The transient and dynamic nature of splitter cells derive from behavior

Naomi Chaix-Eichel[1,2,†], Snigdha Dagar[1,2,†], Fréderic Alexandre[1,2, ‡], Thomas Boraud[1,‡], Nicolas P. Rougier[1, 2,‡,*],

1 Univ. Bordeaux, CNRS, IMN, UMR 5293, F-33000 Bordeaux, France
2 Univ. Bordeaux, CNRS, Bordeaux INP, INRIA, LaBRI, UMR 5800, F-33400 Talence, France

*** nicolas.rougier@inria.fr**
†These authors contributed equally to this work.
‡These authors also contributed equally to this work.

Abstract

Over the past decades, the hippocampal formation has undergone extensive studies leading researchers to identify a vast collection of cells with functional properties. The origin of such cells has been the topic of several investigations complemented with carefully crafted models whose latest hypothesized the role of temporal sequences to be the probable cause of observed spatial properties. We aim at investigating whether a random recurrent structure is sufficient to allow for such latent sequence to appear. To do so, we simulated an agent with egocentric sensory inputs that must navigate and alternate choices at intersections. We were subsequently able to identify several splitter cells inside the model. Lesioning those cells induce the incapacity for the agent to solve the task unless some hyper-parameters are tuned. In such case, new splitter cells appear even though the initial structure of the networks has been conserved. These results tend to show that splitter cells activity may appear spontaneously inside a random reccurrent network as long as the agent is performing the alternating task. These results may also modify our view regarding the origin on splitter cells. They are not the cause nor the consequence of learning and behavior.

Keywords Hippocampal formation, reservoir computing, spatial navigation, splitter cells.

1 Introduction

The hippocampal formation has been heavily studied in the past few decades [Buzsáki and Moser, 2013, Hartley et al., 2014, Maguire et al., 1996, Moser et al., 2017] and researchers have since then established a huge repertoire of cells displaying very specific properties. In his book "The Brain from Inside out" [György Buzsáki, 2019], György Buzsáki named a few of them in a footnote (page 356): *place cells, time cells, grid cells, head direction cells, boundary vector cells, border cells, object cells, object vector cells, distance cells, reward cells, concept cells, view cells, speed cells, memory cells, goal cells, goal direction cells, splitter cells, prospective cells, retrospective cells, island cells, ocean cells, band cells, object cells, pitch cells, trial order cells, etc.* Each and every cells have been characterized in terms of correlation between their activity and some combination

of high level property involving space, time and internal state. This is the case for place (cell fires when animal is in a specific place), head direction (cell fires preferentially when head is pointing toward a specific direction), time cells (cell fires at successive moments), etc. Facing such huge repertoire, one may legitimately wonder how the brain orchestrates all these information and makes use of it to ensure survival. Part of the answer is given by György Buzsáki himself in the same footnote where he further explained that *physiological attributes of neurons in the hippocampal-enthorinal system ... might be explained by the apparent distinctiveness of a few sets of continuous distributions*. In other words, even though we can and we do observe these cells *in vivo*, they might be a simple epiphenomenon: their activity might be correlated to some unknown latent variables.

This is actually hardly different from the hypothesis provided by Raju et al. [2024] where authors suggest that spatial representations are not explicitly encoded but emerge as a byproduct of sequence learning. Their model is based on a variant of Hidden Markov Model (HMM) theory and employs a clone-structured causal graph (CSCG) [George et al., 2021] to differentiate between various sequences of input. In this framework, multiple "clones" represent the same observation across different contexts, enabling the model to handle ambiguous sensory input sequentially. Consequently, the representation of space arises implicitly through the process of sequence learning, rather than being an explicit function of the hippocampus. Other research have similarly challenged the traditional space-centric view of the hippocampus. Using the concept of successor representation [Dayan, 1993], Stachenfeld et al. [2017] suggest that hippocampal cells encode a predictive map rather than a purely euclidean spatial map. Sanders et al. [2020] propose that the hippocampus builds more abstract representations of environmental structure based on the animal's subjective beliefs rather than directly representing objective spatial properties of the environment. Whittington et al. [2020] introduced the Tolman-Eichenbaum machine (TEM), a model that unifies spatial and non spatial functions, arguing that the emergence of the hippocampal cells emerge as a result of the system learning to represent abstract relationships and can represent both spatial and non-spatial information.

In this study, we align with current approaches and propose to reconsider the role of hippocampal cells. More precisely, we decided to focus our study on splitter cells about which Duvelle et al. [2023] published recently an extended review of the literature. Splitter cells encode information about the past or the future trajectory ([Wood et al., 2000]). In other words, they demonstrate different firing patterns depending on the animal's origin or intended destination, even if the animal is in the same physical location.

We adopt a novel perspective based on György Buzsáki's interpretation of the brain as a large reservoir of neurons with pre-existing patterns of activity [György Buzsáki, 2019]. According to this view, when a new experience occurs, the brain assigns it to a pre-existing pattern from a vast reservoir of internally generated patterns, thereby assigning behavioral meaning to the neuronal sequence. This mechanism is considered more biologically plausible than creating entirely new patterns for each experience. Building on this concept, we propose to study a reservoir computing model composed of a pool of recurrently interconnected neurons, representing a vast reservoir capable of generating an extensive repertoire of neuronal patterns. The model's synaptic weights are randomly initialized and kept fixed, implying that the neuronal patterns within the reservoir remain unchanged after learning. The only trained part is the readout layer that connect the reservoir to the output. This fixed-weight approach do not model plasticity in the reservoir, but we argue that it is sufficient and consistent with György Buzsáki [2019]'s perspective. This is supported by experiments showing that learning only modestly and transiently impacts network dynamics [Golub et al., 2018].

Furthermore, this view of the navigation system allows for a simple model that is easy to understand and manipulate, which is our main novelty and contribution compared to the other models listed: we don't require complex architectures or concepts to study splitter cells, just a pool of randomly interconnected neurons. Despite its simplicity, this model has already demonstrated great success and robustness in several works by Eric A. Antonelo [Antonelo and Schrauwen, 2012, 2009, Antonelo et al., 2007, 2008], and has even been successfully implemented on real robots navigating spatial environments [Aislan Antonelo and Schrauwen, 2015]. We demonstrate that this model can robustly solve a T-maze alternation task, specifically within a continuous state-space. Additionally, like the model in Raju et al. [2024], it handles ambiguous sensory inputs reflecting a more realistic scenario. Rather than relying on allocentric information such as spatial coordinates, the model has access only to partially observable inputs, i.e. egocentric information, like distance sensors. These sensors can read identical values in different locations within the maze, making the sensory information ambiguous. This feature increases the task's realism and difficulty and distinguishes our approach from George et al. [2021], Stachenfeld et al. [2017], Whittington et al. [2020], where such ambiguity was not explicitly addressed.

We show that the internal state of some units within the model exhibits similar characteristics to biological splitter cells. By conducting a theoretical analysis similar to that performed by Duvelle et al. [2023], we examined various hypotheses and their implications within the reservoir computing model. The results of our study demonstrate that the majority of these hypotheses are indeed validated by the reservoir computing model. While the model fails to solve the alternation task when splitter cells are removed, we show that re-optimizing the model without the splitter cells allows it to do the task again, with new splitter cells emerging. This strongly suggests that splitter cells arise from the behavior (the sequence of actions within the maze) rather than the other way around: it is not that the bot navigates because splitter cells exist. Additionally, we show that our model can decode place cells, head direction cells, and decision cells. Overall, this work aligns with previous studies but employs a simpler model consistent with György Buzsáki [2019]'s perspective.

