Predictive representations can link model-based reinforcement learning to model-free mechanisms

Evan M. Russek[1]*¶, Ida Momennejad[2]¶, Matthew M. Botvinick[3], Samuel J. Gershman[4], Nathaniel D. Daw[2]

[1] Center for Neural Science, New York University, New York, NY, United States of America

[2] Princeton Neuroscience Institute and Department of Psychology, Princeton University, Princeton, NJ, United States of America

[3] Google DeepMind, London, United Kingdom

[4] Department of Psychology and Center for Brain Science, Harvard University, Cambridge, MA, United States of America

* Corresponding author
E-mail: emr443@nyu.edu

¶ EMR and IM contributed equally to this work

Abstract

Humans and animals are capable of evaluating actions by considering their long-run future rewards through a process described using model-based reinforcement learning (RL) algorithms. The mechanisms by which neural circuits perform the computations prescribed by model-based RL remain largely unknown; however, multiple lines of evidence suggest that neural circuits supporting model-based behavior are structurally homologous to and overlapping with those thought to carry out model-free temporal difference (TD) learning. Here, we lay out a family of approaches by which model-based computation may be built upon a core of TD learning. The foundation of this framework is the successor representation, a predictive state representation that, when combined with TD learning of value predictions, can produce a subset of the behaviors associated with model-based learning at a fraction of the computational cost. Using simulations, we delineate the precise behavioral capabilities enabled by evaluating actions using this approach, and compare them to those demonstrated by biological organisms. We then introduce two new algorithms that build upon the successor representation while progressively mitigating its limitations. Because this framework can account for the full range of observed putatively model-based behaviors while still utilizing a core TD framework, we suggest that it represents a neurally plausible family of mechanisms for model-based evaluation.

Author Summary

According to standard models, when confronted with a choice, animals and humans rely on two separate, distinct processes to come to a decision. One process deliberatively evaluates the consequences of each candidate action and is thought to underlie the ability to flexibly come up with novel plans. The other process gradually increases the propensity to perform behaviors that were previously successful and is thought to underlie automatically executed, habitual reflexes. Although computational principles and animal behavior support this dichotomy, at the neural level, there is little evidence supporting a clean segregation. For instance, although dopamine — famously implicated in drug addiction and Parkinson's disease — currently only has a well-defined role in the automatic process, evidence suggests that it also plays a role in the deliberative process. In this work, we present a computational framework for resolving this mismatch. We show that the types of behaviors associated with either process could result from a common learning mechanism applied to different strategies for how populations of neurons could represent candidate actions. In addition to demonstrating that this account can produce the full range of flexible behavior observed in the empirical literature, we suggest experiments that could detect the various approaches within this framework.

Introduction

A key question in both neuroscience and psychology is how the brain evaluates candidate actions in complex, sequential decision tasks. In principle, computing an action's expected long-run cumulative future reward (or *value*) requires averaging rewards over the many future state trajectories that might follow the action. In practice, the exact computation of such expectations by dynamic programming or tree search methods may be prohibitively expensive, and it is widely believed that the brain simplifies the computations occurring at decision time, in part by relying on "cached" (pre-computed) long-run value estimates [1].

Such caching of values is the hallmark of prominent temporal difference (TD) learning theories, according to which prediction errors reported by phasic dopamine responses update these cached variables [2–4]. This, in turn, provides a neuro-computational account of inflexible stimulus-response habits, due to the fact that TD learning cannot update cached values in response to distal changes in reward (e.g., following reward devaluation). The computationally cheap but inflexible "model-free" nature of TD learning, which relies only on trial-and-error interactions with the environment, contrasts with the flexible but computationally expensive "model-based" nature of dynamic programming and tree search methods, which rely on an explicit internal model of the environment. Due to the complementary properties of model-free and model-based value computation, it has been suggested that the brain makes use of both in the form of parallel RL systems that compete for control of behavior [1].

Although flexible choice behavior seems to demonstrate that humans and animals may use model-based evaluation in some circumstances, very little is known about how this is actually implemented in the brain. Suggestively, but also confusingly, multiple lines of evidence suggest that the dopamine system and its key targets are implicated not just in the model-free behaviors that theory endows them with, but also in the more flexible choice adjustments that seem to reflect model-based learning [5–11]. This is particularly perplexing for the standard account because typical model-based algorithms do not make use of a TD prediction error of the sort associated with dopamine and instead store internal variables (specifically, predictions about immediate "one-step" rather than long-run consequences of actions), which require different update rules and error signals [12, 13]. Additionally, at choice time, such algorithms require computations that are structurally different than those typically prescribed to the dopaminergic-striatal circuitry [14].

In this article, we revisit and extend a different reinforcement learning algorithm, based on the *successor representation* (SR) [15,16; see also 17-21], a predictive state representation that can endow TD learning with some aspects of model-based value computation, such as flexible adjustment following reward devaluation. That the SR, when combined with TD learning, can produce such flexible behavior makes it a promising candidate for a full solution to the puzzle of dopaminergic involvement in model-based learning. However, this approach, in its original form, results in behavioral inflexibilities that make it inadequate. The SR simplifies planning by relying on caching a set of intermediate quantities, expectations about cumulative future state occupancies. For this reason, unlike model-based learning, it is incapable of solving many tasks that require adjusting to changes in contingencies between actions and their long-run consequences (e.g. [22,23]).

Below, we first use simulations to delineate the precise behavioral consequences of these inflexibilities and evaluate them with respect to the empirical literature. We then introduce two new versions of the SR that progressively mitigate these limitations, and again simulate their expected consequences in terms of flexible or inflexible choice behavior. The first variant gains flexibility by minimizing pre-computation so that only the agent's action selection policy is cached. We show that this caching predicts errors that could in principle be behaviorally detectable, but have not been elicited by standard experimental tasks. Our second variant, which integrates and repurposes aspects of Sutton's Dyna architecture [24], eliminates even this limitation by using replay of previous experiences to update a cached SR. Both of these approaches can account for human and animal behavior in a wide range of planning tasks, and also provide an explanation as to why model-based and model-free learning should rely on overlapping neural substrates. Overall, these approaches represent a family of plausible and computationally efficient hypothetical mechanisms for the full range of flexible behaviors associated with model-based learning, which could provide a clear theoretical foundation for future experimental work.

Results

Formalism: Model-based and model-free reinforcement learning

We begin by briefly reviewing the formalism of reinforcement learning in a Markov decision process (MDP), which provides the foundation for our simulations (see [22] or [23] for a fuller presentation).

An MDP is defined by a set of states, a set of actions, a reward function $R(s)$ over states, and a state transition distribution, $P(s'|s,a)$, where a denotes the chosen action. States and rewards occur sequentially according to these one-step functions, driven by a series of actions; the goal is to learn to choose a probabilistic policy over actions, denoted by π, that maximizes the expected cumulative discounted reward (value), denoted by $V^{\pi}(s)$. The value function is defined recursively as the sum of the state's immediate reward, $R(s)$, and the value of its successor state s', averaged over possible actions, a, and transitions that would occur if the agent chose according to π:

$$V^{\pi}(s) = R(s) + \gamma \sum_a \pi(a|s) \sum_{s'} P(s'|s,a) V^{\pi}(s') \tag{1}$$

Here, γ is a parameter controlling temporal discounting. The value function under the optimal policy is given by:

$$V^{*}(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V^{*}(s') \tag{2}$$

Knowledge of the value function can help to guide choices. For instance, we can define the state-action value function as the value of choosing action a and following π thereafter:

$$Q^{\pi}(s,a) = R(s) + \gamma \sum_{s'} P(s'|s,a) V^{\pi}(s') \tag{3}$$

Then at any state one could choose the action that maximizes $Q^{\pi}(s,a)$. (Formally this defines a new policy, which is as good or better than the baseline policy π; analogously, equation 2 can be used to define the optimal state-action value function, the maximization of which selects optimal actions.) Note that it is possible to write a recursive definition for Q in the same manner as equation 1, and work directly with the state-action values, rather than deriving them indirectly from V. For expositional simplicity, in this article, we work instead with V wherever possible (mainly because this is easier to depict visually, and simplifies the notation), and accordingly we assume in our simulations that the agent derives Q using equation 3 for guiding choices. This is not intended as a substantive claim; indeed, the last algorithm we propose will work directly with Q values.

The problem of reinforcement learning is then reduced to learning to predict the value function $V^{\pi}(s)$ or $V^{*}(s)$. There are two main families of approaches. Model-based algorithms learn to estimate the one-step transition and reward functions, $P(s'|s,a)$ and $R(s)$, from which it is easy to compute V^{*} (or Q^{*}) using equation 2. This typically involves unrolling the recursion in equation 2 into a series of nested sums, an algorithm known as value iteration. The alternative, model-free, approach exemplified by TD learning bypasses estimating the one-step model. Instead, it directly updates a cached estimate of the value function itself. In particular, following a transition $s \rightarrow s'$, a reward prediction error, δ, is calculated and used to update $V(s)$:

$$\delta = R(s) + \gamma V(s') - V(s) \tag{4}$$

$$V(s) \leftarrow V(s) + \alpha_{TD}\delta$$

where α_{TD} is a learning rate parameter.

