PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Model-based spatial navigation in the hippocampus-ventral striatum circuit: A computational analysis

Ivilin Peev Stoianov[1], Cyriel M. A. Pennartz[2], Carien S. Lansink[2], Giovani Pezzulo[1]*

1 Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy, **2** University of Amsterdam, Swammerdam Institute for Life Sciences–Center for Neuroscience Amsterdam, The Netherlands

* giovanni.pezzulo@istc.cnr.it

OPEN ACCESS

Citation: Stoianov IP, Pennartz CMA, Lansink CS, Pezzulo G (2018) Model-based spatial navigation in the hippocampus-ventral striatum circuit: A computational analysis. PLoS Comput Biol 14(9): e1006316. https://doi.org/10.1371/journal.pcbi.1006316

Editor: Jean Daunizeau, Brain and Spine Institute (ICM), FRANCE

Received: October 27, 2017

Accepted: June 20, 2018

Published: September 17, 2018

Data Availability Statement: Custom software implementing the computational model described in the study is fully available (in a self-maintained repository) at https://github.com/stoianov/MBRL.

Funding: This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 785907 (Human Brain Project SGA2 to GP and CP), the European Commission's Seventh Framework Programme (PIEF-GA- 2013-622882 to IS), and the Human

Abstract

While the neurobiology of simple and habitual choices is relatively well known, our current understanding of goal-directed choices and planning in the brain is still limited. Theoretical work suggests that goal-directed computations can be productively associated to model-based (reinforcement learning) computations, yet a detailed mapping between computational processes and neuronal circuits remains to be fully established. Here we report a computational analysis that aligns Bayesian nonparametrics and model-based reinforcement learning (MB-RL) to the functioning of the hippocampus (HC) and the ventral striatum (vStr)–a neuronal circuit that increasingly recognized to be an appropriate model system to understand goal-directed (spatial) decisions and planning mechanisms in the brain. We test the MB-RL agent in a contextual conditioning task that depends on intact hippocampus and ventral striatal (shell) function and show that it solves the task while showing key behavioral and neuronal signatures of the HC—vStr circuit. Our simulations also explore the benefits of biological forms of look-ahead prediction (forward sweeps) during both learning and control. This article thus contributes to fill the gap between our current understanding of computational algorithms and biological realizations of (model-based) reinforcement learning.

Author summary

Computational reinforcement learning theories have contributed to advance our understanding of how the brain implements decisions—and especially simple and habitual choices. However, our current understanding of the neural and computational principles of complex and flexible (goal-directed) choices is comparatively less advanced. Here we design and test a novel (model-based) reinforcement learning model, and align its learning and control mechanisms to the functioning of the neural circuit formed by the hippocampus and the ventral striatum in rodents—which is key to goal-directed spatial cognition. In a series of simulations, we show that our model-based reinforcement learning agent replicates multi-level constraints (behavioral, neural, systems) emerged from rodent cue- and context- conditioning studies, thus contributing to establish a map between computational and neuronal mechanisms of goal-directed spatial cognition.

Frontier Science Program (grant no. RGY0088/2014 to GP). The GEFORCE Titan GPU card used for this research was donated by the NVIDIA Corp. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

The neurobiology of goal-directed decisions and planning in the brain is still incompletely known. From a theoretical perspective, goal-directed systems have been often associated model-based reinforcement learning (MB-RL) computations [1,2]; yet, a detailed mapping between specific components (or computations) of MB-RL controllers and their brain equivalents remains to be established. Much work has focused on brain implementations of single aspects of MB-RL controllers, such as action-outcome predictions or model-based prediction errors [3,4]. A more challenging task consists in mapping MB-RL computations to a systems-level neuronal circuit that provides a complete solution to decision and control problems in dynamic environments–or in other words, identifying the biological implementation of a complete MB-RL agent rather than only one or more components.

The neuronal circuit formed by the (rodent) hippocampus and ventral striatum circuit is particularly appealing, and can be productively taken as a "model system" to understand biological implementations of model-based computations during spatial navigation (see Fig 1A). The hippocampus (HC) has long been implied in place-based and goal-directed navigation [5]. Recent findings suggest that the role of hippocampus in goal-directed navigation may be mediated by the strong projections from the hippocampal CA1 and subicular areas to the ventral striatum (vStr) [6], which might convey spatial-contextual information and permit forming place-reward associations [7–10]. From a computational perspective, the hippocampus and ventral striatum may jointly implement a *model-based controller* for goal-directed choice [8,11–17]. In this scheme, HC and vStr might be mapped to the two essential components of a model-based reinforcement learning (MB-RL) controller [1,18]: the *state-transition model*, which is essentially a model of the task that permits to predict the next location given the current state (say, a given place) and chosen action, and the *state-value model*, which encodes the (expected) reward associated to each state, respectively.

Different from a model-free controller, a model-based controller can use an explicit form of look-ahead prediction (or internal simulation) that permits to imagine future trajectories and covertly evaluate them [1]. Similar look-ahead predictions have been reported in the HC: at difficult decision points, such as when they are at the junction of a T-maze, rodents sometimes stop and produce *internally generated sequences* of neuronal activity in the HC that resemble the sequential neuronal activity observed in the same area when they navigate through the left or right branches of the T-maze [7,20]. These internally generated sequences may serve to serially simulate future spatial trajectories (e.g., a trajectory to the left and successively a trajectory to the right). In turn, these look-ahead predictions might elicit covert reward expectations in the vStr [9]. By linking spatial locations with reward information, the HC-vStr might thus jointly implement a model-based mechanism that allows an animal to covertly simulate and evaluate spatial trajectories [13,21], using a serial scheme that has some analogies with machine learning algorithms (e.g., forward simulations in Bayes nets [22,23] or Monte Carlo rollouts in decision trees [24]). Internally generated sequences were also reported during sleep or rest periods in the hippocampus and associated structures such as the ventral striatum [25]; in this case, they have been associated with multiple functions such as planning [26] and the off-line "replay" of past trajectories for memory consolidation [27,28]–a mechanism that has inspired early (DYNA [29]) as well as more recent ("experience replay" [30]) machine learning schemes.

Here we present a biologically-grounded computational analysis of model-based RL, by comparing side-by-side the behavior and neuronal activity of living organisms (rodents) and of a probabilistic MB-RL agent, in *cue-* and *context- conditioning* tasks that depend on intact hippocampal and ventral striatal (shell) function [19,31,32]. The results we report indicate that

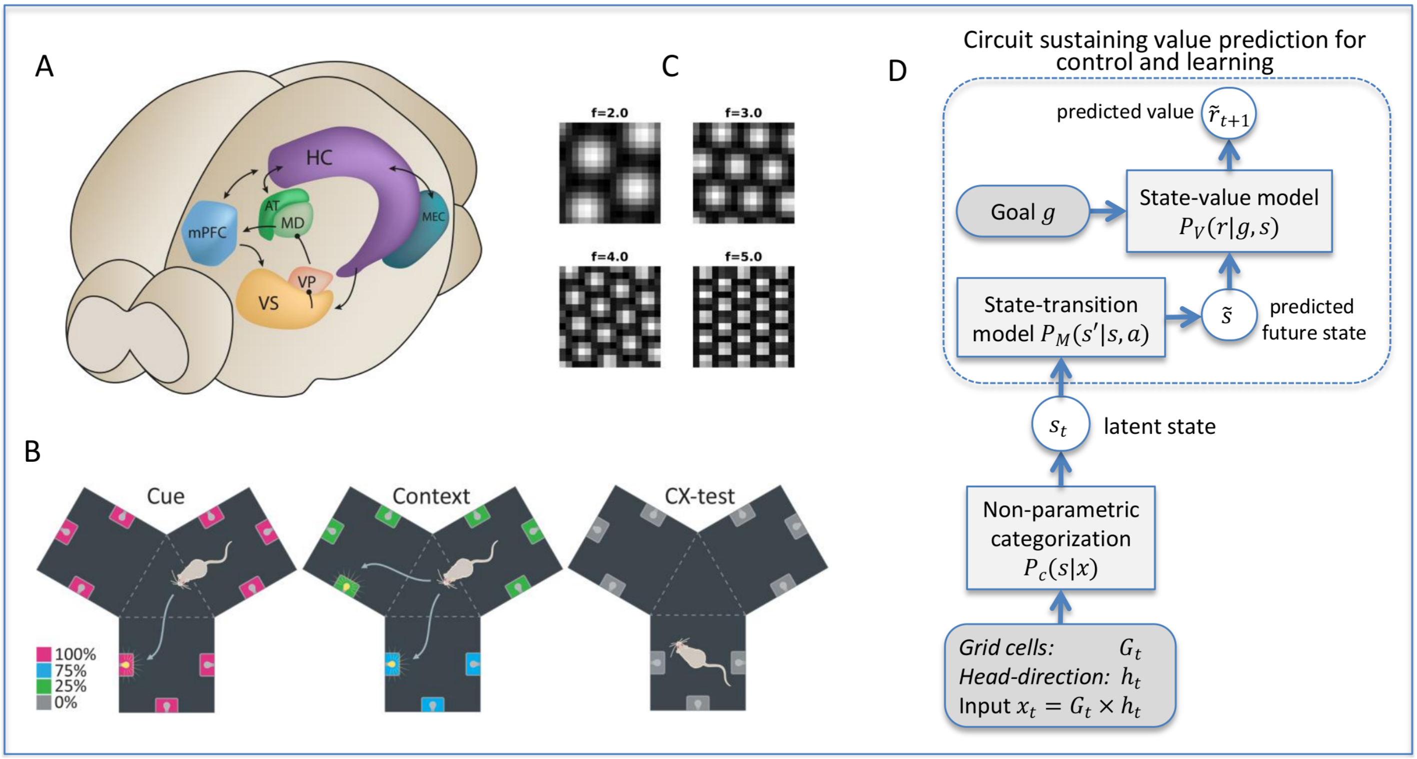
A
HC
AT
MD
mPFC
MEC
VP
VS
C
f=2.0
f=3.0
f=4.0
f=5.0
D
Circuit sustaining value prediction for control and learning
predicted value $\tilde{r}_{t+1}$
Goal g
State-value model $P_V(r|g,s)$
State-transition model $P_M(s'|s,a)$
$\tilde{s}$ predicted future state
s_t latent state
Non-parametric categorization $P_c(s|x)$
Grid cells: G_t
Head-direction: h_t
Input $x_t = G_t \times h_t$
B
Cue
Context
CX-test
🟥 100%
🟦 75%
🟩 25%
⬜ 0%

Fig 1. Spatial navigation in rodents: functional organization, scenario, and overall architecture of the model. (A) Structures in the rodent brain that are involved in goal-directed navigation. HC-VS constitute the essential structures of the putative model-based control system, which supports goal-directed behavior. AT and MEC provide input to the model-based control system. Output of the HC-VS circuitry may reach the cortex/mPFC via VP and MD. (B) The Y-maze used here and in [19]. Each room contains 3 goal locations, which can be cued with a light with different probability in three different phases (separate panels) according to the color legend. See the main text for more explanation. (C) Sample grid cells providing systematic (spatial) information about the environment. (D) Architecture of our biologically inspired model-based reinforcement-learning model for spatial navigation that combines non-parametric clustering of the input signal $P_c(s|x)$, a state-transition model $P_M(s'|s,a)$, and a state-value model $P_V(r|g,s)$ with lookahead prediction mechanism for learning and control. See the Methods for details. Abbreviations: AT, anterior thalamus; HC, hippocampus; MD, mediodorsal thalamus; MEC, medial entorhinal cortex; mPFC, medial prefrontal cortex, VP, ventral pallidum, VS, ventral striatum.

https://doi.org/10.1371/journal.pcbi.1006316.g001

the MB-RL agent replicates the multi-level constraints (behavioral, neural, systems) emerged from rodent cue- and context- conditioning studies. First, we show that the MB-RL agent correctly learns to obtain reward in multiple goal locations and develops contextual preferences that are analogous to those of rodents. Second, we show that latent states emerging in the *state-transition* and *state-value* models of the MB-RL agent show key coding properties of HC and vStr neurons, respectively. Third, by comparing multiple variants of the MB-RL algorithm, we show that forward (look-ahead) predictions—of the kind that can be associated to hippocampal forward sweeps [20]—improve both action selection and learning. This latter result is important as the computational efficacy of model-based schemes based on forward sweeps has been put into question [33].

Taken together, the results of this study speak to the possibility of aligning the MB-RL scheme to the HC—vStr circuit to reproduce both behavioral and neuronal data. Establishing this kind of mappings is particularly important to foster cross-fertilizations between reinforcement learning and neurophysiology; for example, to identify specific algorithmic solutions that the brain uses to face problems that are still challenging in machine learning, or conversely, to advance novel computationally-guided theories of neural processing [10,34].

Finally, our MB-RL scheme has two main features that go beyond the state of the art from a computational perspective, and which are necessary for an adaptive agent to deal with openended situations. First, the MB-RL agent successfully faces the challenge to learn simultaneously three components: (the latent categories that form) the state space, the state-transition and the state-value models. Our novel scheme that combines non-parametric and reinforcement learning ensures that only the latent categories that afford accurate state-transition and state-value functions are retained, thus linking inextricably the three learning processes. Second, the MB-RL agent can deal with multiple goals in a native way, without the necessity of re-learning.

Results

We report the results of a series of simulations, which aim to assess whether the novel MB-RL agent introduced here replicates the multi-level constraints (behavioral, neural, systems) that emerge from a context conditioning task, in which neural ensembles in rat hippocampus and ventral striatum were simultaneously recorded [19].

Experimental set-up: Y-maze

The maze used in the animal study and in our simulations is a y-shaped symmetric arena consisting of 3 identical square chambers rotated 120 degrees from each other and connected through a central triangular passage, see Fig 1B. Each chamber contained three goal locations located along the chamber walls, where reward was (probabilistically) delivered. Each reward location had a cue light above it.

Our simulation follows the protocol used in the animal study [19] and consists of three phases. In the first phase (*Cue*: *cue conditioning*), the correct goal location was cued with a light; reward was delivered if the animal reached the goal location and made a nose-poke (but note that we do not simulate the nose poke here). In the second phase (*Context*: *contextual conditioning*), the correct goal location was cued with a light, too; but the probability of reward (following a nose poke) depended on the chamber’s spatial position: south room goals brought reward in 75% of the trials, whereas the other goals brought 25% reward. Finally, a *context-conditioning test* (*CX test*) was performed, with no cues or rewards (i.e., "free run"), to probe the animals’ acquired contextual place preferences.