2 Materials and Methods

2.1 Environment

Task presentation. The class of tasks called spatial alternation has been widely used to study hippocampal and working memory functions [Frank et al., 2000]. For the purpose of our investigation, we simulated a continuous versions of the task, wherein an agent must navigate through an 8-shaped track and alternate between right and left choices at a decision point, then returns to the central corridor, essentially following an 8-shape trace while moving (see 'L-R' of Figure 1-A). This 8-maze environment offers an ideal setup for observing splitter cells when the agent enters the central corridor, because the agent can have various past trajectories leading to the central corridor and different future trajectories upon exiting it. While the classical alternation task produces trajectories with both the same past and future (e.g., right-left and left-right), a more challenging variant requires the agent to alternate every two steps (see 'LL-RR' in Figure 1-A), generating left-left-right-right sequences. This variation allows trajectories with the same past but different futures (e.g., right-left vs. right-right) or different pasts but the same future (e.g., right-right vs. left-right). Consequently, this variant facilitates distinguishing between retrospective splitter cells (encoding past trajectories) and prospective splitter cells (encoding future trajectories).

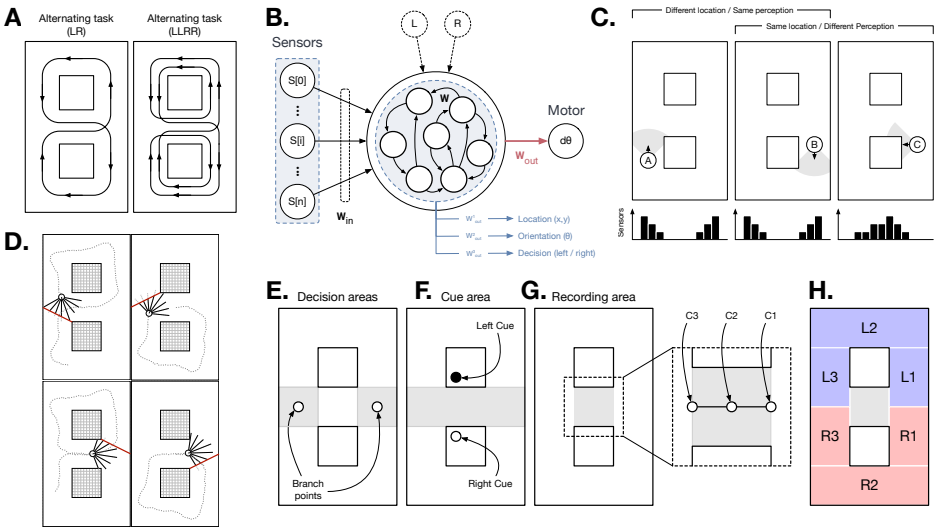

Figure 1. A. An expanded view of a T-Maze. At the decision point, the agent has to decide whether to go right (R) or left (L). The first task involves alternating every time it is located at the decision point, consequently doing the sequence of decision 'R-L-R-L-R...' . The second task involves alternating every two times, consequently doing the sequence of decision 'R-R-L-L-R-R-...'. **B. Model Architecture**. Composed of 8 sensor inputs, a reservoir and a motor output (relative orientation change). The black arrows are fixed connections while the red arrows are plastic connections and are trained. A left (L) and right (R) cue can be fed to the model depending on the experiment. The reservoir states can be used to decode supplementary information such as location (x,y coordinates), orientation (absolute θ) and decision to go left or right. **C. Ambiguous egocentric sensory inputs during the agent simulation**. In several locations within the maze, such as when the bot is at positions A and B, the sensors record identical distance values despite the bot being in different positions. This illustrates the reservoir model's difficulty in handling ambiguous sensory information. Additionally, there are instances where the bot, despite being at the same location, receives varying inputs, as seen in the comparison between positions B and C. **D. Generation of the 8-shape trajectory** with the addition of walls at the intersection points. **E. Decision areas** located at the intersection points of the maze. **F. Addition of contextual cues in the central corridor.** In the model, they are formalized as additional inputs corresponding to binary values, set to 1 only when the agent is expected to turn in the corresponding direction. **G. Recording area of the splitter cells.** Splitter cell recordings require positioning in regions with overlapping trajectories, such as the central corridor of the maze.**H. Color code of each zone.** Firing activity is color-coded based on the bot's direction: blue for approaching from the left and red for approaching from the right.

2.2 Modeling framework

Reservoir computing. Solving an alternating task necessitates the presence of a functional working memory, as the agent must retain information about its previous direction, such as turning right in the prior loop, to inform its decision to turn left at the subsequent decision point. Reservoir computing emerges as a promising solution for it, thanks to its fading memory Pascanu and Jaeger [2011]. This attribute enables the system to retain past experiences, facilitating their recall for subsequent decisions. Moreover, the reservoir computing approach has been shown to exhibit some biologically plausible characteristics [Lukoševičius and Jaeger, 2009]. Consequently, our model consists of a reservoir computing network of type Echo State Network (ESN) [Jaeger, 2007] that controls the movement of the agent solving a continuous navigation task in the 8-maze of Figure 1-B. An ESN is a recurrent neural network (called reservoir) composed of randomly connected units, associated with an input and an output layer. Inputs are projected into the reservoir, a non-linear, high-dimensional space, allowing the integration of information over time and space. Only the output neurons, referred as the readout neurons, are trained (in red in Figure 1-B). The neurons have the following dynamics:

$$\frac{1}{\alpha}\frac{d\mathbf{x}}{dt} = -\mathbf{x} + \tanh(W.\mathbf{x} + W_{in}.\mathbf{u}) \quad and \quad \mathbf{y} = W_{out}.\mathbf{x} \tag{1}$$

where $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{y}$ represent the reservoir states, input, and output. W, W_{in}, and W_{out} are weight matrices, while *tanh* refers to the hyperbolic tangent function. α refers to the leak rate, a crucial parameter of the ESN that plays a central role in controlling the memory and timescale of the network's dynamics: a low leak rate indicates a longer memory and a slower dynamics, whereas a high leak rate leads to a shorter memory but a higher speed of update dynamics [Lukoševičius, 2012]. The model was built thanks to the python library ReservoirPy [Trouvain et al., 2020].

Training. Only the output weights W_{out} are trained, using supervised learning. We tested two categories of training: offline and online learning rules. The offline learning rule uses linear ridge regression method (the Tikhonov regularization) on pre-generated data:

$$W_{out} = Y^{target}X^T(XX^T + \beta I)^{-1} \quad with \quad X = [1, \mathbf{u}, \mathbf{x}] \tag{2}$$

where Y^{target}, β and I are respectively the target signal to approximate, the regularization coefficient and the identity matrix. The online learning approach employs methods that enable task learning using only local temporal information, which is more biologically plausible than the global optimization of linear regression. Specifically, we utilized online learning by incorporating Recursive Least Squares (RLS) for rapid weight updates. The readout weights are adjusted according to the following equation: It iteratively applies the following equations to train the readout of the networks:

$$\mathbf{W_{out}}(t) = \mathbf{W_{out}}(t - \Delta t) - e(t)\mathbf{P}(t)X(t) \tag{3}$$

With:

- $e(t)$ is the error at time step t: $e = Y^{target}(t) - Y(t)$
- $P(t)$ is the inverse correlation matrix at time step t: $\mathbf{P}(t) = \mathbf{P}(t - \Delta t) - \frac{\mathbf{P}(t-\Delta t)X(t)X^T(t)\mathbf{P}(t-\Delta t)}{1+X^T(t)\mathbf{P}(t-\Delta t)X(t)}$

This iterative process allows for the effective training of the network's readout by continuously updating the weights based on local error feedback.

In order to generate data for learning, we implemented a simple Braintenberg vehicle [Braitenberg, 1986] where the agent moves automatically with a constant speed and changes its orientation according to the values of its sensors. The model takes as input the sensor values $S_0, ..., S_n$ of the agent and outputs the next variation of orientation $d\theta$ (see Figure 1-B). At each time step the sensors measure the distance to the walls and the bot turns in order to avoid the walls. The sensors are limited in their ability to calculate distance, saturating for distances exceeding 100 distance units. Additionally, the sensor readings are affected by Gaussian noise, represented as $N(0, 0.03)$. A reading of 0 indicates proximity to an object, while a reading of 1 signifies that the object is far away or not detected. The sensors can record identical distance values even when the bot is not at the same spatial location in the maze, as illustrated in Figure 1-C. This phenomenon highlights the ambiguity of the sensory input. The challenge for the model lies in disambiguating this sensory information to accurately determine the bot's location. The decision to use distance sensors as input provides a simplified egocentric representation of the environment, abstracting the complexities of the multiple sensory modalities involved in visual and auditory processing that animals like rats typically utilize. However, we believe this simplification is justified, as the distance sensors still capture ambiguous sensory information. Furthermore, some animals, such as bats [Wenstrup and Portfors, 2011] and other species [Brinkløv et al., 2013, Evans, 1973], utilize echolocation, which offers distance information analogous to that provided by distance sensors.

