de "sensibiliser" le réseau à la fréquence de présentation des stimuli. Pour aller plus loin et être capable de gérer des stimuli reposant sur plusieurs fréquences "internes", différentes couches secondaires accordées avec différents paramètres h_s pourraient être considérées comme des bases permettant la décomposition temporelle de stimuli. Un air de musique capté par une oreille par exemple présente plusieurs variations de la pression de l'air à plusieurs échelles de temps différentes. L'échelle de temps rapide concerne la hauteur des notes. L'échelle de temps plus globale et donc plus lente concerne la durée des notes et on peut aussi considérer une troisième échelle de temps. La même analyse peut être faite en prenant l'exemple de la parole qui peut se décomposer en mots qui eux mêmes peuvent être décomposés en syllabes élémentaires. Ces syllabes peuvent être décomposées en formants. La phrase prononcée est alors une combinaison de ces éléments sur plusieurs échelles de temps. Le mouvement lui aussi peut être décomposé en sous mouvements élémentaires. D'autre part, chaque neurone de la couche secondaire et sa pondération peut être vu comme une projection du stimulus dans un espace à une dimension définie aléatoirement selon la loi de distribution gaussienne vue dans ???. Tout se passe comme si chaque neurone de la couche secondaire extrayait une caractéristique spatiale du stimulus. Ces caractéristiques spatiales sont combinées entre elles d'une itération à l'autre par le biais du lien récurrent. La sortie de la couche secon-

daire peut donc être vue à chaque instant comme un encodage spatio-temporel selon une échelle de temps donnée des stimuli présentés jusqu'à lors. Il devrait donc être possible de décomposer selon le temps des stimuli génériques en considérant un banc de couches secondaires paramétrées judicieusement avec des h_s différents. Rien n'interdit la communication entre ces couches secondaires et elles peuvent s'influer entre elles. Ainsi, peut-être est-il possible d'observer un mécanisme tel que la couche à l'échelle de temps la plus lente influe sur la dynamique de l'échelle de temps plus rapide de manière à lui faire générer des états particuliers à haute fréquence, propres à cette même "basse fréquence" qui sert finalement de support. Jaeger ([19]) a déjà eu un raisonnement similaire avec les ESN. Selon lui, les caractéristiques à basse fréquence temporelle guidaient l'extraction de caractéristiques à plus haute fréquence. Le signal ainsi décomposé est reconstitué pour former la prédiction correspondant à la prochaine entrée.

En définitive, cette étude a permis de mettre en évidence l'apport fondamental de l'utilisation d'équations de mise à jour à temps continu : On dispose maintenant de RNRA capable d'apprendre et de restituer des séquences de stimuli à plusieurs échelles de temps. Ces RNRA ont la particularité d'avoir leurs dynamiques propres et d'être accordés à différentes "fréquences". Il est très tentant d'aller vers l'analyse/synthèse multi échelle. La figure 23 décrit ce que pourrait être une architecture *ReST* utilisant plusieurs échelles de temps.

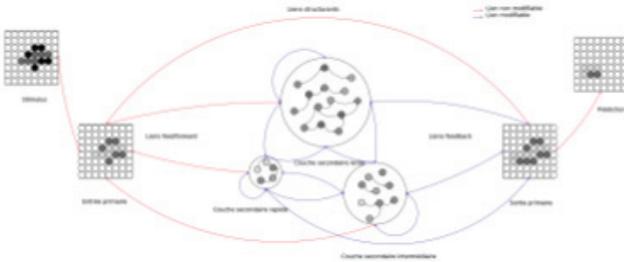


FIGURE 23 – Schéma préliminaire de l'architecture *ReST* multi-échelles

L'intuition est la suivante : le stimulus appliqué à la couche primaire alimenterait les couches secondaires paramétrées à des échelles de temps h_s différentes pour capturer une gamme plus large de dynamiques des stimuli. Pour chaque neurone de la couche d'entrée, plusieurs chemins existent jusqu'aux neurones de la couche de sortie. Certains empruntent toutes les couches secondaires et d'autres qu'une seule. D'autres encore utilisent plusieurs fois la même couche avant de passer par une autre couche secondaire. La combinatoire est très grande et tous ces chemins sont calculés en parallèles. La distribution des poids en $\mathcal{N}(0, \frac{1}{\sqrt{N(p)}})$ fait en sorte que les couches secondaires comportent beaucoup de sous réseaux indépendants. En calculant l'activité du réseau, on calcule en fait tous les chemins possibles. La plupart de ces chemins auront une longueur différente dans le temps et c'est ce qui va nous permettre de décomposer les stimuli selon les plusieurs bases de temps que sont les couches secondaires. Seuls les chemins ayant une activité corrélée avec les stimuli seront utilisés dans

le décodage des activités des couches secondaires par les liens de feedback et seront renforcés par apprentissage.