The TD update rule is derived from the recursion in equation 1: each step iteratively pushes the left hand side of the equation, $V(s)$, closer to $R(s) + \gamma V(s')$ which is a one-sample estimate of the right hand side.

Finally, analogous sample-based updates may also be conducted offline (e.g., between steps of actual, "online" experience). This is a key insight of Sutton's Dyna architecture [21] (see also [24]). The approach, like TD, caches estimates of $V(s)$. Here TD learning is supplemented by additional offline updates, notably by executing equation 4 for a single state transition and reward (s, a, r, s'). Rather than coming from direct, online experiences, these ersatz experiences can either be sampled from a learned one-step model's probability distributions $P(s'|s, a)$ and $R(s)$, or instead simply replayed, model-free, from stored episodes of raw experience (s, a, r, s'). Either way, given sufficient iterations of sampling between steps of real experience, this approach can substitute for explicit value iteration and produce estimates at each step comparable to model-based approaches that more systematically solve equations 1 or 2.

A further distinction, which will become important later, is that between *on-policy* methods, based on equation 1, and *off-policy* methods, based on equation 2. On-policy methods estimate a policy-dependent value function $V^{\pi}(s)$, whereas off-policy methods directly estimate the optimal value function $V^{*}(s)$. Typically, model-based methods are off-policy (since having learned a one-step model it is possible to use equation 2 to directly compute the optimal policy); whereas different TD learning variants can be either on- or off-policy. For example, the SARSA algorithm is on-policy, whereas the Q-learning algorithm is off-policy [25].

Model-free and model-based learning in the brain

Due to the similarity between the phasic responses of midbrain dopamine neurons, and the TD prediction error δ (equation 4), it has long been suggested that this system implements TD learning [2,3]. More specifically (e.g. [4]; Fig 1a) it has been suggested that values V or Q are associated with the firing of medium spiny neurons in striatum, as a function of an input state (or state-action) representation carried by their afferent neurons in frontal cortex, and that learning of the value function is driven by dopamine-mediated adjustment of the cortico-striatal synapses connecting these neurons. Selection among these striatal value representations would then drive action choice. Although not entirely uncontroversial, a great deal of evidence about dopamine and its targets supports this hypothesis (see [26,28] for fuller reviews).

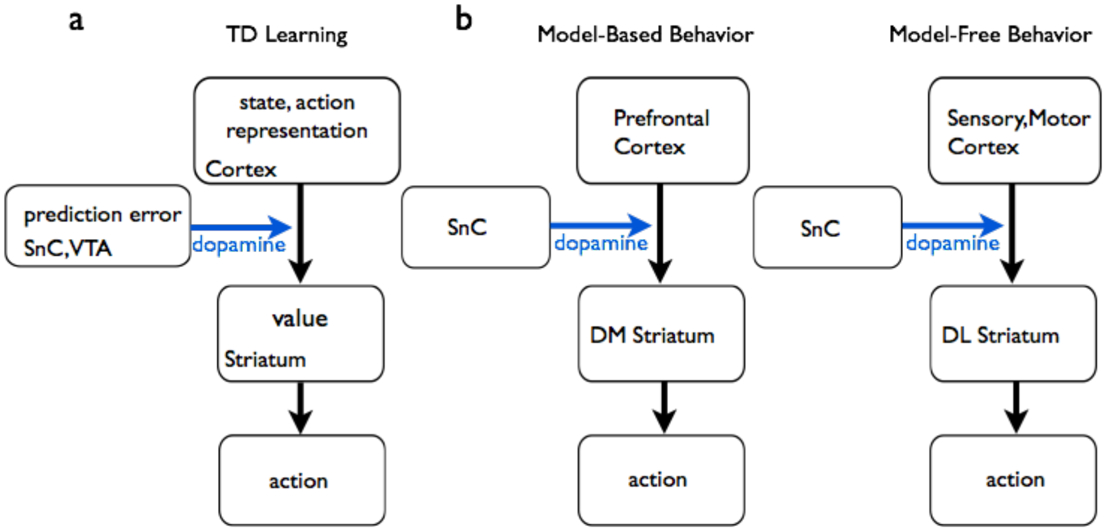

Fig. 1. Cortico-striatal loops and reinforcement learning. *a) Canonical circuit for TD learning. A dopaminergic prediction error, signaled in substantia nigra pars compacta (SnC) and ventral tegmental area (VTA), updates the value of cortically represented states and actions by modifying cortico-striatal synapses. Depending on value, represented in striatum, actions are passed through to basal-ganglia action systems. b) Results of rodent lesion studies. Lesions to a cortico-striatal loop passing through dorsomedial (DM) striatum prevent model-based learning. This area receives input from ventromedial prefrontal cortex and projects, via globus pallidus, to dorsomedial nucleus of the thalamus. Lesions to cortico-striatal loop passing through dorsolateral (DL) striatum prevent model-free learning. This area receives input from sensory and motor areas of cortex and projects, via globus pallidus, to posterior nucleus of the thalamus. In addition to receiving similar dopaminergic innervation from substantia nigra pars compacta (SnC), such loops are famously thought to be homologous to one another.*

Such models provide a neural implementation of Thorndike's early law of effect, the reinforcement principle according to which rewarded actions (here, those paired with positive prediction error) tend to be repeated [29]. However, the hypothesis that animals or humans rely exclusively on this principle to make decisions has long been known to be false, as demonstrated by a line of learning experiments whose basic logic traces to rodent spatial navigation experiments by Tolman [22] (for modern variants, see [6,27,30–32]).

To facilitate simulation and analysis, we here frame the logic of these experiments in terms of "grid-world" spatial MDPs. Framed this way, such experiments can be divided into two categories. These require subjects to adjust to two different sorts of local changes in the underlying MDP. Experience with these changes is staged so as to reveal whether they are relying on cached values or recomputing them from a representation of the full MDP.

Accordingly, revaluation tasks, such as latent learning and sensory preconditioning, examine whether animals appropriately adjust behavior following changes in $R(s)$, such as a newly introduced reward (Fig 2a). Analogously, contingency change (e.g., detour or contingency degradation) tasks examine whether animals appropriately adjust behavior following changes in $P(s'|s,a)$, such as a blocked passageway (Fig 2b). Model-free RL is insensitive to these manipulations because it caches cumulative expected rewards and requires additional learning to update the stored V. Conversely, model-based algorithms, which use the one-step model directly to compute V at decision time, react flexibly to any experience that affects it. That animals can flexibly adapt their behavior following such manipulations has been interpreted as evidence for their use of model-based RL [1].

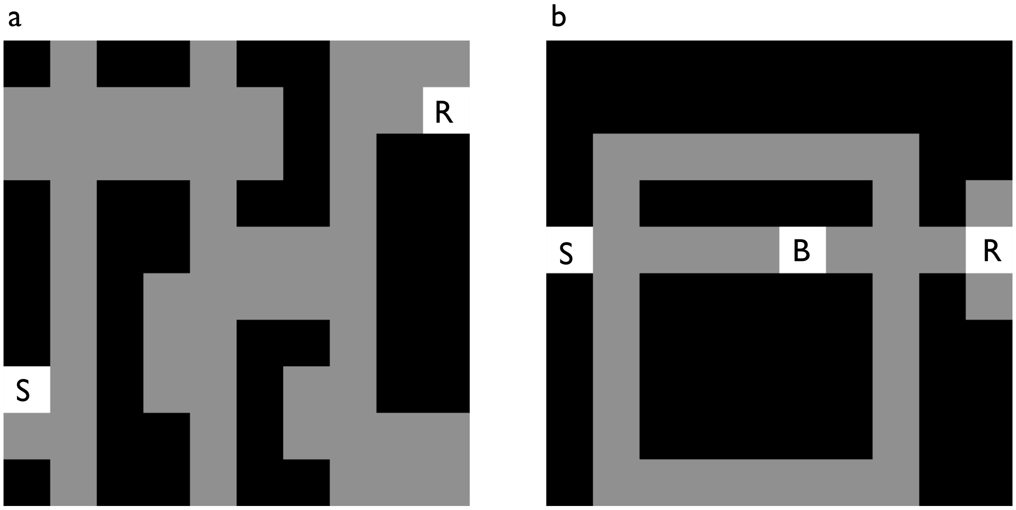

Fig. 2. Grid-world representation of Tolman's tasks*. Dark grey positions represent maze boundaries. Light grey positions represent maze hallways. a) Latent learning: following a period of randomly exploring the maze (starting from S) the agent is notified that reward has been placed in position R. We examine whether the agent's policy immediately updates to reflect the shortest path from S to R. b) Detour: after the agent learns to use the shortest path to reach a reward state R from state S, a barrier is placed in state B. After the agent is notified that state B is no longer accessible from its neighboring state, we examine whether its policy immediately updates to reflect the new shortest path to R from S.*

The puzzle of model-based learning

A further set of rodent lesion studies has used reward devaluation tasks to suggest that apparently model-based and model-free behaviors (i.e., behavior that is either flexible or insensitive following reward devaluation) depend on dissociable sets of brain areas (e.g. [5,33]). This led to the hypothesis (e.g., [1,34,35]) that these two forms of reinforcement learning depend on competing systems in the brain—the dopaminergic TD system previously described, plus a second – less clearly understood – circuit supporting model-based behavior. The accompanying presumption has been that the model-based system does not rely on a dopaminergic prediction error signal, because the prediction error of equation 3 is specifically useful for directly learning long-run cumulative values V, whereas the idea of model-based learning is to *derive* these iteratively by stringing together short-term predictions from a learned one-step model [12,36].