At each timestep, the position of the bot is updated as follows:

$$\theta(t) = \theta(t-1) + 0.01 \sum_i \alpha_i s_i \tag{4}$$

$$p(t) = p(t-1) + 2 * (cos(\theta(t)) + sin(\theta(t))) \tag{5}$$

where $p(t)$ and $p(t+1)$ are the positions of the agent at time step t and $t+1$, $\theta(t)$ is the orientation of the agent, calculated as the weighted (α_i) sum of the values of the sensors s_i. The norm of the movement is kept constant and fixed at 2.

The ESN is trained using supervised learning with data containing samples from the desired 8-shaped trajectory. Since the Braitenberg algorithm only aims at avoiding obstacles, the agent is forced to follow the desired trajectory thanks to the timely addition of walls at the intersection points as shown on Figure 1-D. After generating the right pathway, the added walls are removed and the true sensor values are gathered as input. Gaussian noise is introduced to the robot's sensor readings at each time step to enhance the robustness of the training process. Approximately 10,000 time steps were generated, equivalent to 14 complete alternating 8-loops, and divided into 75% for training and 25% for testing. The training and testing processes completed in just a couple of seconds, making the simulation process fast and easy to use, modify, implement, and optimize.

Modeling Contextual Cues. Two model variants are studied. The first relies solely on sensory input from the bot to make decisions. In the second variant, the model is helped with two contextual cues alongside the sensory inputs. These cues, which can be interpreted as visual indicators in the central corridor (see Figure 1-F), indicate to the bot's which next direction it has to take at the next decision point. If the left cue is active, the bot is instructed to turn left, and similarly for the right cue. These cues are represented by two additional inputs, labeled "L" for the left-turn cue and "R" for the right-turn cue, as shown in Figure 1-B. These are binary values, set to 1 when the bot is expected to turn in the corresponding direction and only active when the bot gets into the central corridor.

2.3 Analysis

Task solving. After the training phase, we look at the model's ability to solve the task by directly looking at the online simulation. If the bot can perform the task and execute at least five loops (one loop is composed of 700 time steps with one alternation, going one time to the right and one time to the left), the model is considered to have successfully solved the task. When the agent navigates the maze, the reservoir's internal states are recorded and analyzed using two methods: single-cell level analysis and population-level analysis.

Single-cell level analysis. We analyze the mean firing activity of individual neurons within the reservoir to observe whether we can observe splitter cells. The differentiation in mean firing activity on overlapping trajectories, i.e. in the central corridor of the 8-maze (see Figure 1-G), serves as a qualitative indicator of splitter cell behavior. We used one of the detection method cited in Duvelle et al. [2023]-Box 3, referred as the ANCOVA (Analysis of Covariance) method, used to detect splitter cells in hippocampal studies [Kinsky et al., 2020, Wood et al., 2000]. It is a statistical technique to assess whether there are significant differences in neuronal firing rates across different behavioral conditions while controlling for other variables that might affect neuronal activity. We record the firing activity of individual neurons in the reservoir as the agent traverses the central stem of the T-maze, which corresponds to a shared portion of different trajectories. The firing rates are grouped based on the two different conditions: right to left (RL) or left to right (LR). We recorded at least 7 activities per trajectory. The main factor of the ANCOVA analysis is the trajectory type, and the covariates are the lateral position of the bot during cell recording and the orientation of the bot. We did not include the bot's speed as a covariate, as it remains constant throughout the experiment. The resulting F-statistic and p-value determine if there's a statistically significant difference between the trajectories. If a neuron shows a statistically significant difference in firing rates between conditions with $p < 0.001$, it is classified as a splitter cell. We visualize the detected splitter cells by generating their raster plot. Their activity are recorded in the central corridor during overlapping trajectories, subsequently turned into a dynamic firing rate according to the Poisson process [Dayan and Abbott, 2005, Heeger et al., 2000]. The activity of each neurons are scaled to firing rates:

$$f_{rate}(t) = baseline_rate + V(t) \times max_rate_change \tag{6}$$

where $f_{rate}(t)$ corresponds to the computed firing rate at time t, $V(t)$ is the normalized activity of the reservoir neuron at time t, $baseline_rate$ is the baseline firing rate (in spikes per second) and set to 45 and max_rate_change is the maximum deviation from the baseline also set to 45. The firing rate is subsequently converted into spike counts using a Poisson process:

$$count_{spike}(t) \sim Poisson(f_{rate}(t) \times dt) \tag{7}$$

where $count_{spike}(t)$ represents the number of spikes occurring at time t, given the firing rate $f_{rate}(t)$. The Poisson process models the probabilistic occurrence of spikes over time with its paramater $\lambda = f_{rate}(t)dt$.

Population-level analysis. We analyze the reservoir state of the ESN at the population level using Principal Component Analysis (PCA) as a dimensionality reduction technique to identify patterns and significant features within the processed data.

Lesioning splitter cells within the model. We investigated the effects of lesioning the detected splitter cells within the model. To do this, we first identify the splitter cells and subsequently inactivate the connections associated with them by setting their weight values to 0 in W_{in}, W, and W_{bias} (see Figure 3). This process effectively rendered the splitter cells silent. The resulting model, denoted as the *Lesioned model*, is not always able to successfully perform the alternating task and requires some adjustments:

- We bypass its disability of solving the task by feeding it with the exact sequence of sensory inputs the bot would receive if it were correctly performing the alternation task, without using the output model to compute the next position. This version of the model is referred as the *Forced Lesioned model*.
- We re-optimize the *Lesioned model* so as to make it performing the alternating task again (without modifying its connectivity structure). This version of the model referred as the *Optimized Lesioned model*.

The *Lesioned*, the *Forced Lesioned* and the *Optimized Lesioned* models possess the same connectivity structure. We subsequently analyze their internal activity when executing the task (see Figure 3).

Decoding other hippocampal cells. In addition to splitter cells, we explore whether the reservoir's dynamical systems could encode other hippocampal cell-like activities, such as head-direction cells, decision cells, and place cells. Since these cells are not explicitly represented in the reservoir, we employ an additional readout, referred as the "decoder readout", trained using offline supervised learning to decode their activity. The decoder was trained on 7500 time steps and tested on 2500 time steps, with a regularization factor of 1e-3 for all readouts. For place cells, we generate training data by dividing the maze into 60 grid squares (10 by 6). Each square is labeled as 1 if the bot is located within it, otherwise it is set to 0. Head-direction cells, which fire in response to a specific orientation,
are decoded by generating output vectors representing eight distinct orientations intervals: $[(-\pi, -3\pi/4), (-3\pi/4, -\pi/2), (-\pi/2, -\pi/4), (-\pi/4, 0), (0, \pi/4), (\pi/4, \pi/2), (\pi/2, 3\pi/4), (3\pi/4, \pi)]$. For example, if the bot's orientation is between $3\pi/4$ and π, the output vector will be $[0, 0, 0, 0, 0, 0, 0, 1]$. For decision cells, we generate training data by recording the bot's next decision (right or left) at one of the intersection point following the central corridor (see Figure 1-E), labeling these decisions as 0 (left) or 1 (right). These decoder readouts allow us to assess whether this spatial information is encoded within the reservoir dynamics. To complement this analysis, we also examine whether this spatial information could be directly decoded from the sensory inputs alone. To do so, we employed a Support Vector Machine (SVM) classifier [Cortes and Vapnik, 1995] to categorize the sensory inputs, evaluating whether it was possible to predict the robot's position, orientation, or decision based solely on the sensor data.