The key behavioral result of the animal study, which we aim to reproduce in the MB-RL agent model, is that the contextual preference for the south room goals is preserved in the last (CX test) phase. We also aim to test whether the internal representations acquired by the two key components (*state-transition* and *state-value* model) of the MB-RL agent during learning, have coding properties that resemble HC and vStr activations, respectively, in the animal study.

Brief introduction to the computational model

The computational model (MB-RL) is fully explained in the Methods section; however, we shortly summarize it here for the sake of the reader. The model essentially includes three interconnected components: a latent state categorization mechanism, which permits to learn the state representations that are useful to solve a task (e.g., place cells); a state-transition model, which learns the contingencies between actions and future states (e.g., what location do I reach by going left?) and a state-value model, which learns the utility of the states (e.g., is reaching location x good or bad?).

There are three important aspects of the model to note (see the Methods section for further details). First, we assume that state-transition and state-value models correspond to HC and vStr, respectively, and they jointly permit to steer look-ahead predictions (analogous to hippocampal forward sweeps [20]). As we will discuss below, our simulations will compare different versions of the same MB-RL agent, which use, or not use, look-ahead predictions for learning, control (action selection) or both.

Second, the MB-RL agent includes a mechanism that adaptively selects whether or not to do a sweep, and the depth of the sweep, depending on the agent's uncertainty and confidence about the choice. Essentially, the depth of sweeps decreases when the agent becomes sufficiently confident about its choice. However, a change in reward contingencies (e.g., the location of reward) would produce a reversal of this effect—and the generation of longer sweeps. This is because after failing to reach a reward at the expected location, the agent would become again uncertain about its choice, hence triggering sweeps to minimize it before a choice [23]. Here we are interested in validating the computational efficiency of this mechanism, which may potentially explain why VTE behavior increases rapidly when animals make errors following a switch in reward contingency [20].

Third, most model-based planning algorithms start from a state space that is predefined by the programmers. However, the brain has to simultaneously learn internal models to generate predictions, and the states to be used by the model. In keeping, in the MB-RL agent, the learning processes required to acquire latent states and the (state transition and state value) models that use the latent states are interdependent (see Eq 1 below). Here we are interested in validating this learning scheme, which permits learning latent states that not only entail high perceptual accuracy but also afford good prediction and control when used by the state-transition and state-value models.

Behavioral analysis: Accuracy

Successful context conditioning was evaluated in [19] with a *CX test*: a "free-run" target preference behavioral test in which the rats freely explored the maze for several minutes without any cue or reward. During this period, to probe the animals' learned preferences, statistics were collected about their visitations of (and "nose pokes" in) rooms that in the previous *Contextual Conditioning* phase were associated with high- and low-reward probabilities. Analogously, to test the MB-RL agent's learned preferences, we ran a free-run simulated test for 2,000 time steps, in which the agent started from a small central area (3x3 units) and with random orientation. In this simulation, action selection maximized the likelihood for (expected) reward pooled from all the 9 goal locations; and the agent did not learn. To better clarify the behavioral effects of conditioning in the MB-RL agent, we performed the same free-run test (and without learning) also at the end of cue conditioning. Finally, and importantly, we envisaged to compare different versions of the same MB-RL agent that uses, or not uses, look-ahead predictive mechanisms (analogous to hippocampal forward sweeps [20]) for both control / action selection and (reward) learning—in order to assess the computational importance of these mechanisms. Specifically, we compared four different versions of the MB-RL agent using a 2x2 design, which considers 2 controllers (i.e., using or not using forward sweeps) and 2 learning procedures (using or not using forward sweeps); see Section 4 for details.

We expected that look-ahead prediction (or forward sweeps) to provide the agent significant advantages for both control / action selection and (reward) learning; this result would be particularly intriguing for action selection, given that the baseline controller uses the full factorized distribution of reward and is thus computationally demanding. The results shown in Fig 2 confirm our hypotheses, by showing an additive advantage of forward sweeps in both action selection and learning. As shown in Fig 2A, the MB-RL agent that uses forward sweeps in both action selection and learning (swControl+Reward), with average success of $\mu = 69\%$ ($\sigma = 2$) during the entire *Cue Conditioning* phase outperforms in amount of collected reward the agent that uses forward sweeps only for control (swControl, $\mu = 62\%$ ($\sigma = 3$), $t(1,9) = 6.9$,

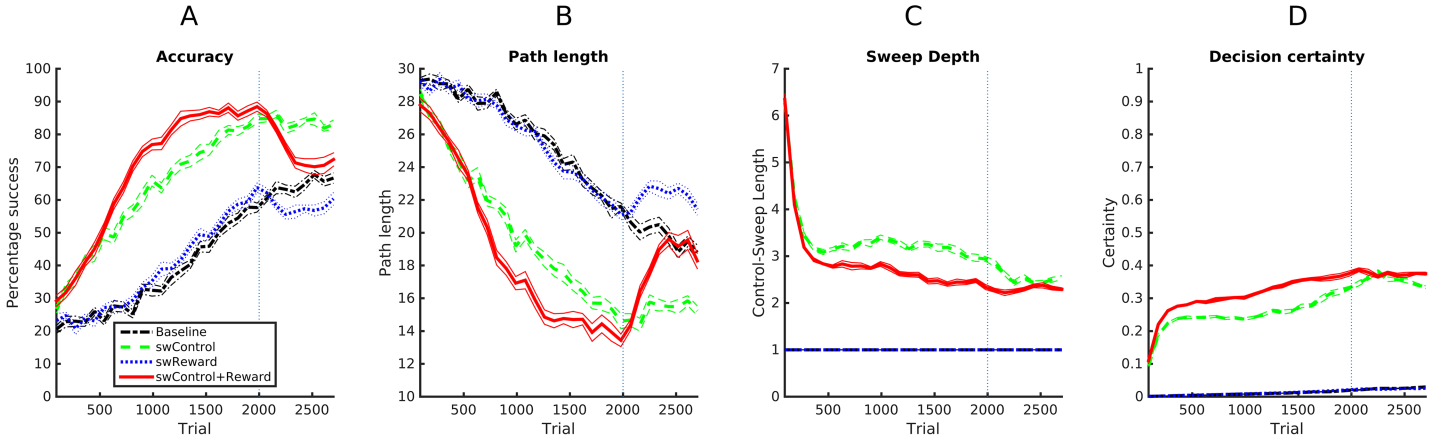

Fig 2. Benefits of forward sweeps for action selection and control. (A,B) Learning performance (accuracy and length of the agent path to the goal). (C) Length of sweeps used for control / action selection. (D) Decision (un)certainty during control / action selection.

https://doi.org/10.1371/journal.pcbi.1006316.g002

n = 10, p<0.001), the agent that uses forward sweeps only for learning (swReward, μ = 39% (σ = 2), t(1,9) = 33.4, n = 10, p<0.001), and the agent that lacks them during both control and learning (Baseline, μ = 36% (σ = 1), t(1,9) = 39.6, n = 10, p<0.001) and this advantage is evident from the very beginning of learning. For all the four agents, performance decreases after the vertical dotted bar, in correspondence of the *Contextual Conditioning* phase—but this is expected, given that less reward is available in this phase.

Fig 2B illustrates the same advantage in overall path length needed to reach the reward site of the four agents. Along with the increased performance, the length of sweeps used for control by the swControl+Reward and swControl agents decreases with time (Fig 2C). This is because the agent's decision certainty progressively increases (Fig 2D) as their state-value model becomes more effective; in turn, the mechanism for sweep length control explained in the Methods Section decreases length (we obtained equivalent accuracy and path length results using a controller having a constant length of 9 and thus requiring more computational results; results not shown). This result aligns well with evidence that hippocampal forward sweeps at decision points progressively diminish during learning [20]—until they disappear when the animal develops a habit (which we do not model here, but see [13]).

Behavioral analysis: Preferences after conditioning

As the above analysis confirmed the advantages of the MB-RL agent that uses forward sweeps in both action selection and learning (swControl+Reward), we used this MB-RL agent for the rest of our behavioral and neural analyses.

Our target rodent study reported that during "free run" in the *CX test* phase, animals showed an increased preference for the room that yielded more rewards in the previous (*Context*) phase [19]. The MB-RL agent correctly reproduces this pattern of behavioral results. As shown on Fig 3A, in the *CX test* phase (in the absence of reward), the agent shows a significant behavioral preference (number of reward-site visits) for the room that was previously most rewarded (room1 vs. room3, t(1,9) = 4.3, n = 10, p = 0.002; room2 vs. room3, t(1,9) = 7.6, n = 10, p<0.001), which matches the animal's learned preference (and differed from an analogue test executed right after cue conditioning: room1 vs. room3, t(1,9) = 1.5, n = 10, p = 0.16; room2 vs.–room3, t(1,9) = 0.1, n = 10, p = 0.92). Fig 3B provides three representative trajectories before (blue trajectories) and after (red trajectories) context conditioning–the latter

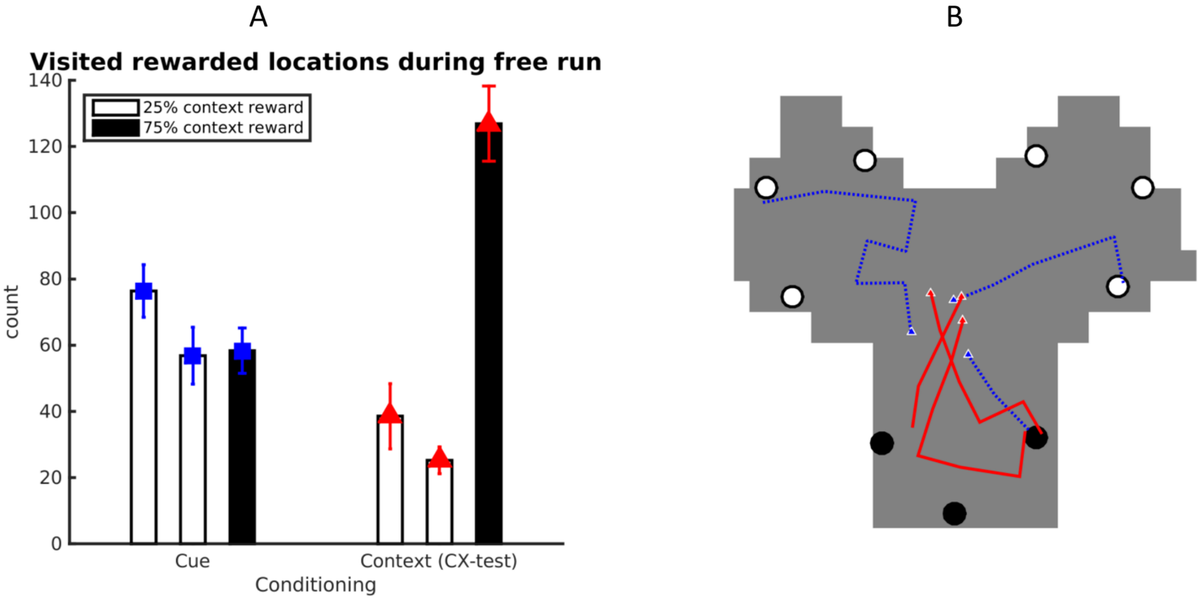

Fig 3. Behavioral results of the simulations. (A) Preferences of the MB-RL agent after *Cue Conditioning* and after Contextual conditioning (i.e., *CX test*). In the CX test, both agent and animals show a preference to visit targets in the room that was previously most rewarded. (B) Sample trajectories of the agents during these tests (after *Cue Conditioning*, blue; after Context Conditioning, or *CX test*, red).

https://doi.org/10.1371/journal.pcbi.1006316.g003

exemplifying the increased preference for the room that included highly rewarded locations in the Context phase (black dots).

Neural-level analysis of conditioning

To further establish a parallel between the MB-RL agent and the HC—vStr circuit, we analyzed the *content* of the latent states that emerged in the agent's *transition model* $P_M(s'|s,a)$ and *value function* $P_V(r|g,s)$ at the end of the *Cue* and *Context Conditioning* phases. We decoded the spatial location corresponding to each latent state and created: (a) value maps for each target showing for each spatial location the greatest reward across all head-directions and (b) transition maps showing how position could change after applying an(y) action at any orientation. We also analyzed the effect of conditioning at the neural level: how the learned internal models change and how this change affects behavior.