Références

- [1] D.J. Amit. *Modeling Brain Function : The World of Attractor Neural Networks*. 1994.
- [2] Jean P. Banquet, Philippe Gaussier, Jean Claude Dreher, Cédric Joulain, Arnaud Revel, Wilfried Günther, and Neuroscience Et Modélisation. Space-time, order, and hierarchy in fronto-hippocampal system : A neural basis of personality. In *In Matthews, G., (Ed.), Cognitive Science Perspectives on Personality and Emotion. Elsevier Science BV*, pages 123–189. Elsevier Science BV, 1997.
- [3] S. T. S. Becker and K. Obermayer, editors. *Adaptive nonlinear system identification with echo state networks*. MIT Press Cambridge, MA, 2003.
- [4] Hugues Berry and Mathias Quoy. Structure and dynamics of random recurrent neural networks. *Adaptive Behavior*, (14), 2006.
- [5] L.C. Bray-Jayet, M. Quoy, P.H. Goodman, and F.C. Harris. A circuit-level model of hippocampal place field dynamics modulated by entorhinal grid and suppression-generating cells. *Frontiers in Neural Circuits*, 4(0) :12, 2010.
- [6] Nicolas Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journ. Comp. Neurosci.*, (8) :183–208, 2000.
- [7] B. Cessac, B. Doyon, M. Quoy, and M. Samuelides. Mean-field equations, bifurcation map and route to chaos in discrete time neural networks. *Physica D*, 74 :24–44, 1994.
- [8] Bruno Cessac, Bernard Doyon, Mathias Quoy, and Manuel Samuelides. Mean-field equations, bifurcation map and route to chaos in discrete time neural networks. *Physica D*, (74) :24–44, 1994.
- [9] Emmanuel Daucé. *Adaptation dynamique et apprentissage dans les réseaux de neurones récurrents aléatoires*. PhD thesis, SUPAERO, 1 2000.
- [10] Emmanuel Daucé, Mathias Quoy, Bruno Cessac, Bernard Doyon, and Manuel Samuelides. Self-organization and dynamics reduction in recurrent networks : stimulus presentation and learning. *Neural Networks*, (11) :521–533, 1998.
- [11] Emmanuel Daucé, Mathias Quoy, and Bernard Doyon. Resonant spatiotemporal learning in large random recurrent network. *Biological Cybernetics*, (87) :185–198, 2002.
- [12] Peter Ford Dominey. Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning. *Biological cybernetics*, (73) :265–274, 1995.
- [13] Peter Ford Dominey. A shared system for learning serial and temporal structure of sensori-motor sequences ? Evidence from simulation and human experiments. *Cognitive Brain Research*, 6(3) :163–172, 1998.

- [14] William E.Boyce and Richard C. DiPrima. *Equations différentielles*, chapter Méthodes numériques. 2002.
- [15] Jaeger H. The "echo state" approach to analysing and training recurrent neural networks - with an erratum note, 2010.
- [16] Jaeger H. A tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach, 2013.
- [17] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79(8) :2554–2558, 1982.
- [18] H. Jaeger and H. Haas. Harnessing nonlinearity : Predicting chaotic systems and saving energy in wireless communication. *Science*, (304) :78–80, 2004.
- [19] Herbert Jaeger. Discovering multiscale dynamical features with hierarchical echo state networks. 2007.
- [20] Adrien Jauffret, Nicolas Cuperlier, Philippe Gaussier, and Philippe Tarroux. Multimodal integration of visual place cells and grid cells for navigation tasks of a real robot. In *12th International Conference on Simulation of Adaptive Behavior, SAB 2012*, number 7426 in LNAI, pages 136–145, Odense, Denmark, August 2012. Springer.
- [21] L. Jayet, M. Quoy, and P. Goodman. Spike timing computational model of hippocampal-frontal dynamics underlying navigation and short-term memory. In *Computational and Systems Neuroscience*, Salt Lake City, Etats-Unis, 2010.
- [22] Daniel J. Jobson and Zia-ur Rahman. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 6(7), 1997.
- [23] Donald E. Knuth. *The Art of Computer Programming, Volume II : Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981.
- [24] Matthieu Lagarde. *Apprentissage de nouveaux comportements : vers le développement épigénétique d'un robot autonome*. PhD thesis, Université de Cergy-Pontoise, 7 2010.
- [25] Rodrigo Laje and Dean V. Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neurosci.*, (16(7)) :925–936, 2013.
- [26] Lavoisier, editor. *Approche dynamique de la cognition artificielle*, chapter Mémoire dynamique et planification. Hermes science, 2002.
- [27] Lavoisier, editor. *Approche dynamique de la cognition artificielle*, chapter La mémoire comme anticipation de la perception. Hermes science, 2002.
- [28] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states : A new framework for neural computation based on perturbations. *Neural Computation*, 14(11) :2531–2560, 2002.
- [29] Natschläger T. Maass W. Real-time computing without stable states : A new framework for neural computation based on perturbations. *Neural Computation*, 2002.
- [30] J Namikawa, R. Nishimoto, and J. Tani. A neurodynamic account of spontaneous behaviour. *PLOS computational biology*, 7, 2011.
- [31] R. Nishimoto and J. Tani. Learning to generate combinatorial action sequences utilizing the initial sensitivity of deterministic dynamical systems. In *Lecture Notes in Computer Science, IWANN2003*, volume 2686, pages 422–429, Menorca, Spain, 2003.
- [32] Ryu Nishimoto and Jun Tani. Learning to generate combinatorial action sequences utilizing the initial sensitivity of deterministic dynamical systems. *Neural Networks*, 17(7) :925–933, 2004.
- [33] Rainer W. Paine and Jun Tani. Motor primitive and sequence self-organization in a hierarchical recurrent neural network. In *Neural Networks*, volume 17, pages 1291–1309, 2004.
- [34] Mathias Quoy. *Apprentissage dans les réseaux neuro-mimétiques à dynamique chaotique*. PhD thesis, SUPAERO, 12 1994.
- [35] D. Ruelle and J.P. Eckman. Ergodic theory of chaos and strange attractors. *Review of Modern Physics*, (57) :617–656, 1985.
- [36] A. Samsonovich and B.L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network mode. *Journal of Neuroscience*, 17(15) :5900–5920, 1997.
- [37] C.A. Skarda and W.J. Freeman. How brains make chaos in order to make sense of the world. *Behavioral and brain science*, 10 :123–189, 1987.
- [38] H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in random neural networks. *Phys. Rev. Lett.*, 61 :259–262, Jul 1988.
- [39] H. Sompolinsky, A. Crisanti, and H.J. Sommers. Chaos in random neural networks. *Phys. Rev. Lett.*, (61) :259–262, 1988.
- [40] S.H. Strogatz. *Nonlinear Dynamics and Chaos*. 1994.
- [41] David Susillo and L.F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, (63) :544–557, 2009.
- [42] J. Tani. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. on Syst. Man Cybern. Part B-Cybernetics*, 26(3) :421–436, 1996.
- [43] J. Tani and S. Nolfi. Learning to perceive the world as articulated : An approach for hierarchical learning in sensory-motor systems. *Neural Networks*, 12(7-8) :1131–1141, 1999.
- [44] Jun Tani and Masato Ito. Self-organization of behavioral primitives as multiple attractor dynamics : A robot experiment. *IEEE Transactions on Systems, Man, and Cybernetics Part A :Systems and Humans.*, 33(4) :481–488, 2003.
- [45] I. Tsuda. Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind. *World Futures*, (32) :167–185, 1991.
- [46] I. Tsuda and T. Umemura. Chaotic itinerancy generated by coupling of milnor attractors. *Chaos*, (13) :926–936, 2003.