3 Results

3.1 Model performances

Robustness. Using an architecture composed of fixed recurrently connected neurons (the reservoir) and a trainable readout layer trained with linear regression, both models, with and without contextual cues as input, were able to successfully perform the alternating task 'R-L' of Figure 1-A for at least five loops. Both models were successfully trained using two different training rules, offline regression and online RLS learning, demonstrating the flexibility of the learning methods. The optimized

hyperparameters of the model are presented in the Tables section. However, the model is quite robust to variations in these hyperparameters. This robustness was also demonstrated in the work of Aislan Antonelo and Schrauwen [2015]. While we fixed the number of units to 1000, a model with 500 neurons also performed well, provided the other hyperparameters were optimized accordingly. The values for input and output connectivity could also vary across a wide range. The leak rate and spectral radius were among the most critical parameters, requiring joint optimization to manage the level of chaoticity, as well as the timescale and memory of the model. The regularization parameter, which helps reduce overfitting, was another important aspect to optimize. All parameters were manually optimized, and the optimal settings were quite straightforward to identify. Overall, we present a set of hyperparameters that yielded good performance, but many other combinations also worked, underscoring the robustness of our models. Due to the slower pace of online learning, we opted to conduct the rest of the analysis using the offline learning rule.

Additionally, both models were also able to perform the 'RR-LL' version of the alternating task of Figure 1-A. However, this task is more challenging as it requires greater working memory capacity, making the models more sensitive to changes in hyperparameters. While the task enabled us to distinguish between retrospective and prospective cells, we did not conduct further analysis using this variant.

Influence of inner structure. We subsequently evaluated the extent to which the architecture of the reservoir model influenced its ability to solve the 'R-L' alternating task. To achieve this, we generated 15 random connectivity matrices W and tested them using the same set of hyperparameters, including the leak rate, regularization parameters, and input and reservoir connectivity. The matrices W were generated with a fixed connectivity, where the non-zero elements were drawn from a normal distribution $\mathcal{N}(0,1)$. Each matrix was then scaled by a factor of $\frac{sr}{\rho(\mathbf{M})}$ to ensure the matrix has the desired spectral radius as shown in equation 8.

$$W = \frac{sr}{\rho(\mathbf{M})}\mathbf{M}, \quad where \quad \mathbf{M}_{ij} = \mathcal{N}(0,1) \quad if \quad non-zero \quad element \tag{8}$$

All connectivity matrices W were effective in solving the alternating task, with no changes to any hyperparameters other than the spectral radius. This result highlights the model's robustness to different architectures, indicating that its success did not rely heavily on the specific architecture, apart from the requirement for some level of recurrence.

3.2 Splitter cells

3.3 Splitting effect at the single cell level

In their recent review, Duvelle et al. [2023] provided a comprehensive analysis of splitter cells, examining both experimental and theoretical aspects. Our goal is to replicate this analysis and verify whether their hypotheses hold true within the reservoir model. Using the ANCOVA method, we identified 341 splitter cells that exhibited significantly different firing rates in the central stem ($p < 0.001$), even when the bot received identical inputs in both cases. This difference in firing activity is illustrated in Figure 2-A, which shows neuronal activity recorded as the bot approaches the central corridor (grey zone in Figure 2-G). Each trace corresponds to the firing activity of a splitter cell in the central corridor over 50 time steps. The differences in firing activity are evident, with red traces representing activity when the bot is moving from left to right and blue traces when moving from right to left (see the color code in Figure 1-H). Raster plots of the same splitter cells are generated from the Poisson process and depicted in the

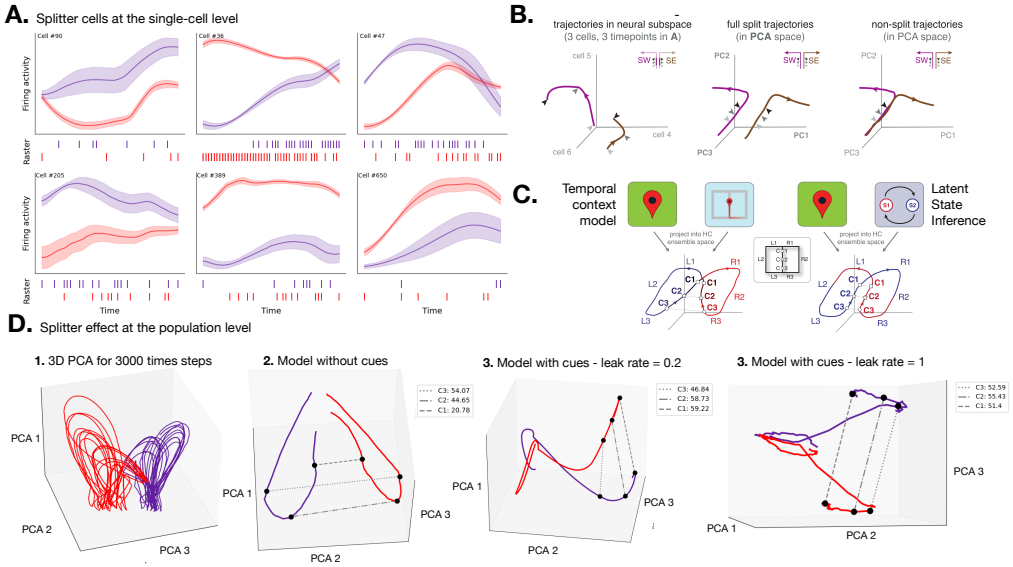

Figure 2. A. Comparing firing rates of 6 splitter cells in the central stem. The identification of these cells was achieved using the ANCOVA method, which revealed significantly different firing activities ($p < 0.001$) in overlapping trajectories 'RL' and 'LR' (shown in the grey area of Figure 1-G). The activity of splitter cells is recorded when the bot enters the central corridor (grey zone in the bottom right figure). Each trace represents the firing activity of a splitter cell in the central corridor over 50 time steps. Each line corresponds to the mean firing activity of at least 5 traversals, and the shaded areas indicates the standard deviation. The differences in firing activity are evident: traces are colored red for activity when the bot is moving from left to right and blue when moving from right to left. Bellow each graph, the corresponding raster plots are depicted showing a difference in spiking activity rate between the overlapping trajectories during one traversal. **B. Hypothesized shape of neural trajectories in neural space with and without the splitting effect**. When applying 3D PCA to the entire population, Duvelle et al. [2023] proposed that neural trajectories remain distinct when the subject enters the central corridor if the splitting effect occurs. In contrast, these trajectories overlap in the absence of a splitting effect. **C. Temporal Context Model vs. Latent State Inference**. In the hypothesized PCA Space illustrating the temporal context model (TCM), the population activity at a specific location exhibits closer proximity in PCA space when the recent past is similar and greater distance when it is less similar. This is demonstrated in the distances between markers C1, C2, and C3. In the hypothesized PCA illustrating the latent-state inference (LSI), the distances between the markers C1, C2, and C3 don't vary. The Figure is taken from Duvelle et al. [2023]. **D. Population analysis with 3D PCA**. **D.1**. 3D PCA applied during multiple loops. The figure shows two sub-attractors, which are linearly separable, that correspond to the two parts of the 8-shape trajectory. **D.2**. PCA applied to the model without contextual cues, resembling the TCM model. The increasing distances between C1, C2, and C3 markers suggest an amplified influence of past trajectories on neural activity. **D.2**. PCA applied to the model with contextual cues (leak rate $\alpha = 0.2$), similar to the LSI model introduced. The decreasing distances between C1, C2, and C3 markers indicate more distinct neural activity between the two trajectories, reflecting a stronger association with one of the two discrete states. **D.2** PCA applied to the model with contextual cues (leak rate $\alpha = 1$). The distances between markers C1, C2, and C3, which remain constant as the agent enters the central corridor. This behavior parallels the dynamics observed in the LSI model.

bottom part of each graph of Figure 2-A, illustrating the average firing activity of a subset of splitter neurons when the bot is traversing the central corridor. Each individual neuron displays a distinct average activity pattern based on the trajectory, even when recorded at the same spatial location.