The neuronal analysis of HC and vStr cells after the cue conditioning task reported different coding characteristics in the two areas, with the former showing high spatial selectivity and the latter showing selectivity for reward- and task-related information, and place/reward combinations [19]. The two components of the MB-RL agent acquire analogous coding preferences after conditioning. Fig 4 shows the transition probabilities learned by the state-state (transition) model after *Cue Conditioning* (Fig 4A) and how they changed after *Context Conditioning* (Fig 4B). In Fig 4, each green square corresponds to a place in the maze (i.e., the central square corresponds to the central place of the maze) and the colors of the smaller squares code the greatest probability of transitioning from the location of that state, regardless the direction, to the location of any other state of the maze (shown are only the possible successors, i.e., nearby locations), by executing any of the 3 actions available to the agent (see Methods section). The more red the color of the cell, the higher the probability. In other words, this map shows the spatial projection of the learned "successor states" (small squares) of each latent state (green squares) implicitly coded in the agent's probabilistic transition model—analogous to

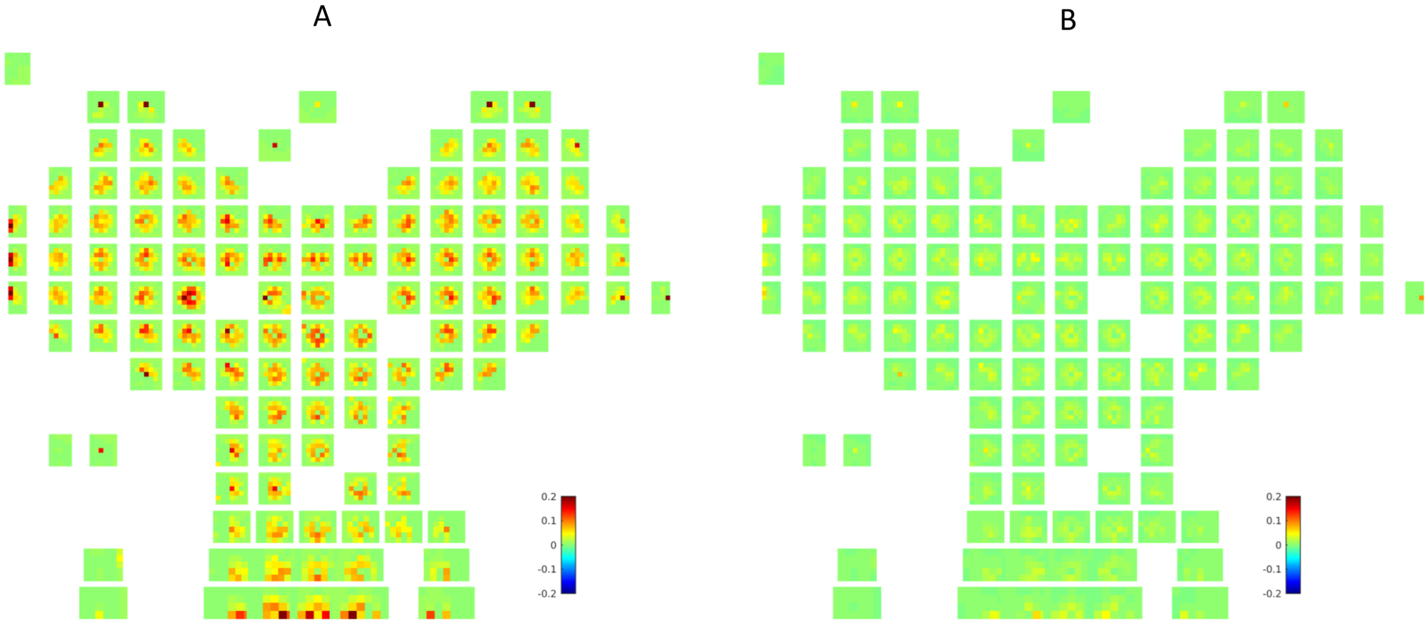

Fig 4. Neural representation of state transitions in the state-state model. Latent states developed by the state-transition models averaged across all 10 learners after the *Cue Conditioning* phase (A); and the changes due to *Contextual Conditioning*, i.e., the differences between probabilities before and after *Contextual Conditioning* (B). Each image from the transition model $P_M(s'|s,a)$ encodes the greatest likelihoods $P_M(xy(s')|xy(s), a)$ across all head-directions and actions to step from the location $xy(s)$ of a given latent state (place) s to nearby places $xy(s')$ located within the range of an(y) action from the location of the current place, following any of the available actions, i.e., the (probabilistic) location of the successors of every state. The locations $xy(s)$ and $xy(s')$ of s and s' are decoded using an inverse of the function providing input to the Dirichlet model. Note that, as expected, the decoding procedure is not perfect—hence the gaps in the maps.

https://doi.org/10.1371/journal.pcbi.1006316.g004

"successor representations" [35,36]. For example, the green squares near the center include transitions to all directions while the green squares on the borders only include transitions towards the center. In other words, the learned transition probabilities encode the actual transitions available in the environment, reflecting the idea of HC encoding a navigational "cognitive map" [5]. Furthermore, importantly for our analysis, these codes only change (remap) to a minor extent between the *Cue* and *Context Conditioning* phases (Fig 4B)—in keeping to what was reported empirically [19]. Essentially, the changes reflect a consolidation of the same transition model, not a remapping, due to additional training during the *Context conditioning* phase.

Fig 5 shows the value of each place (i.e., the learned probability of obtaining reward in the place) when a goal location is selected or cued, as learned by the state-value function component of MB-RL agent after *Cue Conditioning* (Fig 5A) and how these values changed after *Context Conditioning* (Fig 5B). The nine outer big Y-shapes represent the possible goal sites; inside each of them, the colors code the probabilities of obtaining reward while starting from any specific location in the maze—the darker red the color, the higher the probability. The central Y-shape represents the combined value function across all goal sites, which is effective when the goal is not cued, as in the behavioral CX test. These probabilities relate well with key coding properties of vStr neurons in [19]. If one assumes that the current goal is unknown, as in the CX-test, one can see the same rotational invariance in the model (Fig 5A, the central Y-shape) as in sample vStr neurons (Fig 5C; for more samples, see [19]). This aligns well with the idea that vStr neurons might encode place-reward associations and these in turn can be conditioned on a specific goal, possibly provided by prefrontal areas [37] or vicariously from hippocampal forward sweeps [21,38].

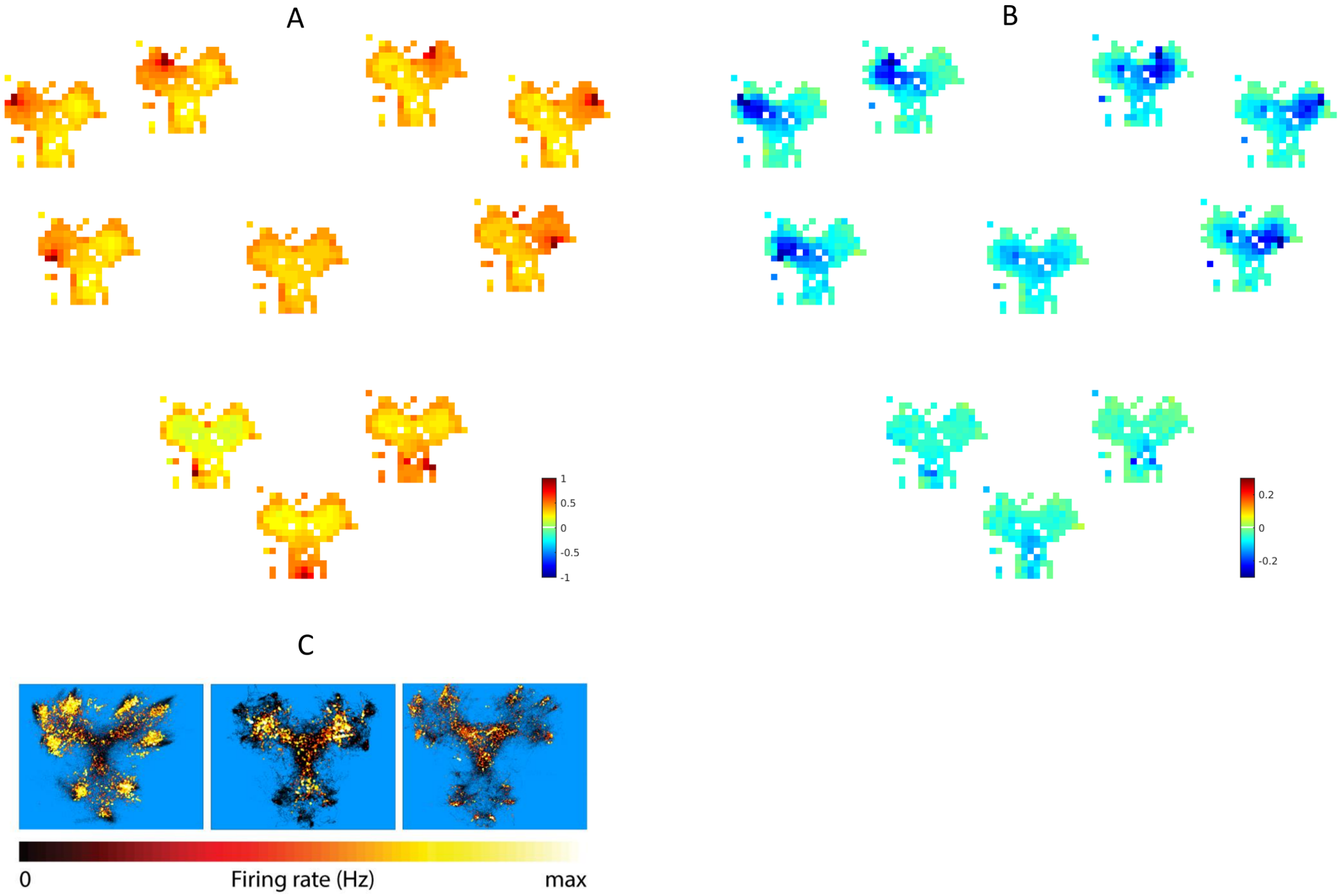

Fig 5. Neural representation of value in the state-value model. Latent states developed by the state-value model averaged across all 10 learners, after the *Cue Conditioning* phase (A); and how they change after *Contextual Conditioning*, i.e., the differences between probabilities before and after *Contextual Conditioning* (B); and the activity of three sample vStr neurons drawn from the experiment in [19] (C). The images in (A, B) represent the greatest value $P_V(r = 1|g,xy(s))$ across all head-directions attributed to a given spatial position $xy(s)$ for a given target g. Each image represents one target and its location in the plot corresponds to its location in the Y-maze. The central image represents the combined value function across all targets. It is rotationally symmetric after the rotationally symmetric reward delivered during Cue Conditioning (as some of the vStr rodent-neurons; see insets C) and becomes asymmetric during the context conditioning phase. The spatial positions $xy(s)$ are decoded from the latent states s using an inverse of the function providing input from the grid cells.

https://doi.org/10.1371/journal.pcbi.1006316.g005

Note that the probabilities emerging in the state-value model are closely related to reward functions of RL, which can be used to learn a policy from the current state [1] and, in a model-based scheme, to retrieve covert expectations of reward using mental simulation or forward sweeps [13,21,23]. As usual in RL, the value of reward places is temporally discounted such that places farther from goal sites have lower values. This creates a sort of gradient or "tessellation" of the task, which might reflect not only proximal distance but also other information such as phases or subtasks that are necessary to secure a reward [19].

Importantly, the coding of value in the state-value model changes drastically after *Contextual Conditioning* (Fig 5B). While after Cue Conditioning one can observe full rotational symmetry, after the Context Conditioning a marked preference is evident for the goal locations in the most rewarded (lower; south) room—which also explains the behavioral results of Fig 2. This change of preference can be better appreciated if one considers the distribution of the change of values associated with latent states in the state-value model after *Contextual*

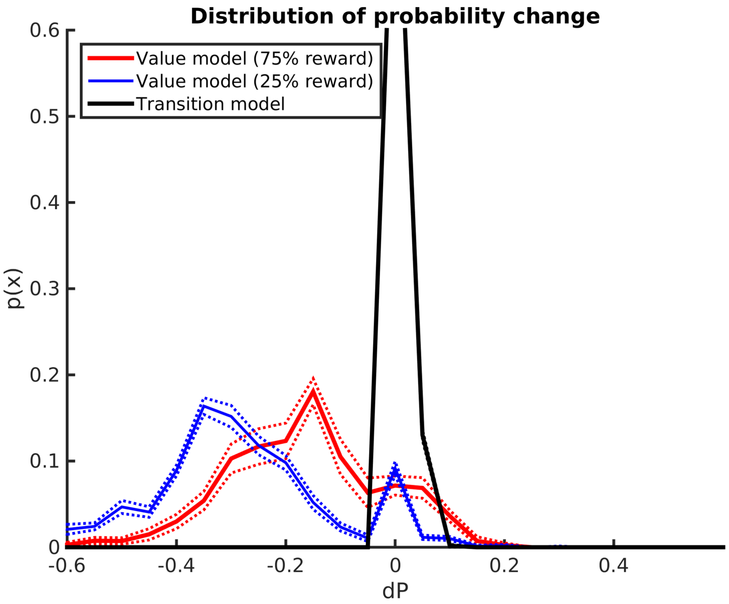

Fig 6. Changes in the state-state and state-value models after contextual conditioning. This figure shows what changes the Contextual Conditioning procedure produces in the probability values of states in the state-state or transition model (black), and in the state-value model, where states correspond to the most rewarded (red) or less rewarded (blue) rooms. For clarity, we only show the changes of states having a probability that is greater than 0.05 (for the state-state model) and 0.5 (for the state-value function). This choice of thresholds in motivated by the fact that in the state-value function we are interested in verifying changes in the states carrying significant value information (e.g., those regarding the goal states or their neighbor's), not in the many states that have a low probability value in all situations (see Fig 5).

https://doi.org/10.1371/journal.pcbi.1006316.g006

Conditioning (Fig 6). Specifically, there is a decrease of all the state values (given that reward is less frequently available), but the value of states corresponding to the less rewarded rooms (blue, mean change of -0.29 across all learners) decreases significantly more than those of the more rewarded rooms (red, mean change of -0.17 across all learners; t(1,9) = 25.0, p<0.001). This result is coherent with a body of literature implying vStr in the coding of reward expectancies [21] and would imply a sensitivity for changing reward contingencies (see Discussion). This is in sharp contrast with the states in the state-state transition model (black), which essentially does not change (mean change of 0.01 across all learners).

Finally, Fig 7 provides analysis of the sweeps for action selection generated during the two conditioning phases. During *Cue Conditioning*, sweep length increases at decision points, e.g., the center of the maze and nearby densely located targets (Fig 7A). This simulates the rodent data that consistently show longer internally generated sequences at branch points [15], [23]. In turn, Fig 7B shows how sweep length changes after *Context Conditioning*. Notably, when the agent is in a location that is far away from targets with low reward probability, it needs longer sweeps to accumulate sufficient evidence about the most valuable action. Instead, sweep length remains the same or decreases for highly rewarded targets.