However, neither the rodent lesion data nor another body of work studying the neural correlates of model-based learning in humans suggests such a clean differentiation between the dopaminergic-striatal circuit (supposed to support TD) and some other presumably non-dopaminergic substrate for model-based learning. Instead, lesions suggest each type of learning is supported by a different subregion of striatum, together with connected regions of cortex (Fig 1b) and basal ganglia. This suggests that model-based and model-free systems may map onto adjacent but structurally parallel cortico-basal ganglionic loops [37], thus perhaps involving analogous (striatal) computations operating over distinct (cortical) input representations [38].

Also contrary to a strict division between systems, both dorsomedial and dorsolateral striatal territories have similar interrelationships with dopamine [39], though the involvement of their differential dopaminergic input in devaluation sensitivity has not been completely assessed [40]. Research on humans' learning in a two-step MDP (which has similar logic to devaluation studies) supports the causal involvement of dopamine in model-based learning [7–10]. Furthermore, dopaminergic recordings in rodents [11] (though see [37]), and neuroimaging of prediction error signals in human striatum [6] suggest that these signals integrate model-based evaluations.

Altogether, this research supports the idea that model-based evaluations are at least partly supported by the same sort of dopaminergic-striatal circuit thought to support TD learning, though perhaps operating in separate cortico-striatal loops. This suggestion, if true, provides strong hints about the neural basis of model-based behavior. However, it also seems puzzlingly inconsistent with the textbook [25] picture of model-based learning.

First, as mentioned previously, the internal variables used by typical model-based approaches are different from the TD value function, and require different update signals than the TD prediction error that dopamine is thought to represent. Specifically, following an experienced transition, the TD prediction error is used to update a cached estimate of the long-run value function, $V(s)$. In contrast, typical model-based approaches do not maintain or directly update long-run value estimates between decisions. Instead, following experienced transitions, such approaches will update an estimate of the one-step transition function, $P(s'|s,a)$ as well as the immediate reward function $R(s)$. The update signals required to learn these quantities are markedly different from the dopaminergic TD error. For instance, unlike the TD prediction error and the well-demonstrated phasic dopaminergic response [41], an update signal for learning $R(s)$ would not be active following presentation of a cue that reliably predicts future rewards. What then is the role then of dopaminergic activity, directed at dorsomedial striatum, for model-based learning?

Additionally, the computations required to obtain the key decision variable (the long-run state-action value) at decision time for TD and typical model-based approaches are quite different. Whereas TD approaches can simply retrieve the stored long-term value associated with a state (or, in the case of linear approximation TD, take the inner product of a vector-valued state representation and learned weights), typical approaches to model-based learning utilize some iterative computation like value iteration at choice time to aggregate one-step predictions into a long-run value prediction. How is it that dopaminergic-striatal circuitry thought to perform TD learning could participate such structurally different computations for both acquisition and retrieval? The modeling in the remainder of this article explores directions for resolving this mismatch.

A framework for solution: the successor representation

The research reviewed above suggests that model-based RL may be accomplished using computations that are homologous to those used in model-free RL. How can this be? In fact, it is known that evaluations with some features of model-based learning can result from TD learning over a different input representation. As shown by Dayan [15], equation 1 can reformulated as:

$$V^{\pi}(s) = \sum_{s'} M^{\pi}(s,s')R(s') \tag{5}$$

Here, M^{π} is a matrix of expected cumulative discounted *future state occupancies*, measuring the cumulative time expected to be spent in each future state s', if one were to start in some state s and follow policy π (Fig 3):

$$M^{\pi}(s,s') = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^{t} \mathbb{I}(s_t = s') \; | s_0 = s], \tag{6}$$

where $\mathbb{I}(\cdot) = 1$ if its argument is true and 0 otherwise. Thus, this form rearranges the expectation over future trajectories in equation 1 by first computing expected occupancies for each state, then summing rewards obtained in each state over these.

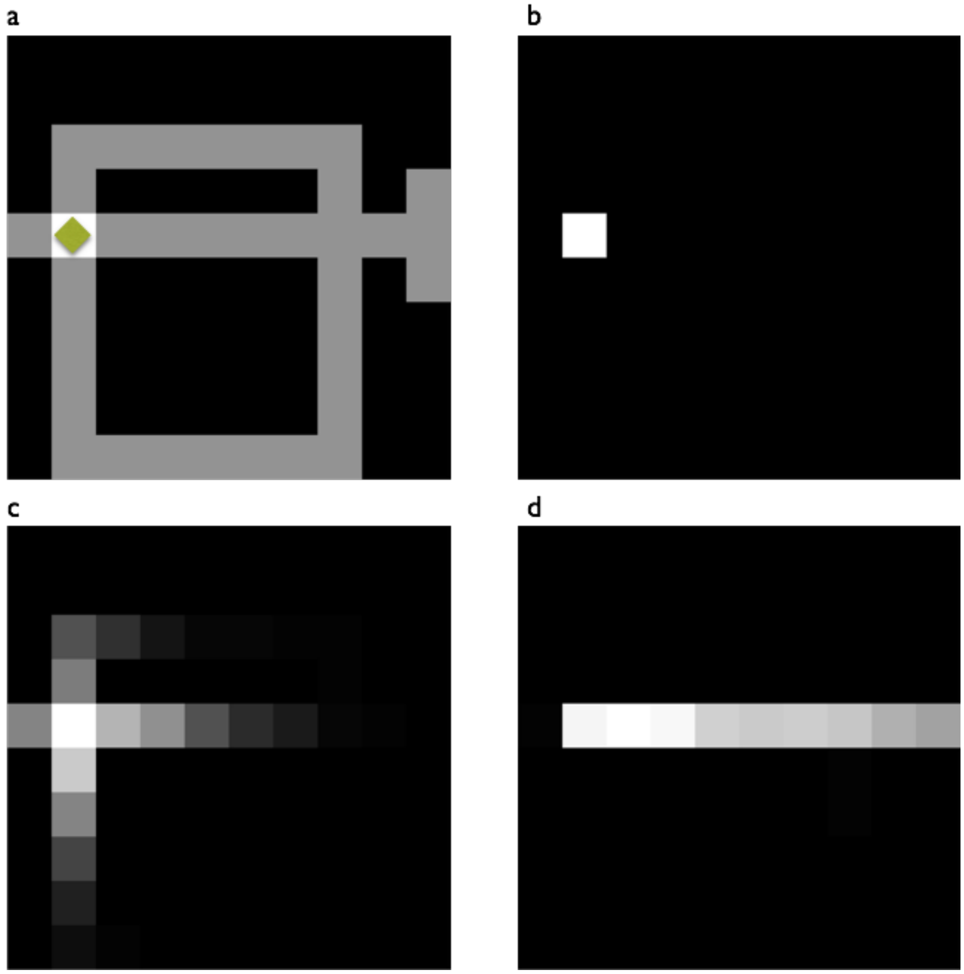

***Fig. 3. Example state representations**. a) Agent position (diamond) in a maze whose hallways are indicated by grey. b) Punctate representation of the agent's current state. Model-free behavior results from TD computation applied to this representation c,d) Possible successor representations of agent's state. Model-based behavior may result from TD applied to this type of representation. The successor representation depends on the action selection policy the agent is expected to follow in future states. The figures show the representation of the current state under a random policy (c) versus a policy favoring rightward moves (d).*

M^{π} can also be used as a set of basis functions for TD learning of values. Specifically, we represent each state using a vector of features given by the corresponding row of M (Fig 3), i.e. by the future occupancies expected for each state s'. Then we approximate $V^{\pi}(s)$ by some weighted combination of these features:

$$V^{\pi}(s) = \sum_{s'} M^{\pi}(s,s')w(s') \tag{7}$$

Comparing equations 5 and 7 demonstrates this approximation will be correct when the weight $w(s')$ for each successor state corresponds to its one-step reward $R(s')$. One way to learn these weights is using standard TD learning (adapted for linear function approximation rather than the special case of a punctate state representation). In particular, following a transition $s \rightarrow s'$, each index i of $\boldsymbol{w}$ is updated:

$$w(i) \leftarrow w(i) + \alpha_{TD}\delta M^{\pi}(s, i) \tag{8}$$

Here, δ is defined as in equation 4.