3.4 Splitting effect at the population level

Temporal Context Model versus Latent State Inference Model The splitting effect is observed at the population level by analyzing trajectories in the neural space. Splitter activity is characterized by distinct neural trajectories that emerge when the animal is at the same spatial location as depicted in Figure 2-B from Duvelle et al. [2023]. In Figure 2-C, Duvelle et al. [2023] hypothesized neural activity trajectories during the execution of an alternation task. Two trajectories are shown: a blue one representing movement from the left loop to the central corridor, and a red one showing movement from the right loop to the same central corridor. Despite sharing the same physical location when the animal traverses the central corridor (where the markers C1, C2, and C3 are placed), these trajectories occupy distinct regions in PCA space. Without the splitting effect, the neural representations of these three markers would overlap (see Figure 2-B), as they correspond to the same spatial location. However, they are distinctly separated with a measurable PCA distance, indicating the presence of differentiated neural activity. Duvelle et al. [2023] argue that this observation provides evidence for the presence of splitter cells in the neural population.

Furthermore, the splitting effect can be studied through the lens of two different models introduced in Figure 2-C. The first model is the "Temporal Context Model" (TCM), which is sensitive to prior experiences. This means that the current neural state of the animal varies depending on its past. When the recent past is similar, population activity at a given location in the maze exhibits closer proximity in PCA space. Conversely, when the recent past differs, the distances increase. This phenomenon is demonstrated by the varying distances between the markers C1, C2, and C3, each representing specific positions within the central corridor. When the animal reaches C3 in the maze, it has traversed the entire central corridor, resulting in a shared trajectory that accounts for the smaller distance between the C3 markers. In contrast, the animal's past experiences at C1 differ significantly, which explains the greater distance between the C1 markers. The second model is the "Latent State Inference" (LSI) model. Unlike the TCM, the LSI model organizes experiences into discrete states, resulting in a neural state less sensitive to historical context. As a result, the distances between the two neural trajectories in the central corridor remain constant: the distance between C1 markers is equal to that between C2 markers, and the distance between C3 markers is also consistent. In the LSI model, experiences can be categorized into distinct states without strong dependencies on past experiences. For example, in scenarios where each stimulus is associated with a specific state, the categorization remains valid regardless of how previous stimuli were encountered. Duvelle et al. [2023] illustrates this with the example of one arm being rewarded while the other is not.

Implicit approximation of the state-space environment and splitting effect at the population level We proceed to a similar population analysis by applying 3D PCA to the reservoir states. This analysis was conducted for both the model without contextual cues (Figure 2-D1-D2) and with contextual cues (Figure 2-D3-D4). The blue trace indicates the agent traversing the left loop and heading to the the central corridor, while the red trace followed indicates the agent traversing the right loop and heading to the central corridor. These two trajectories occupy distinct regions in the PCA space, despite they share the same location when the bot is traversing the main corridor. The three markers (C1, C2, C3) indicating specific positions within the main corridor are distinctly separated with a measurable PCA distance, demonstrating the presence of splitter cells. The PCAs show two linearly separable sub-attractors, corresponding to the two loops of the 8-shape. This result demonstrates the ESN's ability to implicitly approximate the state-space of the 8-maze environment through continuous temporal dynamics. The model achieves this by mapping the input data into a

higher-dimensional space [Jaeger, 2001]. The recurrent connections within the reservoir enable the network to incorporate and reverberate past information, allowing for the encoding of temporal context and the integration of the past trajectory.

The model without contextual cues corresponds the TCM. In the model without contextual cues, we observed decreasing distances between C1,C2 and C3 markers in the central corridor (Figure 2-D2), which aligns with the TCM (Figure 2-C). This makes sense, as the model is influenced not only by current sensory inputs but also by recently experienced events, as the past trajectory of the bot is stored through the reverberation of recurrently connected neurons, allowing for a context-dependent response to present events.

The model with contextual cues corresponds the LSI. In the model with contextual cues, we observed the opposite effect: as the bot enters the central corridor, the distances between neural trajectories increase, indicating more distinct neural activity and a stronger association with one of the two discrete states (Right or Left). We conducted this analysis with varying values of the leak rate, which corresponds to the hyperparameter that controls sensitivity to past activities. A low leak rate results in greater memory capacity, making the model more sensitive to previous information, while a high leak rate indicates smaller memory capacity and reduced sensitivity to past experiences. Figure 2-D3 displays activity for a leak rate of 0.2, which reflects high past dependencies; nonetheless, we still observed a more discrete configuration consistent with the LSI model (see Figure 2-C). Furthermore, we applied the same PCA analysis to a model with cues and a leak rate of 1 (indicating no memory). The resulting PCA, shown in Figure 2-D3, reveals consistent distances between neural trajectories in the central corridor, also aligning with the LSI model. The presence of contextual cues provides direction information at each time step within the central corridor, positioning the model in a more discrete configuration. Consequently, the model does not require extensive memory of the past trajectory to successfully complete the navigation task.

These results confirm Duvelle et al. [2023]'s theoretical framework introducing TCM and LSI. These two theories aren't mutually exclusive, leaving possibility for hybrid models that contain elements of both. While reservoir computing primarily relies on maintaining working memory within the reverberation of reservoir neurons, it's natural to view this model as having a temporal gradedness comparable to TCM. This forms the basis of reservoir computing. By incorporating contextual cues as input, it is possible to transition from a TCM to LSI model. Adjusting the leak rate of the models allows for a flexible hybrid model, combining temporally decaying representations of the past with discrete state representations. For instance, in tasks like the continuation alternation task, where task-state switches occur with changes in cue values, the internal states shift between distinct regions of activity space while still permitting continuous neural activity.

3.5 Silencing splitter cells inside the models

Model without cues. We identified 344 splitter cells (SC1) in the initial model without contextual cues. 50 of the SC1 cells were subsequently silenced, and the model re-optimized by adjusting only the leak rate and spectral radius, without altering the structural connectivity (see Figure3-A1). Adjusting the leak rate and spectral radius compensated for the absence of splitter cells by introducing more chaos and inertia into the reservoir dynamics, allowing the model to maintain task performance. Despite the inactivation of SC1, the *Optimized Lesioned model* successfully performed the alternation task. Notably, 87 new splitter cells (SC2) were found, with 40 distinct from