Discussion

Computational methods of reinforcement learning have been widely used to analyze neuronal data [2,39–41]. However, this has been done far more systematically for model-free RL–mostly associated to habitual behavior—than for model-based RL methods [22,23,42]–more associated to goal-directed behavior and planning. A challenge consists in mapping the components of a MB-RL agent to a neural circuit that offers a complete, systems-level solution to goal-

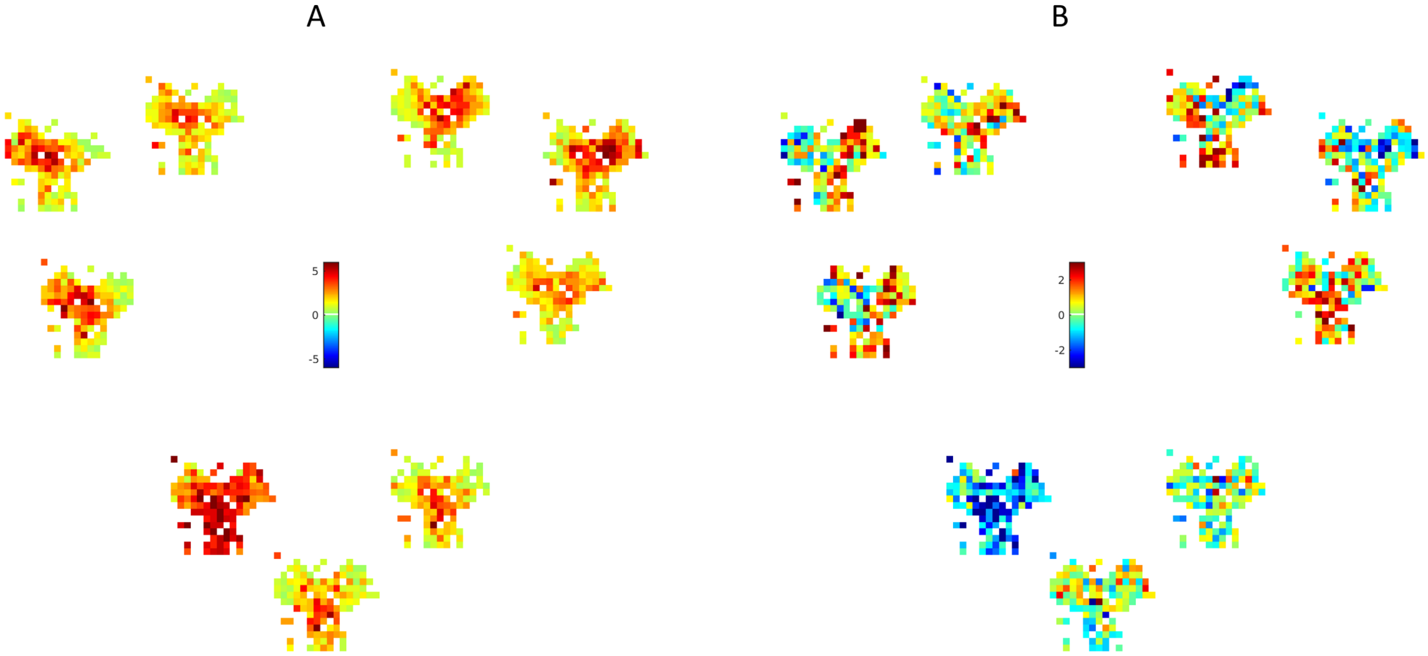

Fig 7. Analysis of sweeps. Length of sweeps during control / action selection, for each target (separate Y-maze images) and each spatial location (dots in the Y-mazes). Sweep length is color-coded. (A) Sweep length after *Cue Conditioning*. (B). Change of sweep length after *Context Conditioning*.

https://doi.org/10.1371/journal.pcbi.1006316.g007

directed choice, rather than a single aspect of it (e.g., the dynamics of dopaminergic neurons and their relations to reward prediction error).

Rodent research suggests that the neuronal circuit formed by the hippocampus (HC) and ventral striatum (vStr) might implement model-based computations in spatial navigation. In this scheme, the HC learns a state-transition model (permitting to predict future states) and the vStr learns a state-value model (encoding goal-dependent state values). The HC might then predict future trajectories (e.g., using forward sweeps and ripple-associated replay) and vStr might covertly evaluate them, permitting to select the best (that is, reward-maximizing) course of action [12,13,21]. To provide a computationally-guided analysis of this idea, here we designed a Bayesian model-based reinforcement learning (MB-RL) agent and tested it in a challenging contextual conditioning experiment that is dependent on hippocampal and ventral striatal function [19].

The results of our simulation show that the MB-RL agent procedure can reproduce animal data at multiple—behavioral, neural, systems—levels. At the behavioral level, the agent shows place preferences that are analogous to rodents [19]. Key to this result is the fact that the agent's value function is conditioned on goal information. Here, in other words, goal information acts as a "context" and permits the agent to learn multiple goals, as required by the experimental set-up [19]. This possibility is precluded to other RL controllers that are only optimized for a single value function (or goal) and thus lack context sensitivity [1] but is important to scale up RL methods to real-world situations that involve multiple potential goals. At the neural and systems levels, the agent develops internal codes that reproduce key signature of hippocampal and ventral striatal neurons [19]. This is important to establish deep relations between formal methods and neurophysiology that potentially span multiple levels of analysis—from computational to algorithmic and implementational [43]. Finally, in our proposed model, hippocampal sweeps can contribute to both action selection and learning. Our comparison of multiple MB-RL schemes shows that mechanisms of look-ahead prediction that resemble hippocampal forward sweeps improve both control and learning. There has been some recent skepticism around the idea of using look-ahead prediction for control and learning, based on the fact that prediction errors tend to accumulate over time [33], which would cast into doubt the idea that hippocampal sequences may serve predictive or planning roles. Furthermore, alternative proposals have cast model-based control (and hippocampal function) in terms of backward [42] or a combination of forward and backward inference [22]. Our results suggest that the limited form of look-ahead prediction that we adopt is computationally advantageous, thus lending computational-level support for theories that assign forward sweeps a predictive role [44,45].

The MB-RL agent is related to other computational models of model-based spatial decisions [22,23,46–52] that use various forms of forward planning. Different from these architectures, the MB-RL agent includes a nonparametric module that learns a "state space" in an unsupervised manner while the agent learns the task, and which works synergistically with the other components of the architecture. Indeed, the state space representations emerging from the nonparametric learning procedure ensure that only states that afford good prediction and control are retained. Using end-to-end learning (i.e., from perception to action) helps keeping the various learning procedures (state space, action-state and state-value learning) coordinated and is more effective and biologically realistic than using a staged approach—still popular in RL—in which unsupervised state space learning is seen as a generic preprocessing phase. From a computational perspective, this approach can be considered to be a novel, model-based extension of a family of Bayesian methods that have been successfully applied to decision-making problems [53–57]. It is also important to note that our approach does not require learning a separate state space (or sensory mapping) for each goal; rather, multiple spatial goals share the same state space—which implies that our MB-RL agent can deal natively with multiple goals. This is different from most current deep RL approaches, which require learning a new state space for each problem or task (but see [1]). In our set-up, sharing the state space is effective because all goals correspond to spatial locations—or in other words, they all lie in the same low-dimensional state space that supports spatial navigation. This reliance on a relatively low-dimensional state space makes spatial navigation quite special and tractable, both in machine learning and the brain; it is thus possible that brain areas that support goal-directed processing outside navigation domains rely on more complex computational solutions that remain to be fully established [10,58,59].

At the neurophysiological level, the latent states that emerged in the agent's state-state and state-value models during conditioning share important similarities with HC and vStr coding, respectively—and in particular the selectivity for space vs. reward-predictive information and place/reward combinations, respectively. Our computational model thus provides a detailed (yet to some extent abstract) mapping between specific components of model-based controllers and neuronal circuits. Establishing this sort of mappings is important if one aims to the cross-fertilization of current research in reinforcement learning / artificial intelligence and neurophysiology; for example, to probe in detail neuronal solutions to challenging machine learning problems. Indeed, one should expect that the neuronal networks in the HC and vStr (as well as other brain structures) have been optimized over time to solve the challenging problems that animals must face every day, and cracking this neural code might be helpful to design artificial agents with similar abilities. In this respect, a limitation of the model proposed here concerns the simplifying assumptions on HC and vStr coding and their respective roles in model-based control. For simplicity, we mapped one-to-one HC to state-transition and vStr to state-value models. However, this sharp division is likely to be simplistic and both neural coding and computational roles of the HC-vStr circuit are certainly more complex. HC cells show some sensitivity to goal and reward information [60] and remap from cue-on to cue-off and spontaneous goal site approach [19]. Furthermore, vStr can code information that goes beyond scalar reward expectations; it can code for example item- or object-specific information, such as the expected reward associated to a specific food or other characteristics of a task, including (for example) intermediate steps to goal locations [19]. This suggests a more complex architecture in which HC and vStr do not map one-to-one to components of a model-based controller but may realize model-based computations in a more distributed manner; and in which they form a latent task model that encodes task variables that are more abstract than spatial codes, permitting "tessellating" tasks into behaviorally meaningful events (e.g., task phases that lend to reward delivery) and forming hierarchies. Understanding how the brain tessellates and organizes the state space hierarchically may be particularly important for the success of future model-based reinforcement learning algorithms. Current model-based RL methods suffer from the problem that, during long look-ahead searches, errors sum up; for this, in practice, most practical applications use model-free solutions. However, in principle, endowing model-based methods with the ability to exploit relevant task structure (e.g., milestones or subgoals) may mitigate this problem, by making forward search more abstract and hierarchical (or "saltatory") [61–63] or by avoiding chaining too many consecutive predictions [33]. To this aim, unsupervised state representation methods (including deep learning methods) that are able to identify latent task structure may be productively incorporated into model-based control systems—yet it remains to be studied in future research how to handle or bound (hierarchical) state representations that may solve realistic problems [36,64–68]. An alternative possibility consists in learning a more complex state space (successor representations) that afford implicit prediction [35,69]. Finally, it is important to remark that overall brain architecture for model-based spatial navigation plausibly includes other areas in addition to HC and vStr. For example, in rodents prelimbic cortex may be particularly important to decide whether or not to engage the model-based system (or to initiate a deliberative event [20,23]) and to code for spatial goals [37]. More broadly, the hippocampus cross-talks extensively with cortical areas, especially during sleep—and this cross talk can be essential to train not only the cortex as usually assumed [70] but also the hippocampus [14,15,71]. Realizing a complete, systems-level architecture for goal-directed spatial cognition remains an open challenge for future research.

In sum, our results highlight the viability of model-based methods to study goal-directed spatial navigation, in the same way model-free methods have been widely adopted to study habitual action control [22,23,42]. Our comparison of alternative MB-RL schemes shows that forward sweeps are useful for both learning and control / action selection. Indeed, our results show that the MB-RL agent that includes both kinds of forward sweeps is the most accurate and the one with lowest uncertainty. This result, again, connects well to neurophysiological evidence that implicates (prospective) internally generated hippocampal sequences to both learning / memory function [27,72] and decision [20,26]. Our approach has a number of implications from both computational and neurophysiological perspectives, which we address next.

From a computational perspective, it is worth noting that the results that we present here may be obtained by a model-free RL system, augmented in various ways (e.g., by using various goals as part of the state classification system). Two points are however important, the former related to coding parsimony, and the latter related to biological plausibility. First, separating the transition function from the value function (which is typical of model-based systems) makes the realization of multiple goals more "economic" from a coding perspective. In the classical RL approach, a Q-value function encodes the expected cumulative future reward for each state and action, but for one single goal [73]. The addition of multiple goals in the Q-value function would increase learning time dramatically. In contrast, adding novel goals to our model-based RL approach requires just learning the new value associated to each state.

Second, and importantly, a pure model-free RL approach would fail in navigation tasks that are typically associated with the hippocampus, such as detour tasks [74]. This is because model-free RL methods lack the required flexibility to rapidly adapt to novel contingencies (e.g., a change of the reward or goal location, or of the state transitions, as in the case of detours).

Another interesting approach to understand predictive aspects of hippocampal coding is in terms of successor representations (SR) [35,69,75,76]—or prospective codes at the single cell level. Our approach is distinct from the successor representation (SR) approach [35,69,75,76], for four main reasons. Firstly, the two approaches target two distinct (possibly complementary) forms of predictive coding in the hippocampus. Our approach addresses predictive coding at the level of sequential activation of multiple cells, e.g., theta sequences and forward sweeps at decision points [13,20]. Rather, the SR addresses predictive coding at the level of single cells; and in particular the backward expansion of place fields (in CA1), which can be considered a form of predictive representation of future states or locations [69]. Secondly, the two approaches use two distinct computational schemes (model-based versus model-free or hybrid) to engage in predictive processing. In our model-based approach, predictions about future locations (and their value) are generated by engaging the state-transition and state-value models on-line, to perform forward sequential planning and policy selection. This requires learning a one-step probability transition model $P_M(s'|s,a)$ and a probabilistic cumulative value function $P_V(r|g,s)$. Rather, the SR approach does not require engaging an internal model for forward prediction during planning (although some variants of SR learn a model and engage it off-line, to "train" SR [29,76]). This is because a SR essentially caches a series of predictions (e.g., of future occupancies) into a single state representation. More formally, the SR approach learns the future occupancy $M_s(s')$ function, or the probability to occupy any state s' following a specific policy (standard, on-policy method) or any policy (extended, off-policy method [76]) starting from state s. Using this function, the SR approach can be sensitive to future events without engaging a model on-line. Thirdly, our approach and the SR have distinct trade-offs. Our model-based approach has the highest flexibility (because all knowledge embedded in the model is used on line) but also the highest computational cost. The usefulness of SR rests on the possibility to decompose the cumulative value function into a product of SR and local reward. The SR approach permits using predictive representations in a model-free manner, thus skipping (costly) model-based computations during planning and choice. Extensions of the SR approach permit to have roughly the same flexibility as model-based approaches, in challenging situations like detour and revaluation tasks [76]. At the same time, a prediction generated using a SR is usually less specific than a model-based prediction, as the SR marginalizes over all possible sequences of actions. Finally, and most importantly here, the two approaches would assign different roles to internally generated hippocampal sequences. Our model-based approach uses internally generated hippocampal sequences for both learning and on-line decision-making and planning. Rather, in the SR approach internally generated hippocampal sequences are not required for on-line decision-making or learning (although some variants of SR use sequences for off-line training, i.e., experience replay [29,76]). In most navigation scenarios, decisions can be done using a single SR (comparable to a single place cell or a small population of place cells [69]), hence this approach would not explain per se why the hippocampus should encode theta sequences, after sufficient learning. Of course, the two forms of prediction entailed by our model-based and the SR approach are not mutually exclusive, but can be productively combined; for example, by using sequences of SR (rather than "standard" place cells) within a model-based scheme. The efficiency and biological plausibility of such combined scheme remains to be tested.

At the neurophysiological level, the proposal advanced here is that the hippocampus encodes a model or cognitive map of spatial transitions and uses this model for state estimation (self localization) and forward inference (prediction / planning). In our scheme, these computations are intimately related. The same model can support different functions (e.g., self localization, forward or even retrospective inference), depending on the specific "message passing": state estimation may rely on the cross-talk between input structures of the hippocampus that encode the observations or inputs of the model (putatively, LEC and MEC) and the hippocampus proper (putatively, dentate gyrus and CA3-CA1) [22]; forward prediction is more dependent on the hippocampus proper (recurrent dynamics in CA3, and CA3- CA1 connections) [13,14,22]; whereas a complete goal-directed decision requires the interplay of the hippocampus with other brain areas that may putatively encode goal states (mPFC [77,78]) and/or state values (vStr [19,25]). The complete systems-level circuit supporting goal-directed decisions has not been systematically mapped. However, various studies point to the importance of dorsolateral and ventromedial prefrontal cortex, and orbitofrontal cortex in supporting model-based computations [3,79] (but see [80] for evidence that orbitofrontal cortex may participate in post-decision processes rather than model-based decision).