Altogether, this algorithm suggests a strategy for using different inputs into a common dopaminergic/TD learning stage to produce different sorts of value predictions (see also [16]). In particular, whereas model-free valuation may arise from TD mapping of a punctate representation of the current state (Fig 3b) in sensory and motor cortex to values in dorsolateral striatum (Fig 1b), at least some aspects of model-based valuation may arise by analogous TD mapping of the successor representation (Fig 3c,d) in prefrontal cortex or hippocampus to values in dorsomedial striatum (Fig 1b). This is possible because the successor matrix M has a predictive aspect reflecting knowledge of the state transitions $P(s'|s,a)$, at least in terms of aggregate occupancy, separate from the states' rewards $R(s)$.

Next we explore the consequences of this architecture for typical behavioral tests of model-based evaluation, and in particular diagnose why it is only capable of explaining behavior on a subset of these tests.

Strategies for learning the successor representation

To simulate learning using the successor representation, we need to also simulate how the successor matrix M^{π} is itself produced from experience. M^{π} can be defined through a recursive equation that is directly analogous to equations 1 and 2:

$$M^{\pi}(s,:) = \mathbf{1}_s + \gamma \ \sum_{s'} T^{\pi}(s,s')M^{\pi}(s',:), \tag{9}$$

where $\mathbf{1}_s$ is the vector of all zeros except for a 1 in the sth position and T^{π} is the one-step state transition matrix that is dependent on π, $T^{\pi}(s,s') = \sum_a \pi(a|s)\, P(s'|s,a)$. Similar to how approaches to estimating V are derived from equations 1 and 2, one could derive analogous approaches to estimating M^{π} from equation 9. Specifically, one could utilize a "model-based" approach that would learn T^{π} and use it to derive a solution for M^{π}. Alternatively, a TD learning approach could be taken to learn M^{π} directly, without use of a one-step model T^{π}. (This approach is analogous to model-free TD methods for learning V, though it is arguably not really model-free since $M^{\pi}$is itself a sort of long-run transition model.) This TD learning approach would cache rows of M and update them after transitioning from their corresponding states, by moving the cached row closer to a one-sample estimate of the right hand side of equation 9. Lastly, such TD updates could also occur offline, using simulated or previously experienced samples. This approach for learning M^{π}would be comparable to the Dyna approach for learning V. In the following sections, we explore the behavioral consequences of each these strategies.

Algorithm 1: The original successor representation (SR-TD)

The original SR [15] (which we call SR-TD) constructed M^{π} using a TD learning approach. This approach caches rows of $M^{\pi}$and incrementally updates them after transitioning from their corresponding states. Specifically, following each state transition $s \rightarrow s'$ each element of row s is updated as follows:

$$M^{\pi}(s,:) \leftarrow M^{\pi}(s,:) + \alpha_{SR}[\mathbf{1}_s + \gamma M^{\pi}(s',:) - M^{\pi}(s,:)], \qquad (10)$$

where $\mathbf{1}_s$ is the vector of all zeros except for a 1 in the sth position. To simulate SR-TD, we have the agent learn $M^{\pi}$and w in parallel, updating each (according to equations 5 and 10) at each transition; and sample actions according to an ϵ-greedy policy (see Methods).

SR-TD can solve some reward revaluation tasks. SR-TD is able to produce behavior analogous to model-based learning in some reward revaluation tasks that defeat simple TD learning. To demonstrate this, we simulated the behavior of SR-TD in a grid-world version of Tolman's *latent learning* task (Fig 2a). The agent first explores the grid-world randomly. Next, it learns that reward is available at position R (importantly, by being placed repeatedly at R and receiving reward, but not experiencing trajectories leading there from any other location). Finally, in a test probe starting at location S, the agent's ability to transfer this training into a new behavioral policy is measured.

Fig 4a shows SR-TD performance on a latent learning task: SR-TD can, without further learning, produce a new policy reflecting the shortest path to the rewarded location (Fig 4a). In contrast, a standard model-free TD learner operating with a non-predictive state representation cannot solve latent learning problems: it would learn nothing about paths around the maze from the reward training, and would have to discover the path to the new reward from scratch by additional exploration (simulations not shown). For similar reasons, SR-TD can solve sensory preconditioning (e.g. [32]) and reward devaluation tasks (e.g. [6,27,30,31]), both of which turn on an analogous ability to update behavior when state transition probabilities are held constant but reward values are changed.

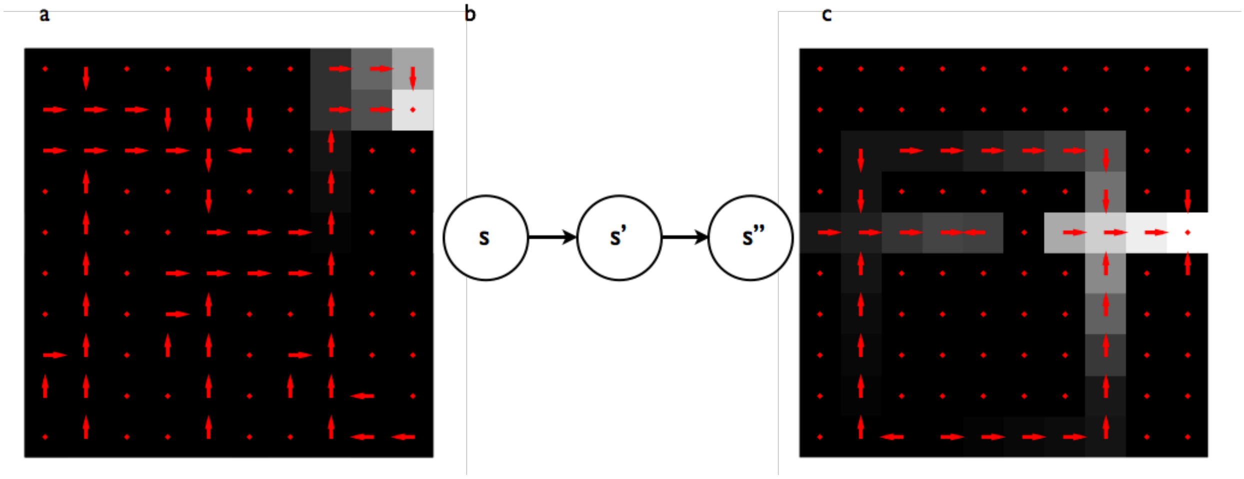

Fig. 4. Behavior of SR-TD*: a) SR-TD can solve some latent learning tasks. Median value function (grayscale) and implied policy (arrows) are shown immediately after SR-TD learns about reward in latent learning task. b) SR-TD can only update predicted future state occupancies following direct experience with states and their multi-step successors. For instance, if SR-TD were to learn that s'' no longer follows s', it would not be able to infer that state s'' no longer follows state s. Whether animals make this sort of inference is tested in the detour task. c) SR-TD cannot solve detour problems. Median value function (grayscale) and implied policy (arrows) are shown after SR-TD encounters barrier in detour task. SR-TD fails to update decision policy to reflect the new shortest path.*

SR-TD cannot solve transition revaluation tasks. However, SR-TD is limited in its ability to react correctly to other seemingly similar manipulations. Because M^{π} reflects long-run cumulative state occupancies, rather than the individual one-step transition distribution, $P(s'|s,a)$, SR-TD cannot adjust its valuations to local changes in the transitions without first updating M^{π}at different locations. This inflexibility prevents SR-TD from flexibly adjusting value estimates after learning about changes in transition structure ("transition revaluation"; Fig 4b). Consider a grid-world version of Tolman's detour task (Figs 2b and 4c). Here, the agent is first trained to seek reward at R, starting from S. Later, a blockade is introduced at B. Again, the agent is allowed to experience this change only *locally*, by learning about the consequences of trying to go right at the spot immediately to the left of the blockade. The states to the right are no longer accessible from this state, but this change is not reflected in M^{π} for the start state. From equation 10, it can be seen that this update can only occur from direct experience, i.e., a series of new trajectories starting at S and encountering the barricade. Thus, SR-TD fails to reduce the value of states that previously led to this path (Fig 4c), and would approach the barricade rather than taking a detour on the first visit back to S. Because there is at least some evidence from the early literature [42] that animals can adapt correctly to detour situations, we suggest that this inflexibility prevents SR-TD from being a plausible mechanism for the full repertoire of model-based behavior.