- [47] T.P. Vogels and L.F. Abbott. Signal propagation and logic gating in networks of integrate-and-fire neurons. *Journ. Neurosci.*, (25(46)) :10786–10795, 2005.
- [48] T.P. Vogels, K. Rajan, and L.F. Abbott. Neural network dynamics. *Annu. Rev. Neurosci.*, (28) :357–376, 2005.
- [49] Maass W. *Computability in Context : Computation and Logic in the Real World*, chapter Motivation, theory, and applications of liquid state machines. Imperial College Press, 2010.
- [50] Y. Yamashita and J. Tani. Emergence of functional hierarchy in a multiple timescale neural network model : A humanoid robot experiment. *PLOS computational biology*, 4, 2008.
- [51] Y. Yamashita and J. Tani. Emergence of functional hierarchy in a multiple timescale neural network model : a humanoid robot experiment. *PLoS computational biology*, 4(11) :e1000220, 2008.

Bibliography

- Adams, Rick a, Klaas Enno Stephan, Harriet R Brown, Christopher D Frith, and Karl J Friston. 2013. "The Computational Anatomy of Psychosis." *Frontiers in Psychiatry* 4 (May): 47. <https://doi.org/10.3389/fpsy.2013.00047>.
- Alvernhe, Alice, Francesca Sargolini, and Bruno Poucet. 2012. "Rats Build and Update Topological Representations through Exploration." *Animal Cognition* 15 (3): 359–68. <https://doi.org/10.1007/s10071-011-0460-z>.
- Ambrose, R. Ellen, Brad E. Pfeiffer, and David J. Foster. 2016. "Reverse Replay of Hippocampal Place Cells Is Uniquely Modulated by Changing Reward." *Neuron* 91 (5): 1124–36. <https://doi.org/10.1016/J.NEURON.2016.07.047>.
- Arleo, Angelo, and Wulfram Gerstner. 2000. "Spatial Cognition and Neuro-Mimetic Navigation: A Model of Hippocampal Place Cell Activity." *Biological Cybernetics* 83 (3): 287–99. <https://doi.org/10.1007/s004220000171>.
- Arleo, Angelo, Fabrizio Smeraldi, and Wulfram Gerstner. 2004. "Cognitive Navigation Based on Nonuniform Gabor Space Sampling, Unsupervised Growing Networks, and Reinforcement Learning." *IEEE Transactions on Neural Networks* 15 (3): 639–52. <https://doi.org/10.1109/TNN.2004.826221>.
- Arulkumaran, Kai, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. "Deep Reinforcement Learning: A Brief Survey." *IEEE Signal Processing Magazine*. <https://doi.org/10.1109/MSP.2017.2743240>.
- Ball, David, Scott Heath, Michael Milford, Gordon Wyeth, and Janet Wiles. 2010. "A Navigating Rat Animat." *Proceedings of the 12th International Conference on the Synthesis and Simulation of Living Systems* 40 (3): 271–83. <https://doi.org/10.1088/0031-9120/36/6/301>.
- Barrera, Alejandra, Gonzalo Tejera, Martin Llofriu, and Alfredo Weitzenfeld. 2015. "Learning Spatial Localization: From Rat Studies to Computational Models of the Hippocampus." *Spatial Cognition & Computation* 15 (1): 27–59. <https://doi.org/10.1080/13875868.2014.961602>.
- Barrera, Alejandra, and Alfredo Weitzenfeld. 2008. "Biologically-Inspired Robot Spatial Cognition Based on Rat Neurophysiological Studies." *Autonomous Robots* 25 (1–2): 147–69. <https://doi.org/10.1007/s10514-007-9074-3>.
- Bartlett, F C. 1932. "A Study in Experimental and Social Psychology." England: Cambridge University Press.
- Baum, Leonard E, and Ted Petrie. 1966. "Statistical Inference for Probabilistic Functions of Finite State Markov Chains." *The Annals of Mathematical Statistics* 37 (6): 1554–63.
- Bendor, Daniel, and Matthew A Wilson. 2012. "Biasing the Content of Hippocampal Replay during Sleep." *Nature Neuroscience* 15 (10): 1439–44. <https://doi.org/10.1038/nn.3203>.
- Burgess, Neil, Michael Recce, and John O'Keefe. 1994. "A Model of Hippocampal Function." *Neural Networks* 7 (6–7): 1065–81. [https://doi.org/10.1016/S0893-6080\(05\)80159-5](https://doi.org/10.1016/S0893-6080(05)80159-5).
- Caluwaerts, K., M. Staffa, S. Nguyen, C. Grand, L. Dollé, A. Favre-Félix, B. Girard, and M. Khamassi. 2012. "A Biologically Inspired Meta-Control Navigation System for the Psikharpax Rat Robot." *Bioinspiration and Biomimetics* 7 (2). <https://doi.org/10.1088/1748-3182/7/2/025009>.
- Carr, Margaret F, Shantanu P Jadhav, and Loren M Frank. 2011. "Hippocampal Replay in the Awake State: A Potential Substrate for Memory Consolidation and Retrieval." *Nature Neuroscience*. <https://doi.org/10.1038/nn.2732>.
- Cenquizca, Lee A., and Larry W. Swanson. 2007. "Spatial Organization of Direct Hippocampal Field CA1 Axonal Projections to the Rest of the Cerebral Cortex." *Brain Research Reviews*. <https://doi.org/10.1016/j.brainresrev.2007.05.002>.
- Chen, Ding Jun, Chung Yeol Lee, Cheol Hoon Park, and Pedro Mendes. 2007. "Parallelizing Simulated Annealing Algorithms Based on High-Performance Computer." *Journal of Global Optimization* 39 (2): 261–89. <https://doi.org/10.1007/s10898-007-9138-0>.
- Chumbley, Justin R, Raymond J Dolan, and Karl J Friston. 2008. "Attractor Models of Working Memory and Their Modulation by Reward." *Biological Cybernetics* 98 (1): 11–18. <https://doi.org/10.1007/s00422-007-0202-0>.
- Cutsuridis, Vassilis, and Michael Hasselmo. 2011. "Spatial Memory Sequence Encoding and Replay During Modeled Theta and Ripple Oscillations." *Cognitive Computation* 3 (4): 554–74.