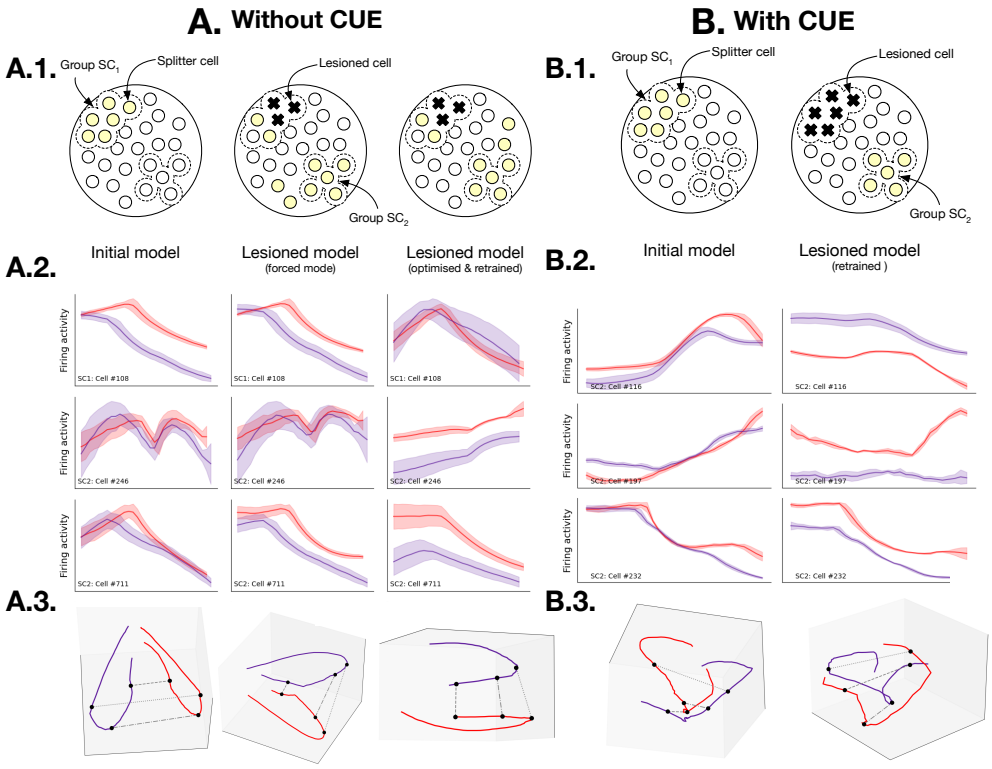

Figure 3. A1. Silencing a fraction of splitter cells in the model without cues. Yellow cells represent splitter cells (SC), and those marked with a black cross are the silenced splitter cells. First, all splitter cells are identified in the model (left). A subset of these cells is then silenced, resulting in the Lesioned Model (middle). While the lesioned model cannot perform the task independently, it is forced to process the exact sensory input it would have received if it were performing the task. This leads to the emergence of a new set of splitter cells (SC2) in the lesioned model. By reoptimizing the lesioned model's leak rate and spectral radius (right), the optimized lesioned model regains its ability to perform the task, accompanied by the emergence of yet another set of splitter cells (SC2). **A.2. Activity of selected cells across the three models**. Each row corresponds to the same neuron across the three models. Row 1: Cell 108 acts as a splitter cell in the initial model and the forced lesioned model, but not in the optimized lesioned model. Row 2: Cell 246 is a splitter cell only in the optimized lesioned model. Row 3: Cell 711 acts as a splitter cell exclusively in the two lesioned models, not in the initial model. These differences in neural activity underscore the impact of lesioning cells in the model, leading to a reorganization of the model's dynamics. Neurons can adapt by changing their activity patterns, becoming splitter cells, or ceasing to function as splitter cells **A3. 3D PCA of applied to the initial, forced lesioned and optimized lesion models**. All model exhibit a splitting effect at the population level. **B1. Silencing all splitter cells in the model with cues.** All identified splitter cells are silenced, resulting in the lesioned model, which remains capable of performing the task despite the lesioning. During task execution, an entirely new set of splitter cells (SC2) emerges within the model. **B1. Activity of selected SC2 cells across the two models**. Each row corresponds to the same neuron across the two models. All SC2 cells act as splitter cells in the lesioned model but were initially not splitter cells in the initial model. **B3. 3D PCA of applied to the initial and lesioned model**. They both exhibit a splitting effect at the population level.

SC1. Moreover, 297 cells initially classified as splitter cells (SC1) no longer exhibited this property. For instance, in the first row of Figure 3-A2, cell 108 shows a splitting effect in the initial model and in the forced lesioned model but loses this characteristic in the Optimized lesioned model. Among the new splitter cells identified in the SC2 model, two examples are shown in Figure 3-A2, under the "Optimized Lesioned Model" column. The splitting activity of cell #246 and #711 become apparent as the bot traverses the central corridor along different trajectories (red for right, blue for left). In contrast, these neurons lack splitting activity in the initial model, confirming their absence from the SC1 population. The variation in firing activity of the splitter cells in all models demonstrates that silencing SC1 led to a reorganization of the model, giving rise to new splitter cells (SC2) in the *Optimized Lesioned model*, which were not initially splitter cells, while also causing the opposite effect for some previously identified splitter cells, like cell #108. Without optimization, the lesioned model cannot perform the task independently. To address this, we forced the non-optimized lesioned model to execute the task by providing it with the exact sensory inputs the bot would receive if it were correctly performing the alternation task. Interestingly, forcing the model to alternate still resulted in the emergence of 199 splitter cells, as shown in Figure 3, column 'Forced Lesioned model'. Although some neurons from SC2 in the Optimized Lesioned model did not show splitting activity in the Forced Lesioned model, others did, highlighting that the presence of splitter cells does not always overlap across models. The splitting activity in both the Optimized and Forced Lesioned models is also evident at the population level when applying 3D PCA to the internal states of the reservoir, as shown in Figure 3-A3. The neural trajectories in the central corridor do not overlap, illustrating this effect. These results support that as long as the model does the alternation task, splitter cells will emerge, potentially from different neuronal populations. Splitter cells represent a dynamic neural mechanism that is observable during task execution. Performing a specific task activates distinct neural patterns, regardless of the model's optimization state, suggesting that splitter cells arise as a direct consequence of executing the correct sequence of actions.

Model with cues. We identified 280 splitter cells (SC1) in the initial model with cues, all of them were subsequently silenced. Despite this inactivation, the Lesioned model still successfully performed the alternation task and did not need any re-optimization of the leak rates and spectral radius. Remarkably, 373 new splitter cells (SC2), distinct from SC1, emerged. The initial and lesioned models were identical in structure, and no re-optimization was required, unlike the model without cues. This is likely due to the contextual cues informing the model of the correct decision at each time step in the central corridor, making the task less challenging. Even with the removal of splitter cells, the model with cues had sufficient information within its dynamics to continue making the correct decisions. The apparition of new splitter cells after silencing the previous ones strongly reinforces the idea that splitter cells are a direct consequence of task execution. The splitter cells may be induced by the behavior itself.

3.6 Decoding other hippocampal cells

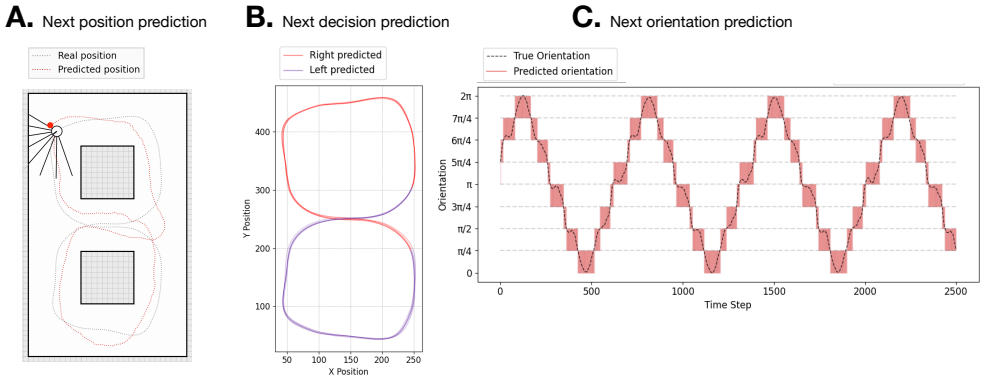

Figure 4. A. Snapshot of bot navigation and position predictions. The grey dotted line indicates the bot's actual trajectory, while the red dotted line represents the predicted position at each time step based on the reservoir state. The large red dot highlights the current prediction. The decoder readout effectively tracks the bot's position within the maze.**B. Decision prediction**. Over 4000 time steps, the reservoir state reveals early internal shifts after the decision point. These shifts enable predictions—within just a few dozen time steps—that the bot will switch direction at the next decision point.**C. Orientation prediction.** The bot's orientation is monitored over approximately 4000 time steps (black dotted line). The reservoir state enables highly accurate predictions of the bot's next orientation, shown in red.

Three readouts were added to the reservoir to extract additional spatial information as the bot navigated, referred as decoder readouts.

Place cell-like activities. The decoder readout achieved a success rate of 96%, correctly identifying the bot's location. While navigating, the decoder consistently activated the correct square corresponding to the bot's position, mimicking the behavior of place cells. This result indicates that place cells can be decoded from the reservoir states. Training the decoder readout to directly decode the position of the bot was also successful as depicted in Figure 4-A. These results demonstrate that place-cells-like activity and spatial coordinates of the bot can be decoded from the reservoir thanks to its capacity of implicitly approximate the state-space of the environment.