Our computational scheme is quintessentially model-based; however, it permits to dynamically modulate the degree of model-basedness and its temporal horizon. In this scheme, there is no fundamental difference between theta sequences observed during "standard" navigation, and forward sweeps that occur at choice points, and which stretch much farther in space [7]. These are part and parcel of the same model-based (theta-paced) inferential mechanism that runs continuously during navigation (or outside navigation, during sequential processing [81]). Whether or not a more intensive, far-reaching sweep occurs depends on a trade-off between its costs (e.g., costs of computation and the time consumed by it) and benefits (e.g., whether or not engaging in far-looking prediction is useful for action selection). From a computational perspective, one can characterize the benefits of a far-reaching sweep by considering the (epistemic) value of information that it may make accessible [23,52]. This would explain why sweeps predominantly occur at difficult choice points [7], and why repeated exposure to the same contingencies lowers choice uncertainty, thus rendering deep search unnecessary. In keeping, our information-theoretic approach automatically determines the utility of performing a forward sweep and sets its depth, by considering the initial uncertainty and the value of the to-be-acquired information [23,82]. Neurophysiologically, the decision of whether or not to engage in deep search has been hypothesized to involve the prelimbic area (in rodents) [20].

The current MB-RL implementation has a number of limitations, which we briefly summarize here. First, the neurophysiological implementation of internally generated neuronal sequences is more sophisticated than our simplified implementation; for example, there are at least two classes of internally generated sequential activity patterns, termed theta sequences (and sometimes forward sweeps) [7,38,83] and sharp wave ripple sequences (also called replay) [84], which have different neurophysiological signatures and possibly different (albeit coordinated) roles; see [13] for a review. Replay activity in the hippocampus has already inspired various methods for improving learning (e.g., experience replay [30]) or hybridizing model-free and model-based computations (e.g., DYNA [29]); understanding neuronal sequential activity in more detail might permit going beyond these metaphorical mappings and potentially design improved algorithms. A second, related limitation of the current model is that only focuses on forward replays and does not take into account backward replay, which may have a separate computational role. Both forward and backward replays are observed at rest periods (e.g., during sleep), suggesting that they are useful for consolidating the internal model [14]. However, recent evidence suggests that reverse replays are more prominent during the awake state, after (novel) reward observations [85,86]. This makes sense if one considers that backward replays may help learning from recent rewards, possibly using some sort of "eligibility trace" [87] to update the current model or the current policy. More broadly, one may conceptualize replays in terms of *epistemic actions* that aim at gathering (the best) evidence to improve the internal model [52]. In this perspective, it becomes plausible that different kinds of memory contents need to be accessed during sleep, before a choice and after obtaining a reward; see [88] for a recent computational characterization of hippocampal replay in terms of prioritized memory access. Endowing the current model with the ability to "direct" (forward or backward) replays to informative memory content—possibly using a mechanism that marks memories or places with saliency information [89]—is an open research objective. Third, since head direction input is used to classify latent states, the resulting place fields are directional. While place cells often have some directionality in their fields, our model does not account for non-directional or omnidirectional aspects of the fields. This also implies that in order to select an action, the MB-RL agent needs to know its orientation (using e.g., head direction cells). This design choice was made for the sake of simplicity and has no major implications for the phenomena we were interested in; however, extending the model to obtaining omnidirectional place fields (e.g., as shown in Fig 4, in which place fields are averaged across all directions) would help reproducing more accurately neurophysiological data. Furthermore, our model treats the (hippocampal) state space as flat. An interesting alternative possibility that deserves future investigation is that the hippocampal code is hierarchically organized along a septo-temporal axis, with more temporal cells that encode information that are broader in space (e.g., place and grid cells having larger firing fields [90–92]). Hierarchical organization of the state space can be productively investigated in our framework by rendering the Bayesian nonparametric approach hierarchical, or manipulating its concentration (α) parameter. A final limitation of the current model is that it uses simplified grid cells. Several modeling approaches have been proposed that explain grid cells in terms of (for example) attractors [93,94], oscillators [95] or an eigendecomposition of state space [69], which may be exploited to develop more realistic grid cells within our proposed model.

Methods

MB-RL agent: Sensors and action primitives

The MB-RL agent is characterized by position and orientation information, provided as real values, relative to a coordinate system with a unit step size. The y-maze is centered on the origin of the agent's coordinate system and the overall diameter of the maze is about 16 arbitrary units. The agent could move with constant speed using three action primitives: (i) step 1.5 units forward, (ii) turn 90 degrees to the left and make a 1.5-units step, and (iii) turn 90 degrees to the right and make a 1.5-units step. The effective turn and step size were noisy values (Gaussian noise, $\sigma = 0.1$). At each discrete time moment, the agent selects one of the three actions and moves; after each action, it obtains sensory information and (sometimes) reward.

The MB-RL agent receives sensory information from two sensors: a head-direction (i.e., orientation) sensor h and (a simplified model of) grid cells G; the agent has no further position or proximity sensors. In the rat, head-direction cells located in the postsubiculum part of the HC and in the anterior thalamic nuclei discharge as a function of the horizontal orientation of the head, also in darkness, within a preferred sector extending about 90 degrees that persists even for weeks [96] and are hypothesized to drive place fields [97]. In keeping, our orientation sensory unit h provides a discrete, 4-level signal that reduces the true orientation to four non-overlapping 90-degree sectors.

In turn, information about the spatial regularities derives by the so-called grid cells located in the medial entorhinal cortex (MEC) as reviewed in [98]. The discharge pattern of each grid cell resembles a hexagonal grid characterized by spatial frequency (spatial range from, e.g., 0.3 to 3 m), phase, and orientation (see Fig 1C). Several models propose detailed explanations of how this pattern could be produced, e.g., oscillatory interference [95] and attractor networks [93]. Here we used a simplified mathematical model that constructs the typical hexagonal pattern of the grid cells by summing three 2D sinusoidal gratings oriented $\pi/3$ apart [99]: $\mathrm{G}(\mathrm{x},\mathrm{y}) = \frac{2}{3}\left(\frac{1}{3}\sum_{\mathrm{i}=1}^{3}\cos(\mathrm{k}_{\mathrm{i}}(\mathrm{r}-\mathrm{r}_0)) + \frac{1}{2}\right)$, where $\mathrm{k}_{\mathrm{i}\,=\,1\ldots3}$ stand for 2D oriented ramps with spatial phase $r_0 = [x_0,y_0]$. We equipped our model with 11 grid cells with spatial frequencies ranging from 2 to 7 cycles per maze (step 0.5) and randomly drawn grid orientation and phase (samples in Fig 1C). Each grid cell provides a binary signal sampled on a unit-step grid. The binary signal from all grid cells was combined multiplicatively to provide a signal *G* with about 500 different levels.

Finally, in order to simplify the recognition of target goal sites, we endowed the agent with an implicit target-recognition mechanism, which provides target identity (but not position, which has to be inferred on the basis of the above sensory information). Upon approaching the target within a unit distance, the MB-RL agent receives reward with a given probability, which depends on the experimental phase (see below).

MB-RL agent: Architecture

The synthetic agent was implemented as a (Bayesian) model-based reinforcement learning (MB-RL) controller, having three key components (Fig 1D). This and the following figures illustrate the model using the formalism of Bayesian networks, which show the model variables (circles) and the conditional distributions (rectangles) linking them.

Latent state categorization mechanism. The latent state categorization mechanism learns to categorize the combined head-direction and grid-cell input signal $x = G{\times}h$ into latent states *s* containing implicit spatial information. This component abstracts the acquisition of place cells [5], which form the model's emergent (latent) state space on top of which state-transition and state-value models are learned. The non-parametric categorization of the internal input signal is implemented using a growing probability distribution $P_c(s|x)$ whereby each novel input signal extends the distribution with a new conditional entry. The new entry is initialized according to a Dirichlet process mixture, also known as the Chinese restaurant process (CRP) [100], with a prior that accounts for the popularity of the categories developed thus far. Critically, this categorization is further updated to account for behaviorally relevant transition contingencies (see later).

State-transition model. The state-transition model learns the effects of executing a given action primitive *a* in the emergent latent state-space domain: $\mathrm{s} \times \mathrm{a} \rightarrow \mathrm{s}'$ where s' is the expected latent state after the transition. This model is intended to abstract the functions of the hippocampus. It is implemented as a conditional multinomial probability distribution $\mathrm{P_M(s'|s,a)} \sim \mathrm{Cat}(\theta)$ parametrized by a Dirichlet conjugate prior $\theta_{\mathrm{s,a}}(\mathrm{s}') \sim \mathrm{Dir}(\alpha)$ that keeps a count of the number of time each state *s′* is reached when the agent started from state *s* and selected action *a*.

State-value model. The state-value model learns the value *r* of each latent state *s* given a goal or target *g*. This model is intended to abstract the functions of the ventral striatum. The *state-value model* defining the reinforcement contingencies is implemented as a conditional binomial probability distribution of reinforcement $\mathrm{P_v(r|g,s)} \sim \mathrm{B}(\varphi)$ (rewarded: $r = 1$, not rewarded: $r = 0$) parameterized by a Beta-conjugate prior $\varphi_{\mathrm{g,s}}(\mathrm{r}) \sim \mathrm{Beta}(\beta)$. For each combination of *g* and *s*, the model (through its conjugate prior) accounts for the number of successes (reward) and failures (no reward) during the entire learning experience (see later for details). Note that at difference with most RL models, the value of states depends on the current goal g. This is in keeping with evidence that vStr encodes item-specific values, not just scalar values [8].

MB-RL agent: Control mechanism

We tested the behavior of the simulated agent in conditions that mimic the experimental manipulations of the animal study [19]. During the *cue conditioning* (*Cue*) and *contextual conditioning* (*Context*) phases, action selection (i.e., the Bayesian inference of the next action) was conditioned on a specific cued target g; this corresponds to the fact that, in the animal study, the correct goal location was cued with a light. Rather, during *the context conditioning test* (*CX test*) phase, action selection operated in a "free-running mode" in which the target was not externally cued and the animal was able to navigate and search for a reward freely.

To select an action, the control mechanism first reads the combined input signal x_t from the head-direction sensor and the grid cells and uses the categorization distribution to select the corresponding most likely latent category, or state $s_t = argmax_s\ P_c(s|x_t)$. Then, assuming a Markov Decision Process, inferential action selection maximizes the likelihood to obtain reward for a given target or goal: $a_t = argmax_a P(r = 1|g,s_t,a)$. Assuming further that the value depends only on the landing state following the selected action, we can factorize the value distribution: $\mathrm{P(r|g,s,a)} = \Sigma_{\mathrm{s}'}\ \mathrm{P_v(r|g,s')\ P_M(s'|s,a)}$. Thus, the control mechanism could use the state-transition model $P_M(s'|s,a)$ to simulate action execution, and the state-value model $P_V(r = 1|g, s')$ to evaluate the (immediate) expected outcomes under the various possible actions. Note that in practice, evaluating all the latent states would require high computational costs, especially for a large latent state-space–and such exhaustive evaluation would consider subsequent states that are highly unlikely, e.g., two states that lie in different corners of a maze. A more effective approach would consist in using an approximation that only evaluates the state $\tilde{s}$ that most likely would be visited upon executing a given action $a:\ \tilde{\mathrm{s}} = \mathrm{argmax}_{\mathrm{s}'}\mathrm{P_M}(\mathrm{s}'|\mathrm{s_t}, \mathrm{a})$; and this approach could generalize to simulated further transitions for deeper value prediction. In the simulations below, we compare the behavior of two control mechanisms: a controller that uses a form of lookahead prediction (forward sweeps) and one that dispenses from using it (baseline).

Baseline (Shallow) controller

The baseline controller shown in Fig 8 selects the action a^i that maximizes the immediately obtainable reward, i.e., $a_t = argmax_{a^i} P(r = 1|g, s_t, a^i)$ locally predicted with the help of the state-transition and state-value models: $\tilde{\mathrm{s}}^{\mathrm{i}} = \mathrm{argmax}_{\mathrm{s}'}\mathrm{P_M}(\mathrm{s}'|\mathrm{s_t}, \mathrm{a^i})$ and $\mathrm{a_t} = \mathrm{argmax_i}\mathrm{P_V}(\mathrm{r} = 1|\mathrm{g}, \tilde{\mathrm{s}}^{\mathrm{i}})$, which we can combine in a single expression: $a_t = argmax_a\ P_V(r = 1|g,argmax_{s'} P_M(s'|s_t,a))$. However, this one-step prediction method is quite myopic as it does not consider future events.