- <https://doi.org/10.1007/s12559-011-9114-3>.
- Cutsuridis, Vassilis, and Jiannis Taxidis. 2013. "Deciphering the Role of CA1 Inhibitory Circuits in Sharp Wave-Ripple Complexes." *Frontiers in Systems Neuroscience* 7 (May): 13. <https://doi.org/10.3389/fnsys.2013.00013>.
- Davidson, Thomas J, Fabian Kloosterman, and Matthew A Wilson. 2009. "Hippocampal Replay of Extended Experience." *Neuron* 63 (4): 497–507. <https://doi.org/10.1016/j.neuron.2009.07.027>.
- Delatour, B, and M P Witter. 2002. "Projections from the Parahippocampal Region to the Prefrontal Cortex in the Rat: Evidence of Multiple Pathways." *The European Journal of Neuroscience* 15 (8): 1400–1407. <https://doi.org/1973> [pii].
- Diba, Kamran, and György Buzsáki. 2007a. "Forward and Reverse Hippocampal Place Cell Sequences during Ripples SUPPL." *Nature Neuroscience* 10: 1241–42. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2039924/pdf/nihms-31548.pdf>.
- . 2007b. "Forward and Reverse Hippocampal Place-Cell Sequences during Ripples." *Nature Neuroscience* 10 (10): 1241–42. <https://doi.org/10.1038/nn1961>.
- Dollé, Laurent, Denis Sheynikhovich, Benoît Girard, Ricardo Chavarriaga, and Agnès Guillot. 2010. "Path Planning versus Cue Responding: A Bio-Inspired Model of Switching between Navigation Strategies." *Biological Cybernetics* 103 (4): 299–317. <https://doi.org/10.1007/s00422-010-0400-z>.
- Dominey, Peter Ford. 1995. "Complex Sensory-Motor Sequence Learning Based on Recurrent State Representation and Reinforcement Learning." *Biological Cybernetics* 73 (73): 265–74. <http://link.springer.com/article/10.1007/BF00201428>.
- . 1998a. "A Shared System for Learning Serial and Temporal Structure of Sensori-Motor Sequences? Evidence from Simulation and Human Experiments." *Cognitive Brain Research* 6 (3): 163–72. <http://www.sciencedirect.com/science/article/pii/S0926641097000293>.
- . 1998b. "Influences of Temporal Organization on Sequence Learning and Transfer: Comments on Stadler (1995) and Curran and Keele (1993)." *Journal of Experimental Psychology* 24 (1): 234–48.
- Dominey, Peter Ford, Michael Arbib, and Jean-Paul Joseph. 1995. "A Model of Corticostriatal Plasticity for Learning Oculomotor Associations and Sequences." *Journal of Cognitive Neuroscience* 7 (3): 311–36. <http://www.mitpressjournals.org/doi/abs/10.1162/jocn.1995.7.3.311>.
- Dominey, Peter Ford, Toshio Inui, and Michel Hoen. 2009. "Neural Network Processing of Natural Language: II. Towards a Unified Model of Corticostriatal Function in Learning Sentence Comprehension and Non-Linguistic Sequencing." *Brain and Language* 109 (2–3): 80–92. <https://doi.org/10.1016/j.bandl.2008.08.002>.
- Dominey, Peter Ford, and Franck Ramus. 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 (1): 87–127. <http://www.tandfonline.com/doi/abs/10.1080/016909600386129>.
- Dorigo, M, and L M Gambardella. 1997. "Ant Colonies for the Travelling Salesman Problem." *Bio Systems* 43 (2): 73–81. [https://doi.org/10.1016/S0303-2647\(97\)01708-5](https://doi.org/10.1016/S0303-2647(97)01708-5).
- Dugad, Rakesh, and Ub Desai. 1996. "A Tutorial on Hidden Markov Models." ... *of Electrical Engineering Indian Institute of ...*, no. February: 1–16. <https://doi.org/10.1109/5.18626>.
- Edwards, Mark J, Rick a Adams, Harriet Brown, Isabel Pareés, and Karl J Friston. 2012. "A Bayesian Account of 'Hysteria'." *Brain: A Journal of Neurology* 135 (Pt 11): 3495–3512. <https://doi.org/10.1093/brain/aws129>.
- Eichenbaum, Howard. 2014. "Time Cells in the Hippocampus: A New Dimension for Mapping Memories." *Nature Reviews Neuroscience*. <https://doi.org/10.1038/nrn3827>.
- Enel, Pierre, Emmanuel Procyk, René Quilodran, and Peter Ford Dominey. n.d. "Dynamical Mixed Selectivity in Reservoir Computing and Primate Prefrontal Cortex."
- Euston, David R, Masami Tatsuno, and Bruce L McNaughton. 2007. "Fast-Forward Playback of Recent Memory Sequences in Prefrontal Cortex during Sleep." *Science (New York, N.Y.)* 318 (5853): 1147–50. <https://doi.org/10.1126/science.1148979>.
- Filliat, David, and Jean-Arcady Meyer. 2002. "Global Localization and Topological Map-Learning for Robot Navigation." *Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, 131–40. <http://portal.acm.org/citation.cfm?id=773380.773400>.
- Fine, Shai, Yoram Singer, and Naftali Tishby. 1998. "Hierarchical Hidden Markov Model: Analysis and Applications." *Machine Learning* 32 (1): 41–62. <https://doi.org/10.1023/A:1007469218079>.