Algorithm 2: Dynamic recomputation of the successor representation (SR-MB)

Here, we explore a novel "model-based" approach, SR-MB, for constructing M^{π}. SR-MB learns a one-step transition model, T^{π} and uses it to derive a solution to equation 9. One key constraint on a model-based implementation suggested by the data is that the computation should be staged in a way consistent with the architecture suggested by Fig 2a. Specifically, the TD architecture in Fig 2a suggests that, because the states are represented in cortex and weights (which capture information about rewards) and value are represented in downstream cortico-striatal synapses and medium spiny striatal neurons, information about $R(s)$ and $V(s)$ should not be used in the online construction of states. For the SR approach, this implies that M be constructed without using direct knowledge of $R(s)$ or $V(s)$. As we see below, this serial architecture – a cortical state-prediction stage providing input for a subcortical reward-prediction stage – if true, would impose interesting limitations on the resulting behavior.

To construct M^{π}, SR-MB first learns the one-step state transition matrix T^{π}, implemented in our simulations through separate learning of $P(s'|s,a)$ as well as $\pi(a|s)$, the agent's previously expressed decision policy (see Methods). Prior to each decision, $T^{\pi}$is used to compute a solution to equation 9:

$$M^{\pi} = (I - \gamma T^{\pi})^{-1} \tag{11}$$

Given M^{π} and w, SR-MB forms V and Q values in the same way as SR-TD.

SR-MB can solve transition revaluation tasks. Using an updated T^{π} to recompute M^{π} at decision time ensures that behavior is sensitive to changes in the transition structure. We demonstrate this by showing that unlike SR-TD, SR-MB successfully solves Tolman's detour task in addition to latent learning. In the detour task, after discovering the barrier, SR-MB immediately adjusts its policy to the new shortest path (Fig 5a). SR-MB thus produces the two behaviors that are considered signatures of model-based RL: immediate adjustment of decision policy following learned changes in either reward structure (latent learning) or transition structure (detour problem). However, unlike typical model-based approaches, it does so by utilizing TD to map a learned state representation to value estimates and thus provides solution for how model-based behaviors could arise from the architecture in Fig 2a.

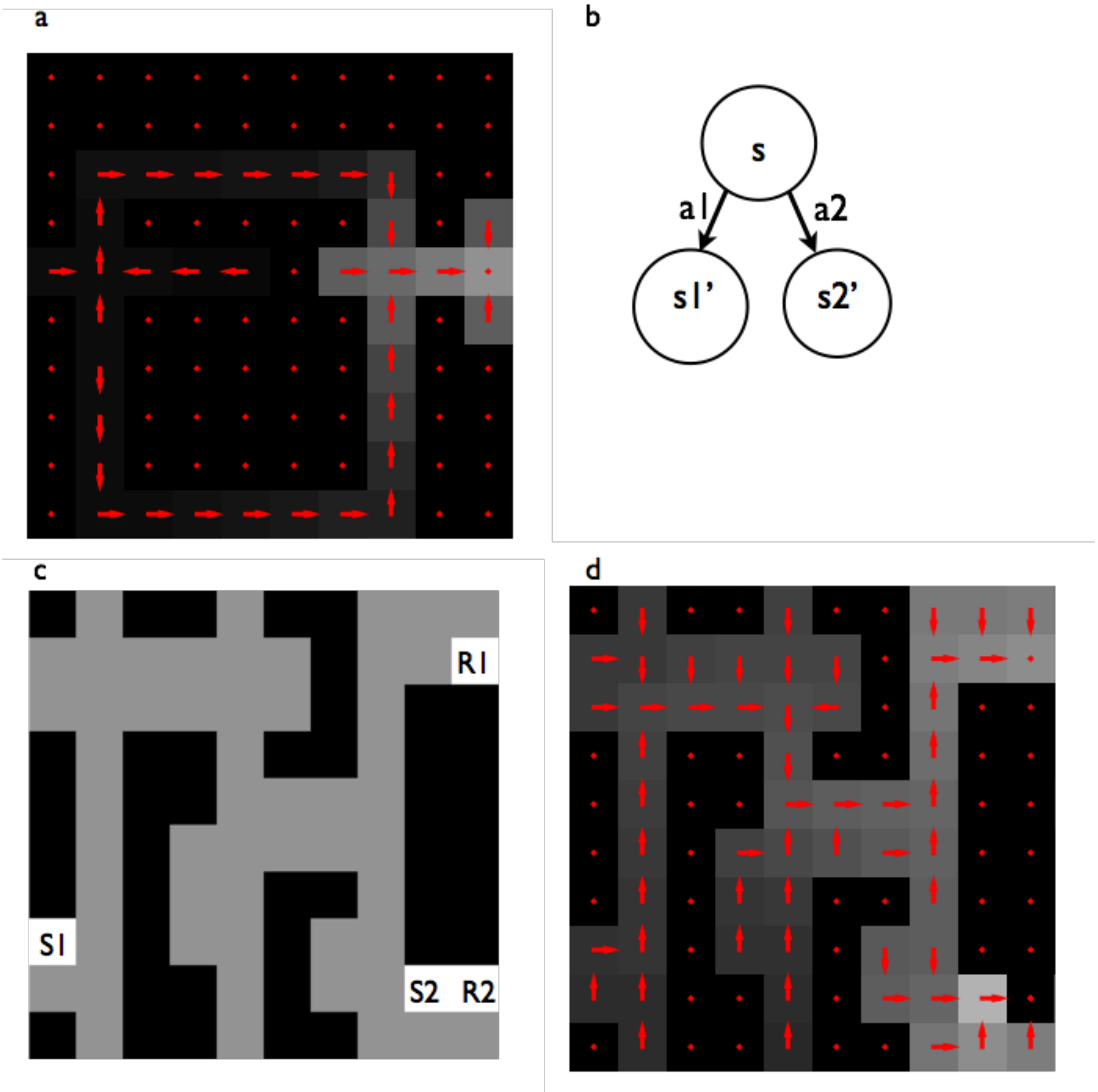

Fig. 5. Behavior of SR-MB*. a) SR-MB can solve the detour task. Median value function (grayscale) and implied policy (arrows) after SR-MB encounters barrier. b) SR-MB determines successor states relative to a cached policy. If SR-MB learned from previous behavior that it will always select action a1, the value of s would become insensitive to changes in reward at s2'. C) Novel "policy" revaluation task. After a phase of random exploration, we place a reward in location R1. The agent completes a series of trials that alternatively start from locations S1 and S2 and end when R1 is reached. We then place a larger reward in location R2 and record the agent's value function and implied policy upon encountering it. d) SR-MB cannot solve the novel "policy" revaluation task. Median value function and implied policy recorded immediately after SR-MB learns about reward placed in location R2. Notice that if the agent were to start from location S1, its policy would suboptimally lead it to the smaller reward at R1.*

SR-MB is limited by policy dependence. Given that SR-MB reproduces previously examined animal behavior in planning tasks, we next sought to determine whether there were any tasks, perhaps not yet explored in the empirical literature, that could differentiate it from approaches that utilize "full" model-based value iteration. A key feature of SR-MB, as well as SR-TD, is that it computes M^{π} with respect to a policy π. For SR-MB, T^{π} is computed using $\pi(a|s)$, which is learned through observation of previous actions. Because M^{π} is policy dependent, so are the value estimates that it produces. SR-TD and SR-MB are thus "on-policy" methods – their estimates of V^{π} can be compared to the estimates of a traditional model-based approach used to solve equation 1. In fact, if weights, $w(s)$ are taken to have reached asymptotic values $R(s)$, then SR-MB is algebraically equivalent to doing standard model-based value iteration to solve equation 1.

The key feature of this strategy is its dependence on the exploration policy π. A natural question is whether a similar approach could be used to define M^* – the successor states expected under the *optimal* policy – which would, in turn provide the state input for solving for V^*, the optimal long-run values. This is an important distinction, as we will see below, because new learning about parts of the MDP can moot the previously expressed policy π (and hence invalidate the associated successor matrix and values). But this problem is not simple to fix, given the architectural constraints we have suggested. In particular, defining equation 9 with respect to the optimal policy would require replacing T^{π} with T*: the one-step transition matrix corresponding to the optimal policy, $T^*(s, s') = P(s'|s, a^*)$, where a^* is the action in state s that maximizes future rewards. Computing a^* online, however, would require access to $R(s)$, which would violate the suggested serial staging of the computation, that M be constructed without using reward information.