Controller using look-ahead prediction or forward sweeps

We designed an action selection mechanism based on forward sweeps (Fig 9) that generalizes the idea of local value maximization by using a limited form of forward search: it performs one forward sweep for each available action primitive a_i in order to estimate the future value obtainable as a result of applying this action. For each action a^i, the policy first simulates one transition that starts from the current state and applies that action: $\mathrm{s_t} \times \mathrm{a^i} \rightarrow \tilde{\mathrm{s}}^{\mathrm{i}}_1$ where $\tilde{\mathrm{s}}^{\mathrm{i}}_1 = \mathrm{argmax}_{\mathrm{s}'}\mathrm{P_M}(\mathrm{s}'|\mathrm{s_t}, \mathrm{a^i})$. Then it iteratively performs a series of simulated transitions, each one locally maximizing the expected reward without restriction on the action: $\tilde{s}^i_j = \mathrm{argmax}_{\mathrm{s}'}\mathrm{P_v}(r = 1|g, argmax_{s'} P_M(s'|\tilde{s}^i_{j-1}, :))$. Reward

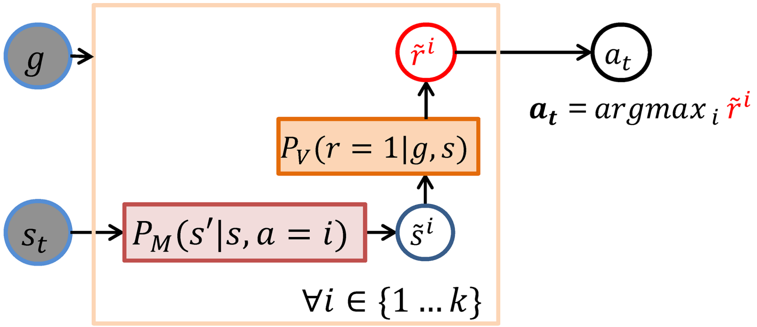

Fig 8. Shallow control mechanism. It exploits the state-transition and state-value models and local maximization to predict the expected value $\tilde{r}^i$ of each action primitive a^i and select the most valuable one. For each action primitive, the mechanism first finds the latent state that most likely would be achieved by applying that action and then finds among those states the predictably most valuable one. The action bringing to that state is the selected one: $a_t = argmax_a$ $P_v(r = 1|g, argmax_{s'} P_M(s'|s_t, a))$.

https://doi.org/10.1371/journal.pcbi.1006316.g008

evidence $\tilde{r}^i$ for each action a^i gradually accumulates along each step of each sweep: $\tilde{\mathrm{r}}^{\mathrm{i}}_{\mathrm{j}} = \sum_{\mathrm{j}=1\dots\mathrm{l}} \mathrm{P}_{\mathrm{v}}(\mathrm{r} = 1|\mathrm{g}, \tilde{\mathrm{s}}^{\mathrm{i}}_{\mathrm{j}})$. The accumulated reward evidence drives stochastic action selection using the *soft-max* function (here, with exponent coefficient β = 80).

We optimized the efficiency of our sweep-based approach by taking into account a cost-benefit trade-off between the availability of more information and the computational costs required to execute long sweeps. From a theoretical perspective, forward sweeps can be stopped when there is sufficient discriminative evidence for action selection choice [23]. To model this, we defined an information-based measure of *decision certainty* that uses the transition- and value- models to decide which action to simulate and how far to deepen the sweeps (Fig 10). First, the value v_s of being in a state s was defined as the negative uncertainty of the probability to obtain reward in that state, i.e., $v_s = \log_2 P_v(r = 1|g,s)$. Then, the decision certainty d_{ij} of taking action a^i (assuming that it brings to state $\tilde{s}^i$) instead of action a^j (expected that it brings to state $\tilde{s}^j$) was defined as the difference $\mathrm{d}_{\mathrm{ij}} = \mathrm{v}_{\tilde{\mathrm{s}}^{\mathrm{i}}} - \mathrm{v}_{\tilde{\mathrm{s}}^{\mathrm{j}}}$. Thus, to optimize sweep length, at each depth we calculated the decision certainty d relative to the two actions with greatest cumulative evidence for reward (see Supporting Information S1 File). Note that the log-

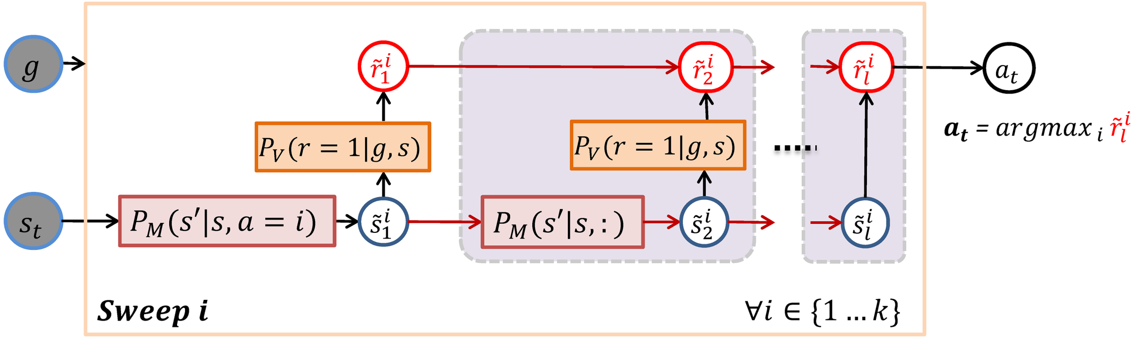

Fig 9. Controller using look-ahead prediction (or forward sweeps). The mechanism includes k sweeps, one for each available action primitive, each consisting of l steps. At each step j, the mechanism first iteratively predicts the next latent state of each sweep i : $\tilde{s}^i_j = argmax_{s'} P_v(r = 1|g, argmax_{s'} P_M(s'|\tilde{s}^i_{j-1}, a))$ and then accumulates the predicted value for that state: $\tilde{r}^i_j = \tilde{r}^i_{j-1} + P_V(r = 1|g, \tilde{s}^i_j)$. The first transition of the i-th sweep departs from the current latent state s_t and applies action primitive a^i while the following transitions recursively depart from the predicted state in the previous step and use any action that maximize the predicted reward. Finally, the mechanism selects the action that maximizes the cumulative predicted value: $a_t = argmax_i\, \tilde{r}^i$.

https://doi.org/10.1371/journal.pcbi.1006316.g009

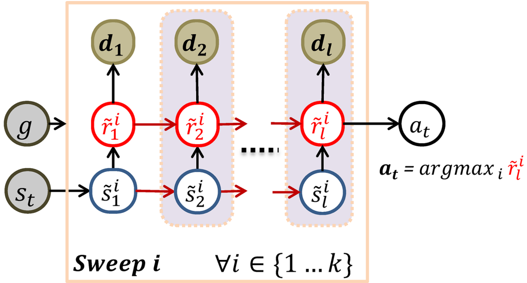

Fig 10. Information-driven adaptive sweep-depth. At each depth j is calculated discriminative certainty $d_j = log_2(\tilde{r}_j^{max1}) - log_2(\tilde{r}_j^{max2})$ for the two currently most valuable sweeps. Sweep depth increases until the selection certainty exceeds a given threshold: $d_j > d_{thr}$.

https://doi.org/10.1371/journal.pcbi.1006316.g010

difference used to calculate this value implicitly normalizes the accumulated evidence into probabilities. The sweeps were stopped when the decision certainty exceeded a threshold (here, 0.15) and the action with the greatest accumulated evidence for reward was selected.

MB-RL agent: Learning procedure

In general, the objective of the MB-RL agent is to obtain maximum reward by moving from a given start-position to a given goal-position, which may vary during the learning process (see later). Following a MB-RL approach, starting from an empty memory, the system has to learn a probabilistic model of the environment (i.e., the transitions from state to state in the maze) and the distribution of reward given the target (or goal), which varies from trial to trial (here, limited to 9 possible goals as in the rodent experiment). Learning thus consists in adjusting the probability distributions of the two (state-state and state-value) agent models on the basis of experience, using Bayes' rule [101]. Importantly, like other deep RL experiments, the model is not provided with a predefined state space, but also has to simultaneously learn to categorize the internal input signal $x = G{\times}h$ into behaviorally useful (spatial) categories s that represent the location of the agent in the maze. In sum, the agent has to simultaneously learn three things: state-state and state-value models, and state space; and learning is end-to-end (i.e., all parameters are trained jointly), see Fig 11.

In our simulations, learning consisted of a sequence of trials, each of which started from a random position and had the goal to reach a randomly selected target within 32 time steps. Upon approaching the target within one unit distance and this time limit, the artificial agent received reward with a certain probability. Upon obtaining reward (or after the time limit had expired), a new trial began. Like our animal study [14], learning was divided into two phases that differed on reward distribution. In the first, *Cue Conditioning* phase that lasted for 2.000 trials, reaching a target within the time limit was rewarded with 100%. The starting position in the first 360 trials of this phase was the central area (within 4 units from the center). In the second, *Contextual Conditioning* phase that lasted for 700 trials, reward probability was 75% for the targets in a selected high-reward chamber (fixed for each agent) and 25% for the targets in the other two, low-reward chambers. The amount of reward obtained was increased to maintain the overall reward availability compatible to that in the first phase. The number of simulated learning trials matched those of the animal experiment [19].

In our simulations, we test two different learning procedures: a *baseline* procedure and a *forward sweep* procedure. The two procedures use the same methods for learning the state space and the state-state (transition) models. However, they differ in how they learn the state-value model. They both use the transition model to retrieve the value of possible future states

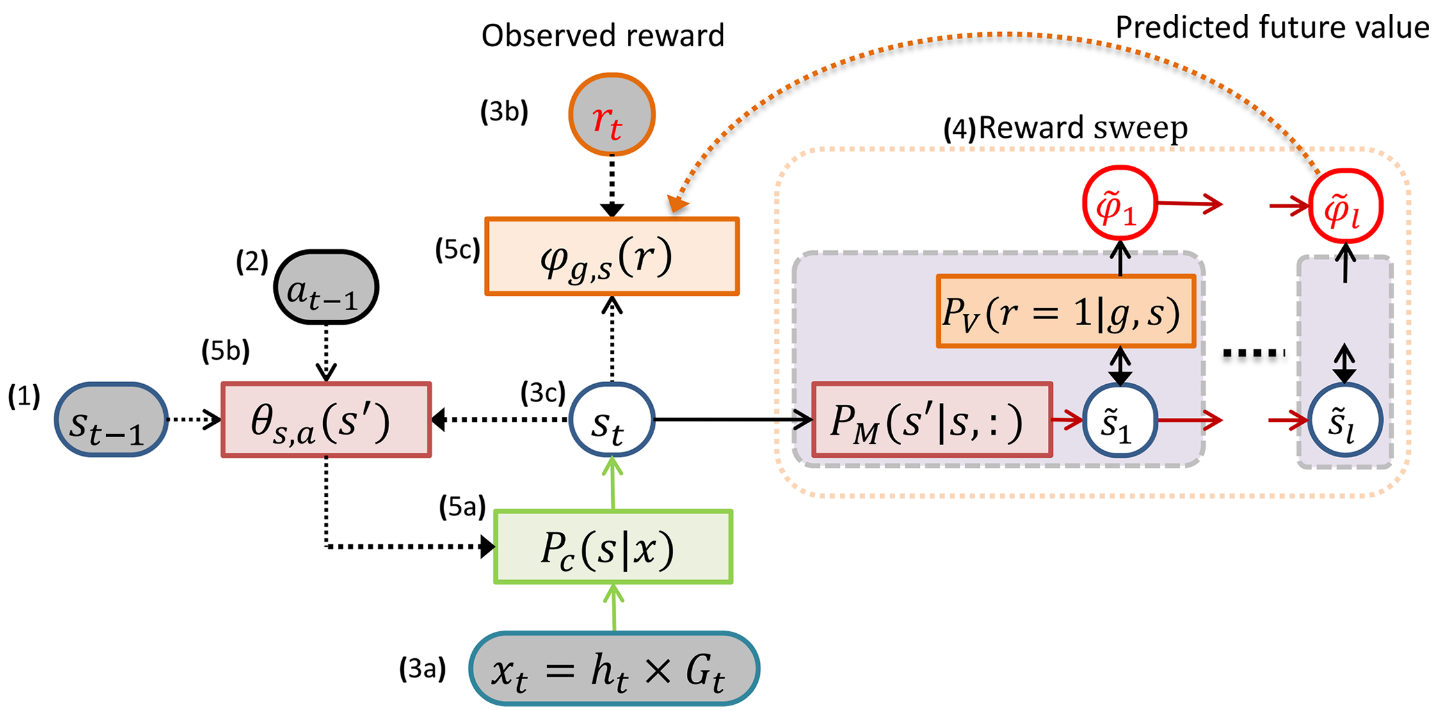

Fig 11. Learning the latent state-space, the state-transition and state-value models. Given (1) the last input x_{t-1} and latent state s_{t-1}, (2) performed action a_{t-1}, (3) observed new input x_t, reward r_t, and inferred latent state s_t, learning consists of (5) adjusting the categorization model $P_c^t(s_i|x_{t-1})$ to make it more congruent with the state-transition model and updating the conjugate priors $\theta_{s_{t-1},a_{t-1}}(s_t)$ and $\varphi_{g,s_t}(:)$ of the state-transition and state-value models to accommodate the internal perception of the experienced behavioral evidence (see the text for details). Notably, the update of the state-value conjugate is a Bayesian analog of TD-learning using predicted discounted future value $\tilde{\varphi}$ accumulated in (4) a forward sweep.

https://doi.org/10.1371/journal.pcbi.1006316.g011

and to update the value function on-line; however, the baseline learning procedure uses a short prediction (equivalent to a sweep of length 1) while the forward sweep procedure uses a longer look-ahead (i.e., sweeps of length 9—a length that is compatible with forward sweeps reported in the literature [7]).

Below we explain in detail how (the probability distributions of) the three components of the MB-RL are learned. All the learning procedures essentially consist in performing Bayesians update of the (conjugate priors of the) probability distributions, in the light of novel observed evidence. Note that hereafter we distinguish the prior from the posterior distributions with the help of additional time-index.

State space learning. State space learning uses Bayesian nonparametrics to categorize the input signal, with Chinese Restaurant Process (CRP) priors over the emergent categories [100]. This approach offers an unbounded latent space, but the CRP prior adapts the model complexity to the data by expanding it in a conservative way. Thus, any novel, or unexperienced so far input x_{n+1} is "assigned" to a new category or state s_{new} with a small probability $\mathrm{P_c(s_{new}|x_{n+1})} = \alpha/\mathrm{A}$ (controlled by a concentration parameter α, here α = 20) or to previously initialized latent state s_i according to its popularity across all experienced input states $\mathrm{P_c(s_i|x_{n+1})} = \Sigma_j \mathrm{P_c(s_i|x_j)}/\mathrm{A}$ where $\mathrm{A} = \alpha + \Sigma_{i,j}\mathrm{P_c(s_i|x_j)}$ is a normalizing factor. At each time step t, an input signal x_t evokes the most probable input-specific latent state $s_t = argmax_s P_c^t(s|x_t)$ that is in turn used for action selection and learning. Critically, the belief in the category-assignment $P_c^t(s_i|x_{t-1})$ of the last input state x_{t-1} is scaled for every state s_i with the state-transition contingencies

$P^t_M(s_t|s_i, a_{t-1})$ (adapted from [56,102]):

$$P^{t+1}_c(s_i|x_{t-1}) = \frac{P^t_M(s_t|s_i, a_{t-1})P^t_c(s_i|x_{t-1})}{\sum_j P^t_M(s_t|s_j, a_{t-1})P^t_c(s_j|x_{t-1})} \tag{1}$$

This method creates an implicit dependency between state space and state-state (transition) learning, in the sense that states assignment or categorization is retained with higher probability if it is congruent with the emergent transition model. Thus, a novel aspect of our MB-RL approach is that state-transition and state-value learning work in synergy with category- (or state-space-) learning to afford the acquisition of an effective internal model of the experienced interactions with the external environment, from scratch.