Decision cell-like activities. The decoder readout achieved a 100% success rate in predicting the bot's next decision. As shown in Figure 4-B, when the bot recently made a right turn, the reservoir adjusted its internal state to favor a left turn within a few dozen time steps. This indicates that simply by observing the reservoir's internal dynamics, it is possible to anticipate the bot's next move. This finding suggests that the internal neurons of the model encode information about future decisions.

Head-Direction cell-like activities. The decoder readout achieved a 98% success rate in predicting the bot's orientation. As shown in Figure 4-C, This result demonstrates that the bot's head direction can be reliably decoded from the reservoir's internal states.

Sensor-Based Decoding. In contrast, applying the same decoding techniques directly to raw sensor inputs failed to predict place cells, head-direction cells, or decision

cells, as expected. This confirms that the reservoir's internal states encode critical information for these cell-like activities, while the ambiguous sensory data alone does not.

4 Discussion

The model we proposed to study is as simple as it can be: a random recurrent network is fed by sensors that allows to compute a direction change. Despite this apparent simplicity, the model is able to solve a continuous and non-trivial alternating navigation task, using ambiguous, limited and egocentric sensory information. Inside that model, we've been able to demonstrate for the existence of splitter cells as described by Duvelle et al. [2023] in their review. By redoing their analyses on the reservoir, we've found activities that characterize splitter cells in rodents and show how they re-emerge following a targeted lesion and retraining. Furthermore, we've also validated the proposed theoretical framework by developing a model that incorporates concepts from both TCM (Temporal Context Model, Howard and Kahana [2002]) and LSI (Latent State Inference, Gershman and Niv [2010]). These concepts, that appear to be two distinct notions, are actually both needed for the interpretation of experimental data on splitter cells. The model we proposed suggests a simple mechanism where the transition from TCM to LSI is driven by contextual cues.

Even though we did not claim any architectural plausibility with the hippocampal formation, the structure of the model is reminiscent of the CA3 structure with its highly recurrent nature, with the notable absence of learning inside the reservoir. In our case, the learning process is a simple linear identification of a random pattern of activity inside the reservoir in order to issue the proper motor command. This process is indeed well aligned with György Buzsáki [2019]'s inside-out view of the brain, where he suggests that brain activity is not generated from scratch during exploration and learning. According to this view, the brain can be considered as a vast reservoir capable of generating an extensive repertoire of neuronal patterns, initially independent of experience. In other words, these patterns have initially no meaning and acquire it through exploration and learning. In this sense, experience is primarily a process of matching preexisting neuronal dynamics to outside events, and learning does not create new brain activity from scratch, but rather selects preexisting pattern of activity responding to the external stimuli. This "inside-out" view of brain function is further reinforced by findings showing that spatial memory cells and their properties are already present when young pups first begin to move [Langston et al., 2010], or even emerge early in postnatal development, before significant spatial exploration takes place [Muessig et al., 2016, Wills et al., 2014]. These studies indicate that the hippocampus is capable of generating a wide range of possible neuronal trajectories even before the organism begins exploring its environment. On this regards, it is interesting to note that the randomness of the network in our model is not strongly constrained such that any matrix with the proper spectral radius will provide a sufficiently rich reservoir of dynamics from which the behavior can be acquired. Furthermore, in case of targeted lesion, it is relatively easy for the model to re-acquire a behavior based on the new dynamic inside the reservoir, leading to the emergence of new splitter cells. The concept of splitter cells bears a striking resemblance to the notion of mixed selectivity [Rigotti et al., 2013], which refers to the property of neurons that respond to multiple, statistically independent variables or task-relevant aspects. This characteristic is a well-established property of reservoir-type networks which emerges due to the random connectivity and dynamic nature of such networks [Dominey, 2024, Enel et al., 2016, Scleidorovich et al., 2022] and is also observed in primate cortical neurons [Rigotti et al., 2010, Tye et al., 2024]. The emergence of new splitter cells when existing ones are

silenced suggests that mixed selectivity is an inherent property of recurrent networks. When cells displaying mixed selectivity are removed, a new network topology is created, inevitably giving rise to mixed selectivity once again. This feature enables the reservoir to represent complex combinations of inputs and internal states, which is crucial for processing temporal information and solving complexe cognitive tasks.

Beyond splitter cells, we've also decoded the presence of place cell-like, head direction cell-like and decision cell-like activity units. Even though we did not look for each and every other type of cells, we're confident that we would probably be able to decode them. This strongly suggests that their activities do not derive from structure nor learning (we have no learning, and no feedback inside the reservoir) but are rather indicative of a correlation with some pre-existing dynamics that are linked to the actual behavior. Considering for example decision cells (indicating decision to go left or right), it comes as no surprise that we can identify them inside the reservoir since the agent is really going left or right at the end of the central corridor, at least for an external observer. For the agent however, the sensory world is fully described by a set of eight sensors that do not convey the notions of *left* or *right*. Ultimately, this means that these decision cells may only exist in the eye of the observer. In other words, if a reservoir can solve a task with a minimal set of inputs, any variable that can be decoded from the activity of the reservoir derived from the resolution of the task itself as opposed to the specificity of a structures involved in the resolution.

Finally, during the past decade, the understanding of neural coding has gradually shifted from a population vector sum were a behavioural output is generated when the sum exceed a threshold, to a more complex theory that involve multidimentional neural activity space [Churchland and Shenoy, 2024]. This theoretical framework implies several subspaces: a null space, that refers to the subspace of neural activity that does not directly affect certain output variables, and a task-relevant subspace. In this context, splitter cells (and any other task related cells) reflect only a small partial dimension of the global encoding of the neuronal population. Disturbing the whole multidimentional space by lesioning/silenting specific neurons induces a general reorganization of the neuronal activity that affects the neural response of the remaining neurons to the task, hence "creating" new splitter cells, place cells, etc. This reorganization may, with some training, allows the network to adapt and to reproduce the same behavioural output as before. If we demonstrated how this can be achieved using supervised offline and online training inside the model, the question of continuous and gradual adaptation remains open.

5 Tables

5.1 Model parameters when using offline linear regression learning

These parameters represent a set of parameters that are making the models work, however these values can change and vary since the model is robust in changing the hyperparameter. This is particularly the case for the model with cues. In this case, the leak rate can vary between 0.08 to 1, the spectral radius can vary between 0.99 to 1.4.

Table 1. Initial model (offline learning)

Parameter	No cues	With cues
Input size	8	10
Number of units	1000	1000
Input connectivity	0.2	0.1
Reservoir connectivity	0.1	0.05
Input scaling	1	1
Spectral Radius	0.99	0.95
Leak Rate	0.15	0.4
Regularization	1e-5	2e-4

Table 2. Optimized Lesioned model

Parameter	No cues	With cues
Input size	8	10
Lesioned units	50	933
Number of units	950	67
Input connectivity	0.2	0.2
Reservoir connectivity	0.1	0.1
Input scaling	1	1
Spectral Radius	0.99	1.4
Leak Rate	0.08	0.09
Regularization	1e-5	1e-3

5.2 Model parameters when using online RLS learning

Table 3. Initial model (online learning)

Parameter	No cues	With cues
Input size	8	10
Number of units	1000	1000
Input connectivity	0.2	0.2
Reservoir connectivity	0.1	0.1
Input scaling	1	1
Spectral Radius	0.8	0.99
Leak Rate	0.1	0.1

6 Code availability

The code associated with this work is publicly available on GitHub at https://github.com/naomichx/splitter_cells_in_reservoir

References

Eric Aislan Antonelo and Benjamin Schrauwen. On learning navigation behaviors for small mobile robots with reservoir computing architectures. *IEEE Transactions on Neural Networks and Learning Systems*, 26(4):763–780, Apr 2015. ISSN 2162-2388. doi: 10.1109/tnnls.2014.2323247. URL `http://dx.doi.org/10.1109/TNNLS.2014.2323247`.