Policy dependence thus defines both the major limitation as well as the empirical signature of SR-MB. Unlike standard value iteration, separating the state prediction and reward prediction stages of the algorithm serially implies that the algorithm must plan using values that are derived with respect to the policy that prevailed during learning.

SR-MB cannot solve novel policy revaluation tasks. What are the behavioral implications of planning using $\boldsymbol{V^{\pi}}$ rather than $\boldsymbol{V}^{*}$? Consider a situation where state $\boldsymbol{s'}$ can be reached from state $\boldsymbol{s}$ using action $\boldsymbol{a}$, but SR-MB learned from past behavior that $\boldsymbol{\pi(a|s)}$ is near 0 (Fig 5b). Then it will not include rewards at $\boldsymbol{s'}$ in the value of $\boldsymbol{s}$, even if separately learning (say, by visiting $\boldsymbol{s'}$ but not in a trajectory starting from $\boldsymbol{s}$) that $\boldsymbol{s'}$ is the most rewarding state reachable from $\boldsymbol{s}$. In other words, caching of the policy at $\boldsymbol{s}$ blinds the agent (without further exploration and relearning) to changes in the reward function that should change that policy. Value iteration based on equation 2 does not have this limitation because the max operation would look ahead to the reward at $\boldsymbol{s'}$ to determine whether it should be included.

These considerations suggest a novel revaluation task (Fig 5c). Here, SR-MB first performs many trials where R1 is rewarded, so that it learns a policy that takes it away from the bottom right corner of the maze. Next, a larger reward is introduced at R2. Despite having learned about this reward (by starting at S2), the agent would choose to head toward R1 from S1, due to caching of the incorrect policy along the way (Fig 5d). This task demonstrates that SR-MB produces errors that could in principle be behaviorally detectable, but have not been exercised by standard experimental tasks.

Given the empirical plausibility of SR-MB, we can consider how the computation in equation 11 may be actually carried out. One potential mechanism is recurrent neural networks. It can be shown that M^{π} is the attractor state of a network with one node for each state, and connection weight matrix γT^{π} (see [17]). This essentially corresponds to value iteration on M. An alternative would involve using sample-based backups – analogous to Sutton's Dyna architecture [24], but using samples (s, a, s') with equation 10 to update cached rows of M^{π} rather than the value function V. As for values, such samples could be either directly replayed from stored experience episodes, or generated from a transition model T^{π}. Given sufficient offline replay prior to each decision (and because of the basic analogy between sample-based and value iteration-based solutions to equation 9), this mechanism essentially amounts to a different implementation of the on-line construction of M^{π} discussed here, with similar advantages and disadvantages. However, in the next section, we explore how this approach, when combined with learning about the state-action value function Q (rather than the state value function V), can be used to solve the off-policy planning problem.

Algorithm 3: Off-policy experience resampling (SR-Dyna)

Here we introduce a third approach towards solving equation 9, SR-Dyna, which can be compared to Sutton's Dyna approach [24] for solving equations 1 and 2. Akin to how Dyna replays experienced transitions offline to update estimates of $V(s)$, SR-Dyna replays experienced transitions to update the successor matrix. When this approach is combined with an 'off-policy' update rule, similar to Q learning, to update the successor matrix offline, it is capable of solving the off-policy planning problem. Utilizing this type of update however, requires us to work with a state-action version of the successor representation, H, which can be used directly to form Q values [43,44]. The key idea here is to define future occupancy not over states but over state/action pairs, which we denote sa. Analogous to equation 3, Q^{π} can then be expressed:

$$Q^{\pi}(sa) = \sum_{s'a'} H(sa, s'a')\, R(s'a'), \tag{12}$$

H is a matrix of expected cumulative discounted future state-action visitations, i.e. given that you are starting with state s and action a, the cumulative (discounted) expected number of times you will encounter each other state/action pair:

$$H(sa, s'a') = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^{t} \mathbb{I}(sa_t = s'a') | sa_0 = sa]. \tag{13}$$

H can then be used as a linear basis for learning $Q(s, a)$, using the SARSA TD algorithm to learn a weight for each column of H. In particular, when state-action $s'a'$ is performed after state action sa, a prediction error is calculated and used to update w:

$$\delta = R(sa) + \gamma Q(s'a') - Q(sa) \tag{14}$$

$$w(i) \leftarrow w(i) + \alpha_{TD} \delta H(sa, i), \text{ for all } i$$

Like M, H can be defined recursively:

$$H(s, :) = \mathbf{1}_{sa} + \gamma \sum_{s'} T^{\pi}(sa, s'a') H(s'a', :)$$

where T^{π} is the one-step state-action transition matrix, $T^{\pi}(sa, s'a') = \sum_{s'} \sum_{a'} P(s'|s, a)\, \pi(a'|s')$. As with SR-TD, this recursion can be used to derive a TD-like update rule by which an estimate of H can be iteratively updated:

$$H(sa,\cdot) \leftarrow H(sa,\cdot) + \alpha_{SR}[\mathbf{1}_{sa} + \gamma H(s'a',\cdot) - H(sa,\cdot)] \tag{15}$$

SR-Dyna can utilize off-policy updates. As with SR-MB, it is also possible to derive H from $T^{\pi}(sa, s'a')$ using an explicit "model-based" solution analogous to equation 9. However, here, we investigate the approach of updating H off-line (e.g., between trials or during rest periods) using replay of experienced trajectories. The key assumption we make is that this off-line replay can implicate both the state and reward (cortical and basal ganglia) stages of Figure 1b, giving rise to an off-policy update of H with respect to the policy π^* that is optimal given the current rewards. By comparison, as articulated above, we assumed such policy maximization was not possible when computing the successor representation M on-line for SR-MB, since this entire computation was supposed to happen in cortex at decision time, upstream of the striatal reward learning stage.

Following each transition, SR-Dyna stores the sample (s, a, s'). Then in between decisions, SR-Dyna randomly selects (with a recency weighted bias) k samples (with replacement). For each sample, it updates H as follows:

$$H(sa,\cdot) \leftarrow H(sa,\cdot) + \alpha_{SR}[\mathbf{1}_{sa} + \gamma H(s'a'^{*},\cdot) - H(sa,\cdot)] \qquad (16)$$

where

$$a'^{*} = argmax_{a'} \sum_{s''a''} H(s'a', s''a'')\, w(s''a'')$$

That is, when H updates from previously experienced samples, it performs an off-policy update using the best action it could have chosen, rather than the one it actually chose.

SR-Dyna can solve policy revaluation tasks. Given sufficient sampling (large enough k), this off-policy updating not only permits SR-Dyna to solve the detour task (Fig 6a), but also to solve the novel policy revaluation task (Fig 6b). SR-Dyna can thus produce behavior identical to "full" model-based value iteration. However, it has the potential advantage that updating can take place fully off-line and thus offload computation to situations that may be ecologically convenient such as sleep or wakeful rest.

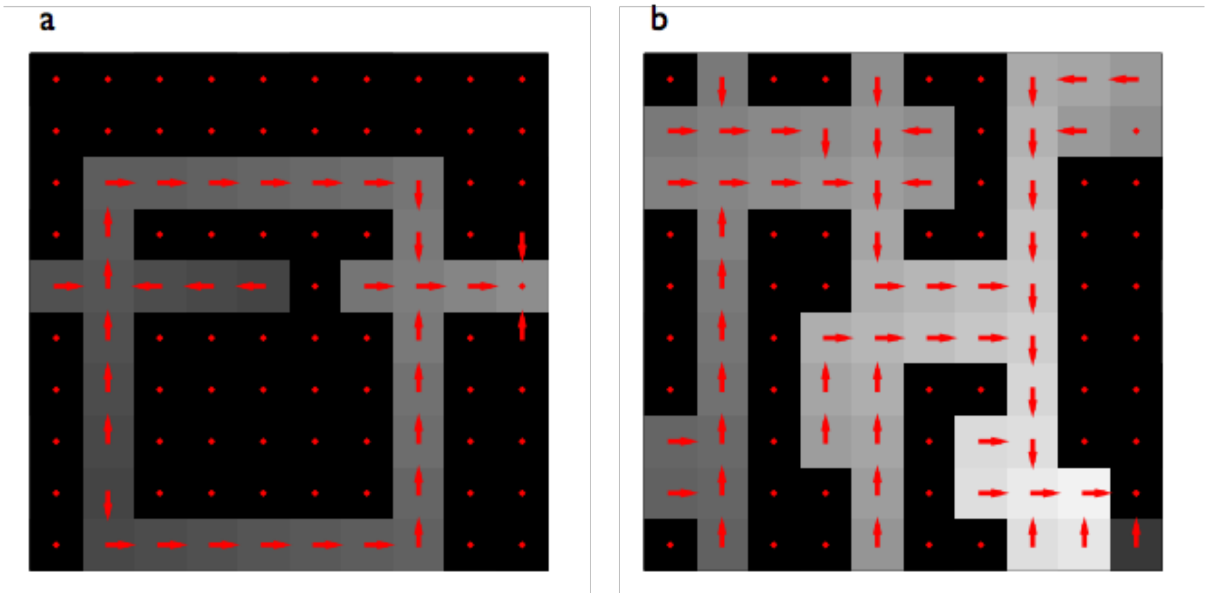

***Fig. 6. Behavior of SR-Dyna**. a) SR-Dyna can solve the detour task. Median value function (grayscale) and implied policy (arrows) after SR-DYNA encounters the barrier. b) SR-Dyna can solve the novel "policy" revaluation task. Median value function (grayscale) and implied policy (arrows) after SR-Dyna learns about the new larger reward.*