Two technical points are worth noticing to contextualize our approach. First, our latent-state model $P_c(s|x)$ is a perceptual model, which permits inferring latent state s (e.g., the current location) corresponding to the current sensory state x. This is distinct from the SR approach [35,69,75,76], which encodes a predictive representation of future states (e.g., of future locations). The two learning methods may seem similar, especially because we use a part of the transition model $P_M(s'|s,a)$ to learn a behaviorally relevant space of latent states, see Eq (1). However, it is important to note that we use the probability of the "source" state s, not of the "outcome" s', to adjust the perceptual model. This renders the state space and state transition model coherent, but does not yield predictive representations as in the SR approach. In other words, the predictive aspect of the MB-RL agent consists in using model-based, look-ahead prediction, as opposed to using predictive state representations as in the SR approach.

Second, in our simulations, the concentration (α) parameter plays a permissive role in the expansion of the latent state-space: the smaller the α, the smaller the probability that new latent states will be formed. This parameter operates only within the state space learning mechanism and is fully distinct from the parameters of the other model components, such as the (information-driven) mechanism that sets the depth of sweeps explained above, or the discounting factors.

State-transition model. After executing an action a_{t-1}, the agent obtains a new observation (sensory measurement x_t), infers the corresponding latent state s_t, and updates its transition distribution $\mathrm{s_{t-1} \times a_{t-1} \longrightarrow s_t}$, i.e., learns the transition model (see Fig 11). To that aim, the conjugate Dirichlet prior of the conditional multinomial distribution is updated with the new evidence in the latent-space domain: $\theta^{\mathrm{t+1}}_{\mathrm{s_{t-1},a_{t-1}}}(\mathrm{s_t}) = \theta^{\mathrm{t}}_{\mathrm{s_{t-1},a_{t-1}}}(\mathrm{s_t}) + 1$ and then normalized, to obtain the posterior of the state-transition multinomial $P^{t+1}_M(s'|s_{t-1}, a_{t-1})$.

State-value model. The state-value model $P^t_v(r|g, s)$ is updated using a temporal-difference (TD) like manner that at each learning step accounts for the past experience, or conjugate prior $\varphi^t_{g,s}$, observed true reward r_t, and predicted future value. The agent makes internal simulations (or sweeps) of length l (here, $l = 1$ for the baseline learning procedure and $l = 9$ for the forward sweep learning procedure) conditioned on the latent state s_t. During the sweeps, the agent accumulates expected future values in terms of parameters of the conjugate distribution of the model of reward, $\tilde{\varphi} = \sum_{\mathrm{i=1...l}} \gamma^{\mathrm{i}} \varphi_{\mathrm{g_t,\tilde{s}_i}}$, applying a discount factor (here, $\gamma = 0.90$). At each step i of the sweep, the BM-RL agent exploits a mechanism analogous to the forward sweeps used for control to collect predicted state values. As shown on Fig 11, the mechanism uses the state-transition and the state-value models, and local maximization to infer the latent states of the sweep, i.e., the states with the greatest reward expectancy among all the states that are achievable within one action from the previous state: $\mathrm{\tilde{s}_i = argmax_{s'} P_v(r = 1|g, argmax_{s'} P_M(s'|\tilde{s}_{i-1}, :))}$ where $\tilde{s}_0 = s_t$. Then, the conjugate prior of the conditional binomial distribution of reward is updated in a way that is analogous to temporal difference (TD) learning [1] and which considers the (immediately) observed reward $\mathrm{r_t}$, and value $\varphi_{\mathrm{Obs}} = [1 - \mathrm{r_t}, \mathrm{r_t}]$ and the (predicted) future value expectancy: $\varphi^{\mathrm{t+1}}_{\mathrm{g_t,s_t}} = \varphi^{\mathrm{t}}_{\mathrm{g_t,s_t}} + \alpha(\varphi_{\mathrm{Obs}} + \tilde{\varphi} - \varphi^{\mathrm{t}}_{\mathrm{g_t,s_t}})$. The learning coefficient α decreases along with time on a log scale ($\alpha = \alpha_0/\log_{10}(t)$, $\alpha_0 = 1.5$). This learning schedule gradually shifts the value-learning policy, from relying on new evidence to exploiting acquired knowledge. Finally, the posterior is obtained by normalizing the updated conjugate: $P^{t+1}_{v}(:|g_t, s_t) = \varphi^{t+1}_{g_t,s_t}(:)/\sum_{r=0..1}\varphi^{t+1}_{g_t,s_t}(r)$.

Supporting information

S1 File. Algorithm of sweep-based action selection.
(DOCX)

Acknowledgments

We thank G. T. Meijer for help with the preparation of Fig 1A and Andrew Wikenheiser.

Author Contributions

Conceptualization: Ivilin Peev Stoianov, Cyriel M. A. Pennartz, Giovani Pezzulo.

Formal analysis: Ivilin Peev Stoianov, Giovani Pezzulo.

Funding acquisition: Giovani Pezzulo.

Methodology: Ivilin Peev Stoianov, Cyriel M. A. Pennartz, Carien S. Lansink, Giovani Pezzulo.

Software: Ivilin Peev Stoianov.

Validation: Ivilin Peev Stoianov, Cyriel M. A. Pennartz, Carien S. Lansink, Giovani Pezzulo.

Writing – original draft: Ivilin Peev Stoianov, Giovani Pezzulo.

Writing – review & editing: Cyriel M. A. Pennartz, Carien S. Lansink.

References

1. Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge MA: MIT Press; 1998.
2. Dolan RJ, Dayan P. Goals and habits in the brain. Neuron. 2013; 80: 312–325. https://doi.org/10.1016/j.neuron.2013.09.007 PMID: 24139036
3. Daw ND, Gershman SJ, Seymour B, Dayan P, Dolan RJ. Model-based influences on humans' choices and striatal prediction errors. Neuron. 2011; 69: 1204–1215. https://doi.org/10.1016/j.neuron.2011.02.027 PMID: 21435563
4. Daw ND, Niv Y, Dayan P. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nat Neurosci. 2005; 8: 1704–1711. https://doi.org/10.1038/nn1560 PMID: 16286932
5. O'Keefe J, Dostrovsky J. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain Res Vol. 1971; 34: 171–175.
6. Schultz W, Apicella P, Scarnati E, Ljungberg T. Neuronal activity in monkey ventral striatum related to the expectation of reward. J Neurosci. 1992; 12: 4595–4610. PMID: 1464759
7. Johnson A, Redish AD. Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. J Neurosci. 2007; 27: 12176–12189. https://doi.org/10.1523/JNEUROSCI.3761-07.2007 PMID: 17989284
8. Pennartz CMA, Ito R, Verschure PFMJ, Battaglia FP, Robbins TW. The hippocampal-striatal axis in learning, prediction and goal-directed behavior. Trends Neurosci. 2011; 34: 548–559. https://doi.org/10.1016/j.tins.2011.08.001 PMID: 21889806
9. Van der Meer MAA, Redish AD. Covert Expectation-of-Reward in Rat Ventral Striatum at Decision Points. Front Integr Neurosci. 2009; 3: 1. https://doi.org/10.3389/neuro.07.001.2009 PMID: 19225578

10. Verschure P, Pennartz CMA, Pezzulo G. The why, what, where, when and how of goal-directed choice: neuronal and computational principles. Philos Trans R Soc Lond B Biol Sci. 2014: 369: 20130483. https://doi.org/10.1098/rstb.2013.0483 PMID: 25267825

11. McDannald MA, Lucantonio F, Burke KA, Niv Y, Schoenbaum G. Ventral striatum and orbitofrontal cortex are both required for model-based, but not model-free, reinforcement learning. J Neurosci. 2011; 31: 2700–2705. https://doi.org/10.1523/JNEUROSCI.5499-10.2011 PMID: 21325538

12. Van der Meer M, Kurth-Nelson Z, Redish AD. Information processing in decision-making systems. The Neuroscientist. 2012; 18: 342–359. https://doi.org/10.1177/1073858411435128 PMID: 22492194

13. Pezzulo G, van der Meer MAA, Lansink CS, Pennartz CMA. Internally generated sequences in learning and executing goal-directed behavior. Trends Cogn Sci. 2014; 18: 647–657. https://doi.org/10.1016/j.tics.2014.06.011 PMID: 25156191

14. Pezzulo G, Kemere C, Meer M van der. Internally generated hippocampal sequences as a vantage point to probe future-oriented cognition. Ann N Y Acad Sci. 2017; 1396: 144–165. https://doi.org/10.1111/nyas.13329 PMID: 28548460

15. Penagos H, Varela C, Wilson MA. Oscillations, neural computations and learning during wake and sleep. Curr Opin Neurobiol. 2017; 44: 193–201. https://doi.org/10.1016/j.conb.2017.05.009 PMID: 28570953

16. Miller KJ, Botvinick MM, Brody CD. Dorsal hippocampus contributes to model-based planning. Nat Neurosci. 2017; 20: 1269–1276. https://doi.org/10.1038/nn.4613 PMID: 28758995

17. Pezzulo G, Donnarumma F, Iodice P, Maisto D, Stoianov I. Model-Based Approaches to Active Perception and Control. Entropy. 2017; 19: 266. https://doi.org/10.3390/e19060266

18. Daw ND, Dayan P. The algorithmic anatomy of model-based evaluation. Philos Trans R Soc B Biol Sci. 2014; 369: 20130478.

19. Lansink CS, Jackson J, Lankelma JV, Ito R, Robbins TW, Everitt BJ, et al. Reward cues in space: commonalities and differences in neural coding by hippocampal and ventral striatal ensembles. J Neurosci Off J Soc Neurosci. 2012; 32: 12444–12459. https://doi.org/10.1523/JNEUROSCI.0593-12.2012 PMID: 22956836

20. Redish AD. Vicarious trial and error. Nat Rev Neurosci. 2016; 17: 147–159. https://doi.org/10.1038/nrn.2015.30 PMID: 26891625

21. Van der Meer MAA, Redish AD. Expectancies in decision making, reinforcement learning, and ventral striatum. Front Neurosci. 2010; 4.

22. Penny WD, Zeidman P, Burgess N. Forward and Backward Inference in Spatial Cognition. PLoS Comput Biol. 2013; 9: e1003383. https://doi.org/10.1371/journal.pcbi.1003383 PMID: 24348230

23. Pezzulo G, Rigoli F, Chersi F. The Mixed Instrumental Controller: using Value of Information to combine habitual choice and mental simulation. Front Cogn. 2013; 4: 92. https://doi.org/10.3389/fpsyg.2013.00092 PMID: 23459512

24. Tesauro G, Galperin GR. On-line Policy Improvement Using Monte-Carlo Search. Proceedings of the 9th International Conference on Neural Information Processing Systems. Cambridge, MA, USA: MIT Press; 1996. pp. 1068–1074. Available: http://dl.acm.org/citation.cfm?id=2998981.2999131

25. Lansink CS, Goltstein PM, Lankelma JV, McNaughton BL, Pennartz CM. Hippocampus leads ventral striatum in replay of place-reward information. PLoS Biol. 2009; 7: e1000173. https://doi.org/10.1371/journal.pbio.1000173 PMID: 19688032

26. Pfeiffer BE, Foster DJ. Hippocampal place-cell sequences depict future paths to remembered goals. Nature. 2013; 497: 74–79. https://doi.org/10.1038/nature12112 PMID: 23594744

27. Wilson MA, McNaughton BL. Reactivation of hippocampal ensemble memories during sleep. Science. 1994; 265: 676–679. PMID: 8036517

28. Diba K, Buzsáki G. Forward and reverse hippocampal place-cell sequences during ripples. Nat Neurosci. 2007; 10: 1241–1242. https://doi.org/10.1038/nn1961 PMID: 17828259

29. Sutton RS. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. Proceedings of the Seventh International Conference on Machine Learning. San Mateo, CA: Morgan Kaufmann; 1990. pp. 216–224.

30. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature. 2015; 518: 529–533. https://doi.org/10.1038/nature14236 PMID: 25719670

31. Ito R, Robbins TW, McNaughton BL, Everitt BJ. Selective excitotoxic lesions of the hippocampus and basolateral amygdala have dissociable effects on appetitive cue and place conditioning based on path integration in a novel Y-maze procedure. Eur J Neurosci. 2006; 23: 3071–3080. https://doi.org/10.1111/j.1460-9568.2006.04883.x PMID: 16819997

32. Ito R, Robbins TW, Pennartz CM, Everitt BJ. Functional interaction between the hippocampus and nucleus accumbens shell is necessary for the acquisition of appetitive spatial context conditioning. J Neurosci. 2008; 28: 6950–6959. https://doi.org/10.1523/JNEUROSCI.1615-08.2008 PMID: 18596169

33. O'Reilly RC, Wyatte DR, Rohrlich J. Deep Predictive Learning: A Comprehensive Model of Three Visual Streams. ArXiv170904654 Q-Bio. 2017; Available: http://arxiv.org/abs/1709.04654

34. Redish AD. The Mind within the Brain: How We Make Decisions and How those Decisions Go Wrong. 1 edition. Oxford: Oxford University Press; 2013.

35. Dayan P. Improving generalization for temporal difference learning: The successor representation. Neural Comput. 1993; 5: 613–624.

36. Momennejad I, Russek EM, Cheong JH, Botvinick MM, Daw N, Gershman SJ. The successor representation in human reinforcement learning. bioRxiv. 2016; 083824. https://doi.org/10.1101/083824

37. Hok V, Save E, Lenck-Santini PP, Poucet B. Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex. Proc Natl Acad Sci U S A. 2005; 102: 4602–4607. https://doi.org/10.1073/pnas.0407332102 PMID: 15761059

38. Wikenheiser AM, Redish AD. Hippocampal theta sequences reflect current goals. Nat Neurosci. 2015; 18: 289–294. https://doi.org/10.1038/nn.3909 PMID: 25559082

39. Schultz W, Dayan P, Montague PR. A neural substrate of prediction and reward. Science. 1997; 275: 1593–1599. PMID: 9054347

40. Botvinick M, Niv Y, Barto A. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. Cognition. 2008; http://dx.doi.org/10.1016/j.cognition.2008.08.011

41. Houk JC, Adams JL, Barto AG. A model of how the basal ganglia generates and uses neural signals that predict reinforcement. In: Houk JC, Davis J, Beiser D, editors. Models of Information Processing in the Basal Ganglia. Cambridge: MIT Press; 1995. pp. 249–270.