- Foster, David J., and Matthew A. Wilson. 2006a. "Reverse Replay of Behavioural Sequences in Hippocampal Place Cells during the Awake State." *Nature* 440 (7084): 680–83. <https://doi.org/10.1038/nature04587>.
- . 2006b. "Reverse Replay of Behavioural Sequences in Hippocampal Place Cells during the Awake State." *Nature* 440 (7084): 680–83. <https://doi.org/10.1038/nature04587>.
- Foster, David J., and James J. Knierim. 2012. "Sequence Learning and the Role of the Hippocampus in Rodent Navigation." *Current Opinion in Neurobiology* 22 (2): 294–300. <https://doi.org/10.1016/J.CONB.2011.12.005>.
- Friston, Karl. 2003. "Learning and Inference in the Brain." *Neural Networks: The Official Journal of the International Neural Network Society* 16 (9): 1325–52. <https://doi.org/10.1016/j.neunet.2003.06.005>.
- . 2008. "Hierarchical Models in the Brain." *PLoS Computational Biology* 4 (11): e1000211. <https://doi.org/10.1371/journal.pcbi.1000211>.
- . 2009. "The Free-Energy Principle: A Rough Guide to the Brain?" *Trends in Cognitive Sciences* 13 (7): 293–301. <https://doi.org/10.1016/j.tics.2009.04.005>.
- Friston, Karl, Michael Breakspear, and Gustavo Deco. 2012. "Perception and Self-Organized Instability." *Frontiers in Computational Neuroscience* 6 (July): 44. <https://doi.org/10.3389/fncom.2012.00044>.
- Friston, Karl, Thomas FitzGerald, Francesco Rigoli, Philipp Schwartenbeck, John O'Doherty, and Giovanni Pezzulo. 2016. "Active Inference and Learning." *Neuroscience & Biobehavioral Reviews* 68 (September): 862–79. <https://doi.org/10.1016/J.NEUBIOREV.2016.06.022>.
- Friston, Karl J., Richard Rosch, Thomas Parr, Cathy Price, and Howard Bowman. 2017. "Deep Temporal Models and Active Inference." *Neuroscience & Biobehavioral Reviews* 77 (June): 388–402. <https://doi.org/10.1016/J.NEUBIOREV.2017.04.009>.
- Friston, Karl J., Jean Daunizeau, and Stefan J. Kiebel. 2009. "Reinforcement Learning or Active Inference?" *PloS One* 4 (7): e6421. <https://doi.org/10.1371/journal.pone.0006421>.
- Gaussier, P., A. Revel, J. P. Banquet, and V. Babeau. 2002. "From View Cells and Place Cells to Cognitive Map Learning: Processing Stages of the Hippocampal System." *Biological Cybernetics* 86 (1): 15–28. <https://doi.org/10.1007/s004220100269>.
- Gaussier, Philippe, and Stéphane Zrehen. 1995. "PerAc: A Neural Architecture to Control Artificial Animals." *Robotics and Autonomous Systems* 16 (2–4): 291–320. [https://doi.org/10.1016/0921-8890\(95\)00052-6](https://doi.org/10.1016/0921-8890(95)00052-6).
- Glover, Fred, and Rafael Martí. 1986. "Tabu Search." *Tabu Search*, 1–16.
- Goldberg, David E., and John H. Holland. 1988. "Genetic Algorithms and Machine Learning." *Machine Learning*, 1988. <https://doi.org/10.1023/A:1022602019183>.
- Goller, C., and A. Kuchler. 1996. "Learning Task-Dependent Distributed Representations By\nbackpropagation through Structure." *Proceedings of International Conference on Neural Networks (ICNN'96)* 1. <https://doi.org/10.1109/ICNN.1996.548916>.
- Greff, Klaus, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2017. "Neural Expectation Maximization." <https://doi.org/10.1093/ajae/aas111>.
- Guazzelli, Alex, Fernando J. Corbacho, Mihail Bota, and Michael A. Arbib. 1998. "Affordances, Motivations, and the World Graph Theory." *Adaptive Behavior* 6 (3/4): 435–71. <https://doi.org/10.1177/105971239800600305>.
- Gupta, Anoopum S., Matthijs A.A. van der Meer, David S. Touretzky, and A. David Redish. 2010. "Hippocampal Replay Is Not a Simple Function of Experience." *Neuron* 65 (5): 695–705. <https://doi.org/10.1016/j.neuron.2010.01.034>.
- Hebb, D. O. 1949. "The Organization of Behaviour." *Organization*, 62. <https://doi.org/citeulike-article-id:1282862>.
- Hendriks-Jansen, Horst. 1996. *Catching Ourselves in the Act: Situated Activity, Interactive Emergence, Evolution, and Human Thought. Complex Adaptive Systems*.
- Hinaut, Xavier, and Peter Ford Dominey. 2013. "Real-Time Parallel Processing of Grammatical Structure in the Fronto-Striatal System: A Recurrent Network Simulation Study Using Reservoir Computing." *PloS One* 8 (2): e52946. <https://doi.org/10.1371/journal.pone.0052946>.
- Hirel, J., P. Gaussier, M. Quoy, J. P. Banquet, E. Save, and B. Poucet. 2013. "The Hippocampo-Cortical Loop: Spatio-Temporal Learning and Goal-Oriented Planning in Navigation." *Neural Networks* 43: 8–21. <https://doi.org/10.1016/j.neunet.2013.01.023>.
- Hoffman, K. L., and B. L. McNaughton. 2002. "Coordinated Reactivation of Distributed Memory Traces in Primate Neocortex." *Science* 297 (5589): 2070–73. <https://doi.org/10.1126/science.1073538>.