Discussion

Despite evidence that animals engage in behaviors suggestive of model-based planning, we have little knowledge of how these computations are actually performed in the brain. Indeed, what evidence we have – particularly concerning the involvement of dopamine in these computations – seems difficult to reconcile with the standard computational picture of planning by tree search using a learned model. We have here proposed variants of the SR that can address this question, serving as empirically consistent mechanisms for some or indeed all of the behaviors associated with model-based learning. Moreover, these are each built as utilizing a common TD learning stage for reward expectancies, allowing them to fit within the systems-level picture suggested by rodent lesion studies, and also explaining the involvement of dopamine in model-based valuation. In particular, they each envision how model-based learning could arise from the same dopaminergic TD learning associated with simple model-free learning, operating over a different and more elaborated cortical input representation.

We have focused our simulations on choice behavior, and have not presented our theories' analogous predictions about the responses of neurons, such as DA cells, thought to signal decision variables. However, whenever the SR algorithms' expectations about action values incorporate "model-based" information (such as latent learning, Fig 4a) neural signals related to those predictions and to prediction errors would be similarly informed. Thus the theories predict systematic expectancy-related effects in the modeled dopamine response, tracking the differences in choice preference relative to the standard "model-free" accounts, which are blind to reward contingencies in these tasks. This observation offers an explanation for several reports that the phasic DA response (or analogous prediction-error related BOLD signals in humans) tracks apparently model-based information [6,11]. Consider the sensory preconditioning task (the Pavlovian analogue of latent learning). Here, following a pre-conditioning phase in which stimulus B is repeatedly presented as following stimulus A, a conditioning phase occurs in which stimulus B is paired with reward. Since, at this point in the experiment, A has neither been followed by reward nor by a stimulus itself previously paired with reward, pure model-free approaches estimate the value of A, $V(A)$ to be 0. They thus predict that presenting A to the animal at this point in the experiment, t_A, would not elicit a dopaminergic response (e.g. $\delta(t_A) = R(t_A) + V(t_A) - \gamma V(t_A - 1) = V(t_A) = V(A) = 0$). In contrast recent experiments have observed that presenting A elicits a significant phasic dopaminergic response [11]. The SR can account for this because once B is paired with reward, A instantly gains value because it predicts future occupancy of B. Thus, the reward prediction error considered in equation 8, which uses SR estimates of $V(A)$, would be greater than 0 (see also [20]).

Situations like this, in which prediction errors constructed using either SR or model-free value estimates diverge, raise interesting questions for the architecture in Fig 1. b. If dorsomedial and dorsolateral striatum are to respectively represent model-based and model-free values simultaneously, then this seems to imply two separate prediction errors: one containing SR value estimates, directed at dorsomedial striatum, and one containing model-free value estimates directed at dorsolateral striatum. There is in fact recently mounting evidence that dopaminergic neurons project different signals to different subareas of striatum, though this work has so far focused on the dorsal-ventral axis (and on states vs. actions) rather than medial-lateral [45,46]. Future work should explore whether separate information may also be projected to the dorsomedial versus dorsolateral areas. Another possibility, however, is that at any given time, net value predictions are derived by some weighted combination of punctate (model-free) and predictive (SR) state representations, which would then drive both choices and a single net reward prediction error. Changes in the differential reliance on these inputs, over time, would then drive changes in behavioral strategy, and differences in the accompanying dopamine response. Indeed, there is some evidence that as rodents gradually alter their behavioral strategy from model-based to model-free learning with overtraining, activity in dorsomedial striatum becomes less structured while activity in dorsolateral striatum becomes more structured, perhaps suggesting that at any given point in time only one of these areas is carrying sufficient information to construct either SR or model-free value estimates [47].

Although our presentation culminated with proposing an algorithm (SR-Dyna) that can in principle perform equivalently to full model-based learning using value iteration, this need not be the only goal and there need not be only a single answer. The behaviors associated with model-based learning may not have unitary sources in the brain but may instead be multiply determined, and all of the algorithms we have considered are viable candidate pieces of a larger set of decision systems. Notably, the experiments we have highlighted as suggesting striatal or dopaminergic involvement in "model-based" learning and inspiring the present work all use extremely shallow planning problems (e.g. operant lever pressing, two-stimulus Pavlovian sequences, or two-step MDPs) together with reward revaluation designs. Even SR-TD is sufficient to explain these. It may well be that planning in other tasks, like chess, or in spatial mazes, is supported by entirely different circuits that really do implement something like tree search; or that they differentially require replay, like SR-Dyna. Also, although replay-based approaches go a long way, value computation at choice time using more traditional model-based approaches is likely needed at the very least to explain the ability to evaluate truly novel options (like the value of "tea jelly"; [48]) using semantic knowledge.

Relatedly, if the brain might cache both endpoint decision variables like V, or their precursors like M, update either or both with off-line replay, and optionally engage in further model-based recomputation at choice time, then the arbitration or control question of how the brain prioritizes all this computation to harvest rewards most effectively and efficiently becomes substantially more complicated than previously considered. The prioritization of replay – which memories to replay when – becomes particularly important. Still, principles of efficient cost-benefit management might apply, extending those that have been considered for simpler, binary model-based vs. model-free tradeoffs [1,35,49] .

It is also worth considering the evidence against the simplest approaches. The main counterexamples to SR-TD are transition revaluation and detour tasks. Apart from the classic work of Tolman and Honzik [42], the original results of which are actually quite mixed (see [50]), there is surprisingly little evidence to go on. A number of different studies have shown that healthy animals will normally choose the shortest alternative route after learning about a blockade preventing a previously preferred route (e.g. [51–53]). However, in these studies, the animal learns about the blockade after starting from the maze starting location. Thus, unlike in our simulations in which the animal learns about the blockade in isolation, animals in these tasks would have the opportunity to learn from direct experience that maze locations leading up to the blockade are no longer followed by maze locations further along the previously preferred path. Such tasks could thus potentially be solved by SR-TD. Studies that show that animals will take a shortcut to a goal that is discovered along a preferred path present a somewhat cleaner test for SR-TD [54,55]; however it is often difficult to interpret a potential role of exploration or visual (rather than cognitive map) guidance in the resulting behavior. Work in humans, however, seems to more clearly suggest an ability to solve detour tasks without re-learning [56]. Simon and Daw [23] for instance directly assessed SR-TD's fit to human subjects' choice adjustments in a changing spatial maze, and found it fit poorly relative to traditional model-based learning.

Overall, additional careful work that measures how animals respond to transition changes, learned in isolation, is needed. Whereas Tolman's other early reward revaluation experiments (latent learning) have been conceptually replicated in many modern, non-spatial tasks like instrumental reward devaluation and sensory preconditioning, the same is not true of detours. Indeed, the modern operant task that is often presented as analogous to detours, so-called instrumental contingency degradation (e.g., [57]), is not functionally equivalent. In such tasks, the association between an action and its outcome is degraded through introduction of background rewards. However, because the information about the changed contingency is not presented separately from the rest of the experience about actions and their rewards, unlike all the other tests discussed here, contingency degradation can be solved by a simple model-free learner that re-learns the new action values. The puzzle here is actually not how animals can solve the task, but why they should ever fail to solve it. This has thus led to a critique not of model-based but of model-free learning theories [38].

In any case, the modeling considerations proposed here suggest that more careful laboratory work on "transition revaluation" type changes to detect use of SR-TD, is warranted. Similarly, "policy revaluations" along the lines of that in Fig 5 would be useful to detect to what extent planning along the lines of SR-MB is contributing. Finally, although SR-Dyna in principle can perform model-based value computation, this depends on sufficient replay. The SR-Dyna hypothesis suggests the testable prediction that behavior should degrade to SR-TD under conditions when replay can contribute less; thus manipulations of factors like rest period length or distractor tasks (see [27]) would be expected to titrate behavior between SR-TD and pure model-based computations, and thus modulate the ability to solve detour tasks or policy revaluations.