42. Solway A, Botvinick MM. Goal-directed decision making as probabilistic inference: A computational framework and potential neural correlates. Psychol Rev. 2012; 119: 120–154. https://doi.org/10.1037/a0026435 PMID: 22229491

43. Marr D. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information [Internet]. New York, NY, USA: Henry Holt and Co., Inc.; 1982. Available: http://portal.acm.org/citation.cfm?id=1096911

44. Lisman J, Redish AD. Prediction, Sequences and the Hippocampus. Philos Trans R Soc B Biol Sci. 2009; 364: 1193–1201. https://doi.org/10.1098/rstb.2008.0316 PMID: 19528000

45. Dragoi G, Buzsáki G. Temporal encoding of place sequences by hippocampal cell assemblies. Neuron. 2006; 50: 145–157. https://doi.org/10.1016/j.neuron.2006.02.023 PMID: 16600862

46. Erdem UM, Hasselmo ME. A biologically inspired hierarchical goal directed navigation model. J Physiol Paris. 2014; 108: 28–37. https://doi.org/10.1016/j.jphysparis.2013.07.002 PMID: 23891644

47. Chersi F, Pezzulo G. Using hippocampal-striatal loops for spatial navigation and goal-directed decision-making. Cogn Process. 2012; 13: 125–129.

48. Chersi F, Donnarumma F, Pezzulo G. Mental imagery in the navigation domain: A computational model of sensory-motor simulation mechanisms. Adaptive Behavior. 2013: 251–262.

49. Friston K, Rigoli F, Ognibene D, Mathys C, Fitzgerald T, Pezzulo G. Active inference and epistemic value. Cogn Neurosci. 2015;0: 1–28. https://doi.org/10.1080/17588928.2015.1020053 PMID: 25689102

50. Friston K, FitzGerald T, Rigoli F, Schwartenbeck P, Pezzulo G. Active Inference: A Process Theory. Neural Comput. 2016; 1–49. https://doi.org/10.1162/NECO_a_00912 PMID: 27870614

51. Friston K, FitzGerald T, Rigoli F, Schwartenbeck P, O'Doherty J, Pezzulo G. Active inference and learning. Neurosci Biobehav Rev. 2016; 68: 862–879. https://doi.org/10.1016/j.neubiorev.2016.06.022 PMID: 27375276

52. Pezzulo G, Cartoni E, Rigoli F, Pio-Lopez L, Friston K. Active Inference, epistemic value, and vicarious trial and error. Learn Mem. 2016; 23: 322–338. https://doi.org/10.1101/lm.041780.116 PMID: 27317193

53. Collins AG, Frank MJ. Cognitive control over learning: creating, clustering, and generalizing task-set structure. Psychol Rev. 2013; 120: 190. https://doi.org/10.1037/a0030852 PMID: 23356780

54. Collins A, Koechlin E. Reasoning, Learning, and Creativity: Frontal Lobe Function and Human Decision-Making. PLoS Biol. 2012; 10. https://doi.org/10.1371/journal.pbio.1001293 PMID: 22479152

55. Donoso M, Collins AG, Koechlin E. Foundations of human reasoning in the prefrontal cortex. Science. 2014; 344: 1481–1486. https://doi.org/10.1126/science.1252254 PMID: 24876345

56. Stoianov I, Genovesio A, Pezzulo G. Prefrontal Goal Codes Emerge as Latent States in Probabilistic Value Learning. J Cogn Neurosci. 2015; 28: 140–157. https://doi.org/10.1162/jocn_a_00886 PMID: 26439267

57. Pezzulo G, Rigoli F, Friston KJ. Hierarchical Active Inference: A Theory of Motivated Control. Trends Cogn Sci. 2018; https://doi.org/10.1016/j.tics.2018.01.009 PMID: 29475638

58. Pezzulo G, Rigoli F. The value of foresight: how prospection affects decision-making. Front Neurosci. 2011; 5.

59. Pezzulo G, Cisek P. Navigating the Affordance Landscape: Feedback Control as a Process Model of Behavior and Cognition. Trends Cogn Sci. 2016; 20: 414–424. https://doi.org/10.1016/j.tics.2016.03.013 PMID: 27118642

60. Hok V, Lenck-Santini P-P, Roux S, Save E, Muller RU, Poucet B. Goal-Related Activity in Hippocampal Place Cells. J Neurosci. 2007; 27: 472–482. https://doi.org/10.1523/JNEUROSCI.2864-06.2007 PMID: 17234580

61. Botvinick M, Weinstein A. Model-based hierarchical reinforcement learning and human action control. Philos Trans R Soc Lond B Biol Sci. 2014; 369: 20130480. https://doi.org/10.1098/rstb.2013.0480 PMID: 25267822

62. Donnarumma F, Maisto D, Pezzulo G. Problem Solving as Probabilistic Inference with Subgoaling: Explaining Human Successes and Pitfalls in the Tower of Hanoi. Sporns O, editor. PLOS Comput Biol. 2016; 12: e1004864. https://doi.org/10.1371/journal.pcbi.1004864 PMID: 27074140

63. Solway A, Diuk C, Córdova N, Yee D, Barto AG, Niv Y, et al. Optimal Behavioral Hierarchy. PLOS Comput Biol. 2014; 10: e1003779. https://doi.org/10.1371/journal.pcbi.1003779 PMID: 25122479

64. Maisto D, Donnarumma F, Pezzulo G. Divide et impera: subgoaling reduces the complexity of probabilistic inference and problem solving. J R Soc Interface. 2015; 12: 20141335. https://doi.org/10.1098/rsif.2014.1335 PMID: 25652466

65. Maisto D, Donnarumma F, Pezzulo G. Nonparametric Problem-Space Clustering: Learning Efficient Codes for Cognitive Control Tasks. Entropy. 2016; 18: 61. https://doi.org/10.3390/e18020061

66. Botvinick MM. Hierarchical models of behavior and prefrontal function. Trends Cogn Sci. 2008; 12: 201–208. https://doi.org/10.1016/j.tics.2008.02.009 PMID: 18420448

67. Friston K. Hierarchical Models in the Brain. PLoS Comput Biol. 2008; 4: e1000211. https://doi.org/10.1371/journal.pcbi.1000211 PMID: 18989391

68. Franzius M, Sprekeler H, Wiskott L. Slowness and sparseness lead to place, head-direction, and spatial-view cells. PLoS Comput Biol. 2007; 3: e166. https://doi.org/10.1371/journal.pcbi.0030166 PMID: 17784780

69. Stachenfeld KL, Botvinick MM, Gershman SJ. The hippocampus as a predictive map. Nat Neurosci. 2017; 20: 1643–1653. https://doi.org/10.1038/nn.4650 PMID: 28967910

70. McClelland JL, McNaughton BL, O'Reilly RC. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. Psychol Rev. 1995; 102: 419–457. PMID: 7624455

71. Friston KJ, Lin M, Frith CD, Pezzulo G, Hobson JA, Ondobaka S. Active Inference, Curiosity and Insight. Neural Comput. 2017; 1–51. https://doi.org/10.1162/neco_a_00999 PMID: 28777724

72. Kumaran D, Hassabis D, McClelland JL. What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated. Trends Cogn Sci. 2016; 20: 512–534. https://doi.org/10.1016/j.tics.2016.05.004 PMID: 27315762

73. Watkins CJCH, Dayan P. Q-learning. Mach Learn. 1992; 8: 279–292.

74. Spiers HJ, Gilbert SJ. Solving the detour problem in navigation: a model of prefrontal and hippocampal interactions. Front Hum Neurosci. 2015; 9. https://doi.org/10.3389/fnhum.2015.00125 PMID: 25852515

75. Momennejad I, Russek EM, Cheong JH, Botvinick MM, Daw ND, Gershman SJ. The successor representation in human reinforcement learning. Nat Hum Behav. 2017; 1: 680. https://doi.org/10.1038/s41562-017-0180-8

76. Russek EM, Momennejad I, Botvinick MM, Gershman SJ, Daw ND. Predictive representations can link model-based reinforcement learning to model-free mechanisms. PLOS Comput Biol. 2017; 13: e1005768. https://doi.org/10.1371/journal.pcbi.1005768 PMID: 28945743

77. Hok V, Chah E, Save E, Poucet B. Prefrontal cortex focally modulates hippocampal place cell firing patterns. J Neurosci. 2013; 33: 3443–3451. https://doi.org/10.1523/JNEUROSCI.3427-12.2013 PMID: 23426672

78. Hok V, Save E, Lenck-Santini PP, Poucet B. Coding for spatial goals in the prelimbic/infralimbic area of the rat frontal cortex. Proc Natl Acad Sci U S A. 2005; 102: 4602–4607. https://doi.org/10.1073/pnas.0407332102 PMID: 15761059

79. Valentin VV, Dickinson A, O'Doherty JP. Determining the Neural Substrates of Goal-Directed Learning in the Human Brain. J Neurosci. 2007; 27: 4019–4026. https://doi.org/10.1523/JNEUROSCI.0564-07.2007 PMID: 17428979

80. Stott JJ, Redish AD. A functional difference in information processing between orbitofrontal cortex and ventral striatum during decision-making behaviour. Philos Trans R Soc Lond B Biol Sci. 2014; 369: 20130472. https://doi.org/10.1098/rstb.2013.0472 PMID: 25267815

81. Terada S, Sakurai Y, Nakahara H, Fujisawa S. Temporal and Rate Coding for Discrete Event Sequences in the Hippocampus. Neuron. 2017; 94: 1248–1262.e4. https://doi.org/10.1016/j.neuron.2017.05.024 PMID: 28602691

82. Keramati M, Dezfouli A, Piray P. Speed/accuracy trade-off between the habitual and the goal-directed processes. PLoS Comput Biol. 2011; 7: e1002055. https://doi.org/10.1371/journal.pcbi.1002055 PMID: 21637741

83. Foster DJ, Wilson MA. Hippocampal theta sequences. Hippocampus. 2007; 17: 1093–1099. https://doi.org/10.1002/hipo.20345 PMID: 17663452

84. Buzsáki G. Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. Hippocampus. 2015; 25: 1073–1188. https://doi.org/10.1002/hipo.22488 PMID: 26135716

85. Ambrose RE, Pfeiffer BE, Foster DJ. Reverse Replay of Hippocampal Place Cells Is Uniquely Modulated by Changing Reward. Neuron. 2016; 91: 1124–1136. https://doi.org/10.1016/j.neuron.2016.07.047 PMID: 27568518

86. Kalenscher T, Pennartz CMA. Is a bird in the hand worth two in the future? The neuroeconomics of intertemporal decision-making. Prog Neurobiol. 2008; 84: 284–315. https://doi.org/10.1016/j.pneurobio.2007.11.004 PMID: 18207301

87. Foster DJ, Wilson MA. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. Nature. 2006; 440: 680–683. https://doi.org/10.1038/nature04587 PMID: 16474382

88. Mattar MG, Daw ND. Prioritized memory access explains planning and hippocampal replay. bioRxiv. 2017; 225664. https://doi.org/10.1101/225664

89. Donnarumma F, Costantini M, Ambrosini E, Friston K, Pezzulo G. Action perception as hypothesis testing. Cortex. 2017; https://doi.org/10.1016/j.cortex.2017.01.016 PMID: 28226255

90. Collin SHP, Milivojevic B, Doeller CF. Memory hierarchies map onto the hippocampal long axis in humans. Nat Neurosci. 2015; 18: 1562–1564. https://doi.org/10.1038/nn.4138 PMID: 26479587

91. Strange BA, Witter MP, Lein ES, Moser EI. Functional organization of the hippocampal longitudinal axis. Nat Rev Neurosci. 2014; 15: 655–669. https://doi.org/10.1038/nrn3785 PMID: 25234264

92. Kjelstrup KB, Solstad T, Brun VH, Hafting T, Leutgeb S, Witter MP, et al. Finite scale of spatial representation in the hippocampus. Science. 2008; 321: 140–143. https://doi.org/10.1126/science.1157086 PMID: 18599792

93. Fuhs MC, Touretzky DS. A spin glass model of path integration in rat medial entorhinal cortex. J Neurosci Off J Soc Neurosci. 2006; 26: 4266–4276. https://doi.org/10.1523/JNEUROSCI.4353-05.2006 PMID: 16624947

94. McNaughton BL, Battaglia FP, Jensen O, Moser EI, Moser M-B. Path integration and the neural basis of the "cognitive map." Nat Rev Neurosci. 2006; 7: 663–678. https://doi.org/10.1038/nrn1932 PMID: 16858394

95. Burgess N, Barry C, O'Keefe J. An oscillatory interference model of grid cell firing. Hippocampus. 2007; 17: 801–812. https://doi.org/10.1002/hipo.20327 PMID: 17598147

96. Muller RU, Ranck JB, Taube JS. Head direction cells: properties and functional significance. Curr Opin Neurobiol. 1996; 6: 196–206. PMID: 8725961

97. Harland B, Grieves RM, Bett D, Stentiford R, Wood ER, Dudchenko PA. Lesions of the Head Direction Cell System Increase Hippocampal Place Field Repetition. Curr Biol. 2017; 27: 2706–2712.e2. https://doi.org/10.1016/j.cub.2017.07.071 PMID: 28867207

98. Giocomo LM, Moser M-B, Moser EI. Computational Models of Grid Cells. Neuron. 2011; 71: 589–603. https://doi.org/10.1016/j.neuron.2011.07.023 PMID: 21867877

99. Solstad T, Moser EI, Einevoll GT. From grid cells to place cells: a mathematical model. Hippocampus. 2006; 16: 1026–1031. https://doi.org/10.1002/hipo.20244 PMID: 17094145

100. Gershman SJ, Blei DM. A tutorial on Bayesian nonparametric models. J Math Psychol. 2012; 56: 1–12.

101. Bishop CM. Pattern Recognition and Machine Learning. Springer; 2006.

102. Sanborn AN, Griffiths TL, Navarro DJ. Rational approximations to rational models: alternative algorithms for category learning. Psychol Rev. 2010; 117: 1144–1167. https://doi.org/10.1037/a0020511 PMID: 21038975