- Holland, Jh H, and Js S Reitman. 1977. "Cognitive Systems Based on Adaptive Algorithms." *ACM SIGART Bulletin*, no. 63: 49. <https://doi.org/10.1145/1045343.1045373>.
- Jadhav, Shantanu P., Caleb Kemere, P. Walter German, and Loren M. Frank. 2012. "Awake Hippocampal Sharp-Wave Ripples Support Spatial Memory." *Science* 336 (6087): 1454–58. <https://doi.org/10.1126/science.1217230>.
- Jaeger, Herbert. 2001. "The 'Echo State' Approach to Analysing and Training Recurrent Neural Networks—with an Erratum Note." *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148 (34): 13.
- . 2007. "Discovering Multiscale Dynamical Features with Hierarchical Echo State Networks." <http://jpubs.jacobs-university.de/handle/579/147>.
- . 2014. "Controlling Recurrent Neural Networks by Conceptors Controlling Recurrent Neural Networks by Conceptors (Revision 2)." <http://>
- Jaeger, Herbert, and Harald Haas. 2004. "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication." *Science* 304 (5667): 78–80. <https://doi.org/10.1126/science.1091277>.
- Ji, Daoyun, and Matthew A Wilson. 2007. "Coordinated Memory Replay in the Visual Cortex and Hippocampus during Sleep." *Nature Neuroscience* 10 (1): 100–107. <https://doi.org/10.1038/nn1825>.
- Johnson, Adam, and David A Crowe. 2008. "Revisiting Tolman, His Theories and Cognitive Maps." *Cognitive Critique* 1: 43–72.
- Jong, Laurel Watkins De, Brian Gereke, Gerard M. Martin, and Jean Marc Fellous. 2011. "The Traveling Salesrat: Insights into the Dynamics of Efficient Spatial Navigation in the Rodent." *Journal of Neural Engineering* 8 (6). <https://doi.org/10.1088/1741-2560/8/6/065010>.
- Jurafsky, Daniel, and James Martin. 2017. "Hidden Markov Models." *Speech and Language Processing*, no. Chapter 20: 21. [https://doi.org/10.1016/S0959-440X\(96\)80056-X](https://doi.org/10.1016/S0959-440X(96)80056-X).
- Karlsson, Mattias P, and Loren M Frank. 2009. "Awake Replay of Remote Experiences in the Hippocampus." *Nature Neuroscience* 12 (7): 913–18. <https://doi.org/10.1038/nn.2344>.
- Kiebel, Stefan J, Katharina von Kriegstein, Jean Daunizeau, and Karl J Friston. 2009. "Recognizing Sequences of Sequences." *PLoS Computational Biology* 5 (8): e1000464. <https://doi.org/10.1371/journal.pcbi.1000464>.
- Kirkpatrick, S, C D Gelatt, and M P Vecchi. 1983. "Optimization by Simulated Annealing." *Science (New York, N.Y.)* 220 (4598): 671–80. <https://doi.org/10.1126/science.220.4598.671>.
- Kudrimoti, H S, C A Barnes, and B L McNaughton. 1999. "Reactivation of Hippocampal Cell Assemblies: Effects of Behavioral State, Experience, and EEG Dynamics." *The Journal of Neuroscience : The Official Journal of the Society for Neuroscience* 19 (10): 4090–4101. <https://doi.org/10.1523/JNEUROSCI.19-10-04090.1999>.
- Lansink, Carien S, Pieter M Goltstein, Jan V Lankelma, Ruud N J M A Joosten, Bruce L Mcnaughton, and Cyriel M A Pennartz. n.d. "Behavioral/Systems/Cognitive Preferential Reactivation of Motivationally Relevant Information in the Ventral Striatum." Accessed June 13, 2018. <https://doi.org/10.1523/JNEUROSCI.1054-08.2008>.
- Lee, Albert K., and Matthew A. Wilson. 2002. "Memory of Sequential Experience in the Hippocampus during Slow Wave Sleep." *Neuron* 36 (6): 1183–94. [https://doi.org/10.1016/S0896-6273\(02\)01096-6](https://doi.org/10.1016/S0896-6273(02)01096-6).
- Lukoševičius, M. 2012. "A Practical Guide to Applying Echo State Networks." *Neural Networks: Tricks of the Trade, Reloaded*. <https://doi.org/10.1007/978-3-642-35289-8-36>.
- Lukoševičius, Mantas, and Herbert Jaeger. 2009. "Reservoir Computing Approaches to Recurrent Neural Network Training." *Computer Science Review* 3: 127–49. <https://doi.org/10.1016/j.cosrev.2009.03.005>.
- Ma, Sheng, and Chuanyi Ji. 1998. "Fast Training of Recurrent Networks Based on the EM Algorithm." *IEEE Transactions on Neural Networks* 9 (1): 11–26. <https://doi.org/10.1109/72.655025>.
- Maass, Wolfgang, Thomas Natschläger, Henry Markram, and Natschläger T Maass W. 2002. "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations." *Neural Computation* 14 (11): 2531–60. <https://doi.org/10.1162/089976602760407955>.
- Markov, Andrey A. 1913. "Essai d'une Recherche Statistique Sur Le Texte Du Roman 'Eugène Oneguine.'" *Bull. Acad. Imper. Sci. St. Petersburg* 7.
- Marr, David. 1971. "A Simple Theory for Archicortex." *Philosophical Transactions of the Royal Society B: Biological Sciences*. <https://doi.org/10.1098/rstb.1984.0053>.