Eric Antonelo and Benjamin Schrauwen. Learning slow features with reservoir computing for biologically-inspired robot localization. *Neural Networks*, 25:178–190, Jan 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2011.08.004. URL http://dx.doi.org/10.1016/j.neunet.2011.08.004.

Eric A Antonelo and Benjamin Schrauwen. Unsupervised learning in reservoir computing: Modeling hippocampal place cells for small mobile robots. In *Artificial Neural Networks–ICANN 2009: 19th International Conference, Limassol, Cyprus, September 14-17, 2009, Proceedings, Part I 19*, pages 747–756. Springer, 2009.

Eric A Antonelo, Benjamin Schrauwen, and Jan Van Campenhout. Generative modeling of autonomous robots and their environments using reservoir computing. *Neural Processing Letters*, 26:233–249, 2007.

Eric A Antonelo, Benjamin Schrauwen, and Dirk Stroobandt. Event detection and localization for small mobile robots using reservoir computing. *Neural Networks*, 21 (6):862–871, 2008.

Valentino Braitenberg. *Vehicles: Experiments in synthetic psychology*. MIT press, 1986.

Signe Brinkløv, M Brock Fenton, and John M Ratcliffe. Echolocation in oilbirds and swiftlets. *Frontiers in physiology*, 4:123, 2013.

György Buzsáki and Edvard I Moser. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience*, 16(2):130–138, 2013.

Mark M. Churchland and Krishna V. Shenoy. Preparatory activity and the expansive null-space. *Nature Reviews Neuroscience*, 25(4):213–236, March 2024. ISSN 1471-0048. doi: 10.1038/s41583-024-00796-z. URL http://dx.doi.org/10.1038/s41583-024-00796-z.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20: 273–297, 1995.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

Peter Ford Dominey. A connectivity gradient in structured reservoir computing predicts a hierarchy for mixed selectivity in human cortex. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

Éléonore Duvelle, Roddy M Grieves, and Matthijs AA Van der Meer. Temporal context and latent state inference in the hippocampal splitter signal. *ELife*, 12:e82357, 2023.

Pierre Enel, Emmanuel Procyk, René Quilodran, and Peter Ford Dominey. Reservoir computing properties of neural dynamics in prefrontal cortex. *PLoS computational biology*, 12(6):e1004967, 2016.

WE Evans. Echolocation by marine delphinids and one species of fresh-water dolphin. *The Journal of the Acoustical Society of America*, 54(1):191–199, 1973.

Loren M Frank, Emery N Brown, and Matthew Wilson. Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron*, 27(1):169–178, 2000.

Dileep George, Rajeev V Rikhye, Nishad Gothoskar, J Swaroop Guntupalli, Antoine Dedieu, and Miguel Lázaro-Gredilla. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications*, 12(1):2392, 2021.

Samuel J Gershman and Yael Niv. Learning latent structure: carving nature at its joints. *Current Opinion in Neurobiology*, 20(2):251–256, April 2010. ISSN 0959-4388. doi: 10.1016/j.conb.2010.02.008. URL `http://dx.doi.org/10.1016/j.conb.2010.02.008`.

Matthew D Golub, Patrick T Sadtler, Emily R Oby, Kristin M Quick, Stephen I Ryu, Elizabeth C Tyler-Kabara, Aaron P Batista, Steven M Chase, and Byron M Yu. Learning by neural reassociation. *Nature neuroscience*, 21(4):607–616, 2018.

MD György Buzsáki. *The brain from inside out*. Oxford University Press, 2019.

Tom Hartley, Colin Lever, Neil Burgess, and John O'Keefe. Space in the brain: how the hippocampal formation supports spatial cognition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1635):20120510, 2014.

David Heeger et al. Poisson model of spike generation. *Handout, University of Standford*, 5(1-13):76, 2000.

Marc W. Howard and Michael J. Kahana. A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3):269–299, June 2002. ISSN 0022-2496. doi: 10.1006/jmps.2001.1388. URL `http://dx.doi.org/10.1006/jmps.2001.1388`.

Herbert Jaeger. 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, 2001.

Herbert Jaeger. Echo state network. *scholarpedia*, 2(9):2330, 2007.

Nathaniel R Kinsky, William Mau, David W Sullivan, Samuel J Levy, Evan A Ruesch, and Michael E Hasselmo. Trajectory-modulated hippocampal neurons persist throughout memory-guided navigation. *Nature communications*, 11(1):2443, 2020.

Rosamund F Langston, James A Ainge, Jonathan J Couey, Cathrin B Canto, Tale L Bjerknes, Menno P Witter, Edvard I Moser, and May-Britt Moser. Development of the spatial representation system in the rat. *Science*, 328(5985):1576–1580, 2010.

Mantas Lukoševičius. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 659–686. Springer, 2012.

Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer science review*, 3(3):127–149, 2009.

EA Maguire, Richard SJ Frackowiak, and Christopher D Frith. Learning to find your way: a role for the human hippocampal formation. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 263(1377):1745–1750, 1996.

Edvard I Moser, May-Britt Moser, and Bruce L McNaughton. Spatial representation in the hippocampal formation: a history. *Nature neuroscience*, 20(11):1448–1464, 2017.

L Muessig, J Hauser, TJ Wills, and F Cacucci. Place cell networks in pre-weanling rats show associative memory properties from the onset of exploratory behavior. *Cerebral Cortex*, 26(8):3627–3636, 2016.

Razvan Pascanu and Herbert Jaeger. A neurodynamical model for working memory. *Neural networks*, 24(2):199–207, 2011.

Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Guangyao Zhou, Carter Wendelken, Miguel Lázaro-Gredilla, and Dileep George. Space is a latent sequence: A theory of the hippocampus. *Science Advances*, 10(31):eadm8470, 2024.

Mattia Rigotti, Daniel Ben Dayan Rubin, Xiao-Jing Wang, and Stefano Fusi. Internal representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Frontiers in computational neuroscience*, 4:24, 2010.

Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590, 2013.

Honi Sanders, Matthew A Wilson, and Samuel J Gershman. Hippocampal remapping as hidden state inference. *Elife*, 9:e51140, 2020.

Pablo Scleidorovich, Alfredo Weitzenfeld, Jean-Marc Fellous, and Peter Ford Dominey. Integration of velocity-dependent spatio-temporal structure of place cell activation during navigation in a reservoir model of prefrontal cortex. *Biological Cybernetics*, 116(5):585–610, 2022.

Kimberly L Stachenfeld, Matthew M Botvinick, and Samuel J Gershman. The hippocampus as a predictive map. *Nature neuroscience*, 20(11):1643–1653, 2017.

Nathan Trouvain, Luca Pedrelli, Thanh Trung Dinh, and Xavier Hinaut. ReservoirPy: An efficient and user-friendly library to design echo state networks. In *Artificial Neural Networks and Machine Learning – ICANN 2020*, pages 494–505. Springer International Publishing, 2020. doi: 10.1007/978-3-030-61616-8_40. URL `https://doi.org/10.1007/978-3-030-61616-8_40`.

Kay M Tye, Earl K Miller, Felix H Taschbach, Marcus K Benna, Mattia Rigotti, and Stefano Fusi. Mixed selectivity: Cellular computations for complexity. *Neuron*, 2024.

Jeffrey J Wenstrup and Christine V Portfors. Neural processing of target distance by echolocating bats: functional roles of the auditory midbrain. *Neuroscience & Biobehavioral Reviews*, 35(10):2073–2083, 2011.

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5):1249–1263, 2020.

Thomas J Wills, Laurenz Muessig, and Francesca Cacucci. The development of spatial behaviour and the hippocampal neural representation of space. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1635):20130409, 2014.

Emma R Wood, Paul A Dudchenko, R Jonathan Robitsek, and Howard Eichenbaum. Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron*, 27(3):623–633, 2000.