- McClelland, James L, Bruce L McNaughton, and Randall O'Reilly. 1995. "Why There Are Complementary Learning Systems in the Hippocampus and Neocortex: Insights Form the Successes and Failures of Connectionist Models of Learning and Memory." *Psychological Review* 102 (3): 419–57. <https://doi.org/10.1037/0033-295X.102.3.419>.
- Milford, Michael, and Gordon Wyeth. 2010. "Persistent Navigation and Mapping Using a Biologically Inspired Slam System." *International Journal of Robotics Research* 29 (9): 1131–53. <https://doi.org/10.1177/0278364909340592>.
- MIT. 2017. "The Emerging Science of Computational Psychiatry - MIT Technology Review." July 21. 2017.
- Nádasdy, Zoltán, Hajime Hirase, András Czurkó, Jozsef Csicsvari, and György Buzsáki. 1999. "Replay and Time Compression of Recurring Spike Sequences in the Hippocampus." *The Journal of Neuroscience* 19 (21): 9497–9507. <https://doi.org/10.1126/science.1182395>.
- Nikolić, Danko, Stefan Häusler, Wolf Singer, and Wolfgang Maass. 2009. "Distributed Fading Memory for Stimulus Properties in the Primary Visual Cortex." *PLoS Biology* 7 (12): e1000260. <https://doi.org/10.1371/journal.pbio.1000260>.
- Pearson, K. 1896. "Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 187 (0): 253–318. <https://doi.org/10.1098/rsta.1896.0007>.
- Pennartz, C. M. A. 2004. "The Ventral Striatum in Off-Line Processing: Ensemble Reactivation during Sleep and Modulation by Hippocampal Ripples." *Journal of Neuroscience* 24 (29): 6446–56. <https://doi.org/10.1523/JNEUROSCI.0575-04.2004>.
- Peyrache, Adrien, Mehdi Khamassi, Karim Benchenane, Sidney I Wiener, and Francesco P Battaglia. 2009. "Replay of Rule-Learning Related Neural Patterns in the Prefrontal Cortex during Sleep." *Nature Neuroscience* 12 (7): 919–26. <https://doi.org/10.1038/nn.2337>.
- Pezzulo, Giovanni, Caleb Kemere, and Matthijs A.A. van der Meer. 2017. "Internally Generated Hippocampal Sequences as a Vantage Point to Probe Future-Oriented Cognition." *Annals of the New York Academy of Sciences*. <https://doi.org/10.1111/nyas.13329>.
- Pezzulo, Giovanni, Matthijs A.A. van der Meer, Carien S. Lansink, and Cyriel M.A. Pennartz. 2014. "Internally Generated Sequences in Learning and Executing Goal-Directed Behavior." *Trends in Cognitive Sciences* 18 (12): 647–57. <https://doi.org/10.1016/j.tics.2014.06.011>.
- Pezzulo, Giovanni, Francesco Rigoli, and Karl J. Friston. 2018. "Hierarchical Active Inference: A Theory of Motivated Control." *Trends in Cognitive Sciences*. <https://doi.org/10.1016/j.tics.2018.01.009>.
- Piaget, Jean. 1967. *The Child's Conception of the World... Transl... Adams & Company*.
- Popa, Daniela, Sevil Duvarci, Andrei T Popescu, Clément Léna, and Denis Paré. 2010. "Coherent Amygdalocortical Theta Promotes Fear Memory Consolidation during Paradoxical Sleep." *Proceedings of the National Academy of Sciences of the United States of America* 107 (14): 6516–19. <https://doi.org/10.1073/pnas.0913016107>.
- Preston, Alison R, and Howard Eichenbaum. 2013. "Interplay of Hippocampus and Prefrontal Cortex in Memory." *Current Biology*. <https://doi.org/10.1016/j.cub.2013.05.041>.
- Puskorius, Gintaras V., and Lee A. Feldkamp. 1994. "Neurocontrol of Nonlinear Dynamical Systems with Kalman Filter Trained Recurrent Networks." *IEEE Transactions on Neural Networks* 5 (2): 279–97. <https://doi.org/10.1109/72.279191>.
- Redish, A. David, and David S. Touretzky. 1997. "Cognitive Maps beyond the Hippocampus." *Hippocampus* 7 (1): 15–35. [https://doi.org/10.1002/\(SICI\)1098-1063\(1997\)7:1<15::AID-HIPO3>3.0.CO;2-6](https://doi.org/10.1002/(SICI)1098-1063(1997)7:1<15::AID-HIPO3>3.0.CO;2-6).
- Ribeiro, Sidarta, Damien Gervasoni, Ernesto S Soares, Yi Zhou, Shih-Chieh Lin, Janaina Pantoja, Michael Lavine, and Miguel A. L Nicolelis. 2004. "Long-Lasting Novelty-Induced Neuronal Reverberation during Slow-Wave Sleep in Multiple Forebrain Areas." Edited by Wolfram Schultz. *PLoS Biology* 2 (1): e24. <https://doi.org/10.1371/journal.pbio.0020024>.
- Rigotti, Mattia, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. 2013. "The Importance of Mixed Selectivity in Complex Cognitive Tasks." *Nature* 497 (7451): 585–90. <https://doi.org/10.1038/nature12160>.
- Rigotti, Mattia, Daniel Ben Dayan Rubin, Xiao-Jing Wang, and Stefano Fusi. 2010. "Internal Representation of Task Rules by Recurrent Dynamics: The Importance of the Diversity of Neural Responses." *Frontiers in Computational Neuroscience* 4 (October): 24. <https://doi.org/10.3389/fncom.2010.00024>.
- Sanger, Terence D. 1989. "Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural

- Network.” *Neural Networks* 2 (6): 459–73. [https://doi.org/10.1016/0893-6080\(89\)90044-0](https://doi.org/10.1016/0893-6080(89)90044-0).
- Schacter, Daniel L., Donna Rose Addis, and Randy L. Buckner. 2008a. “Episodic Simulation of Future Events.” *Annals of the New York Academy of Sciences* 1124 (1): 39–60. <https://doi.org/10.1196/annals.1440.001>.
- . 2008b. “Episodic Simulation of Future Events.” *Annals of the New York Academy of Sciences* 1124 (1): 39–60. <https://doi.org/10.1196/annals.1440.001>.
- Schwindel, C. Daniela, and Bruce L. McNaughton. 2011. “Hippocampal-Cortical Interactions and the Dynamics of Memory Trace Reactivation.” *Progress in Brain Research* 193: 163–77. <https://doi.org/10.1016/B978-0-444-53839-0.00011-9>.
- Sharp, Patricia E., Hugh T. Blair, and Michael Brown. 1996. “Neural Network Modeling of the Hippocampal Formation Spatial Signals and Their Possible Role in Navigation: A Modular Approach.” *Hippocampus* 6 (6): 720–34. [https://doi.org/10.1002/\(SICI\)1098-1063\(1996\)6:6<720::AID-HIPO14>3.0.CO;2-2](https://doi.org/10.1002/(SICI)1098-1063(1996)6:6<720::AID-HIPO14>3.0.CO;2-2).
- Shin, Justin D., and Shantanu P. Jadhav. 2016. “Multiple Modes of Hippocampal–prefrontal Interactions in Memory-Guided Behavior.” *Current Opinion in Neurobiology*. <https://doi.org/10.1016/j.conb.2016.07.015>.
- Singer, Annabelle C., and Loren M. Frank. 2009. “Rewarded Outcomes Enhance Reactivation of Experience in the Hippocampus.” *Neuron* 64 (6): 910–21. <https://doi.org/10.1016/J.NEURON.2009.11.016>.
- Stephan, Klaas E., Andreea O. Diaconescu, and Sandra Iglesias. 2016. “Bayesian Inference, Dysconnectivity and Neuromodulation in Schizophrenia.” *Brain*. <https://doi.org/10.1093/brain/aww120>.
- Stephan, Klaas Enno, W. D. Penny, Rosalyn J. Moran, E. M. den Ouden, Hanneke, Jean Daunizeau, and Karl J. Friston. 2010. “Ten Simple Rules for Dynamic Causal Modeling.” *NeuroImage* 49 (4): 3099–3109. <https://doi.org/10.1016/j.neuroimage.2009.11.015>.
- Stickgold, Robert, and Matthew P. P. Walker. 2007. “Sleep-Dependent Memory Consolidation and Reconsolidation.” *Sleep Medicine* 8 (4): 331–43. <https://doi.org/10.1016/j.sleep.2007.03.011>.
- Sutherland, Gary R., and Bruce McNaughton. 2000. “Memory Trace Reactivation in Hippocampal and Neocortical Neuronal Ensembles.” *Current Opinion in Neurobiology* 10 (2): 180–86. [https://doi.org/10.1016/S0959-4388\(00\)00079-9](https://doi.org/10.1016/S0959-4388(00)00079-9).
- Sutskever, Ilya. 2013. “Training Recurrent Neural Networks.” *PhD Thesis*, 101.
- Sutton, Richard S., and Andrew G Barto. 1998. “Introduction.” *Reinforcement Learning*.
- Tani, J., and S. Nolfi. 1999. “Learning to Perceive the World as Articulated: An Approach for Hierarchical Learning in Sensory-Motor Systems.” *Neural Networks* 12 (7–8): 1131–41. [https://doi.org/10.1016/S0893-6080\(99\)00060-X](https://doi.org/10.1016/S0893-6080(99)00060-X).
- Tatsuno, M., P. Lipa, and B. L. McNaughton. 2006. “Methodological Considerations on the Use of Template Matching to Study Long-Lasting Memory Trace Replay.” *Journal of Neuroscience* 26 (42): 10727–42. <https://doi.org/10.1523/JNEUROSCI.3317-06.2006>.
- Tolman, Edward C. 1948. “Cognitive Maps in Rats and Men.” *Psychological Review* 55 (4): 189–208. <https://doi.org/10.1037/h0061626>.
- Unkelbach, Jan, Sun Yi, and J Schmidhuber. 2009. “An EM Based Training Algorithm for Recurrent Neural Networks.” *Artificial Neural Networks--ICANN 2009*, 964--974. http://link.springer.com/chapter/10.1007/978-3-642-04274-4_99.
- Valton, Vincent, Liana Romaniuk, J. Douglas Steele, Stephen Lawrie, and Peggy Seriès. 2017. “Comprehensive Review: Computational Modelling of Schizophrenia.” *Neuroscience and Biobehavioral Reviews*. <https://doi.org/10.1016/j.neubiorev.2017.08.022>.
- Vertes, Robert P., Walter B. Hoover, Klara Szigeti-Buck, and Csaba Leranth. 2007. “Nucleus Reuniens of the Midline Thalamus: Link between the Medial Prefrontal Cortex and the Hippocampus.” *Brain Research Bulletin* 71 (6): 601–9. <https://doi.org/10.1016/j.brainresbull.2006.12.002>.
- Viterbi, Andrew. 1967. “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.” *IEEE Transactions on Information Theory* 13 (2): 260–69.
- Welch, Lloyd R. 2003. “Hidden Markov Models and the Baum-Welch Algorithm.” *IEEE Information Theory Society Newsletter* 53 (4): 10–13.
- Werbos, Paul J. 1990. “Backpropagation Through Time: What It Does and How to Do It.” *Proceedings of the IEEE* 78 (10): 1550–60. <https://doi.org/10.1109/5.58337>.
- Wikenheiser, Andrew M., and A. David Redish. 2014. “Decoding the Cognitive Map: Ensemble Hippocampal Sequences and Decision Making.” *Current Opinion in Neurobiology* 32: 8–15.

<https://doi.org/10.1016/j.conb.2014.10.002>.

- Wilson, Stewart W. 1991. "The Animat Path to AI." *From Animals to Animats 1*, 15–21.
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.2047>.
- Yamashita, Y, and J Tani. 2008. "Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment." *PLOS Computational Biology* 4 (11).
- Yang, Jinhui, Xiaohu Shi, Maurizio Marchese, and Yanchun Liang. 2008. "An Ant Colony Optimization Method for Generalized TSP Problem." *Progress in Natural Science* 18 (11): 1417–22.
<https://doi.org/10.1016/j.pnsc.2008.03.028>.