As pointed out in [27], Sutton's original Dyna algorithm – in which experience replayed offline is used to update action values V or Q directly – offers another avenue toward solving some of the puzzles motivating this work. Specifically, enough offline replay can also mimic model-based evaluation, and if such replay simply activates the same circuits used for online TD learning, then such model-based behaviors would be dopamine-dependent. This is a promising piece of the puzzle, but such heavy reliance on replay to underpin all behavioral flexibility seems unrealistic. Among our innovations here is to suggest that replay can also be used to learn and update a successor representation H, which then confers many of the other advantages of model-based learning (such as flexibility in the face of reward devaluation) without the dependence on further replay to replan. Further research (particularly dissecting exactly how behavior degrades when replay is prevented) will be needed to distinguish the contributions of these two Dyna mechanisms from each other, and from other options like SR-MB or true model-based planning, none of which are mutually exclusive.

As with Sutton's Dyna, SR-Dyna dovetails nicely with neural evidence about memory replay. Specifically, the subsequent reactivation of patterns of hippocampal activity that characterize a prior experience, seem well suited to support the sort of off-line updates imagined by both Dyna approaches. However, the SR-Dyna approach, in which such hippocampal replay phenomena may update a (presumably pre-frontal) successor matrix, lines up neatly with complimentary system theories in the memory literature, according to which such hippocampal replay plays a role post-encoding stabilization of memories by restructuring how information is represented across neocortical networks [58,59]. Such a connection should be explored in future research.

Although this paper has focused on the behavioral consequences of planning using various forms of the SR, another line of evidence supporting the proposal is that predictive representations seemingly consistent with the successor representation have been observed in multiple brain regions. For instance, in human neuroimaging it has been found that hippocampal representation of a fractal image becomes more similar to images that tend to follow it [60,61], and similar results have been found in prefrontal cortex [62]. It has also been suggested that hippocampal place fields themselves might encode the SR [18].

Finally, the SR may contribute to a number of other cognitive processes. Above we noted that there is evidence that areas of medial temporal lobe seem to encode predictive representations. In line with this, it has been noted that there is a close correspondence between the update rule used by SR-TD and update rules in the temporal context model of memory [19]. Also, recent approaches to reinforcement learning in the brain have advocated for a hierarchical approach in which punctate actions are supplemented by temporally abstract policies [63]. In this context, it has been suggested that the SR may be useful for discovering useful temporal abstractions by identifying bottlenecks in the state space that can then be used to organize states and action into a hierarchy [18,64]. The efficacy of the SR for model-based RL opens the possibility that the brain accomplishes planning, action chunking, and grouping episodic memories using a common mechanism.

Overall, this article has laid out a family of candidate mechanistic hypotheses for explaining the full range of behaviors typically associated with model-based learning, while connecting them with the circuitry for model-free learning as currently understood. In addition to the transition and policy revaluation behavioral experiments suggested above, future neuroimaging work could seek evidence for these hypotheses. Specifically, failures to flexibly update decision policies that are caused by caching of either the successor representation (as in SR-TD or SR-Dyna with insufficient replay) or a decision policy (as in SR-MB) should be accompanied by neural markers of non-updated future state occupancy predictions. Such neural markers could be identified using representational similarity analysis (e.g. [65]), cross-stimulus suppression (e.g. [66]) or through use of category specific, decodable, visual stimuli (e.g. [67]). Similar work in experimental animals such as rodents (e.g. [45]) could use the full range of invasive tools to trace the inputs to dorsomedial vs. dorsolateral striatum, so as to examine the information represented there and how it changes following the various sorts of revaluation manipulations discussed here. As has been the case for model-free learning, the emergence of an increasingly clear and quantitative taxonomy of different candidate algorithms is likely to guide this work and help to elucidate the neural basis of model-based learning.

Methods

General Simulation Methods

All simulations were carried out in 10x10 (N = 100 states) grid-worlds in which the agent could move in any of the four cardinal directions, unless a wall blocked such a movement. States with rewards contained a single action. Upon selecting that action, the agent received the reward and was taken to a terminal state. Each task was simulated with each algorithm 500 times. For each simulation, we recorded the agent's value function at certain points. For SR-Dyna, which worked with action values rather than state values, the state value function was computed as the max action value available in that state. Figures display the median value, for each state, over the 500 runs. To determine the implied policy for the median value function, we computed, for each state, which accessible successor state had the maximum median value.

Specific Task Procedures

Latent learning task. The latent learning task was simulated in the grid-world environment shown in Fig. 2a. Starting from position S, the agent first took 25000 steps exploring the maze. After exploration, the reward in position R1 was raised to 10. To learn about the reward, the agent completed a single step, starting from position R1, 20 times. We then recorded the state value function.

Detour task. The detour task was simulated using the grid-world environment shown in Fig. 2b. Starting from position S, the agent first took 10000 steps exploring the maze. The reward in position R was then increased to 10. The agent then completed 5 trials, starting from position S that ended when the reward was reached. A wall was then added to in position B. To learn about the wall, the agent completed a single step, starting from the position immediately left of the wall, 40 times. We then recorded the state value function.

Novel revaluation task. The novel revaluation task was simulated using the environment in Fig. 5c. The agent first completed the entire latent learning task. After successfully reaching position R1 from position S, the agent then completed 20 trials. Each trial alternately started at S1 or S2 and ended when the agent reached position R1. We then set the reward in position R2 to 20. To learn about the new reward, the agent completed one step, starting from position R2, 20 times. We then recorded the state value function.

Additional Details on Algorithms

Original successor representation (SR-TD). SR-TD computed $V(s)$ using two structures: the successor matrix, $M(s, s')$ and a weight vector, $w(s)$. At the beginning of each simulation, M was initialized as an identity matrix; however, rows corresponding to terminal states were set to $\mathbf{0}$. The weight vector was initialized as $\boldsymbol{w} = \mathbf{0}$. Following each transition, M and $\boldsymbol{w}$ were updated using equations (8) and (9). Prior to each choice, V was computed using equation (7). Q-values for each action in state a were then computed as $Q(s, a) = V(s')$ where s' is the state that deterministically follows action a in state s.

Recomputation of the successor matrix (SR-MB). This algorithm starts each task with a basic knowledge of the 'physics' of grid-world: which successor state, s', would follow each action sa in a situation in which sa is available (e.g. not blocked by a wall). It also stores and updates, for each state s, A_s, the set of actions currently available in state s as well as a policy $\pi(a|s)$, which stores the probability of selecting action a in state s (as learned from the agent's own previous choices). A_s was initialized to reflect all four cardinal actions being available in each state. Each row π were initialized as a uniform distribution over state-actions, $\pi(a|s) = 0.25$.

After performing action a in state s and transitioning to state s', $A_{s'}$ was updated to reflect which actions are available in state s' and π is updated using a delta rule:

$$\pi(:|s) \leftarrow \alpha_\pi \mathbf{1}_a + (1 - \alpha_\pi)\pi(:|s)$$

where α_π is a free parameter.

Prior to each choice, the model computed each row, s, of one-step transition matrix T as follows:

$$T(s,:) = \frac{\sum_{a \in A_S} \pi(a|s)\, \mathbf{1}_{s'}}{\sum_{a \in A_S} \pi(a|s)}$$

where $\mathbf{1}_{s'}$ is a vector zeros of length S with a 1 in position corresponding to state s' and s' is the state to which action a in state s deterministically leads. T was then used to compute M using equation 10. Computation of V and Q was then the same as in SR-TD.

Episodic replay algorithm (SR-Dyna). This algorithm computed $Q(sa)$ using two structures: a state-action successor matrix, $H(sa, s'a')$ and weight vector $w(sa)$. At the beginning of each simulation, the successor matrix H was initialized to an identity matrix; however rows corresponding to terminal states were set to $\mathbf{0}$. The weight vector was initialized to $\boldsymbol{w} = \mathbf{0}$. The algorithm also stored every sample (s, a, s'). After performing action a in state s and transitioning to state s' the sample (s', a', s') was stored, and H and **w** were updated according to equations (13) and (14). Following each step, we also selected 40 one-step samples (according to recency weighted probabilities with replacement) from the stored history, and replayed each to update H according to equation 15. Following transitions in which a learned change occurred to either the reward function or available actions, 70,000 one-step samples were selected and used to update the model. Samples were drawn by selecting an integer, corresponding to an experienced transition (where 0 corresponds to the transition that most recently occurred). Integers were selected randomly according to an exponential distribution with $\lambda = N/5$ where N is the number of transitions experienced so far in the task.

Parameters. All algorithms converted Q-values to actions using an ϵ-greedy policy which selects the highest-valued action with probability $1 - \epsilon$, and chooses randomly with probability ϵ. For all models, ϵ was set to 0.1. In addition, all models used a discount parameter $\gamma = 0.95$ and a weight learning rate paremter $\alpha_w = 0.1$. In addition to these parameters, SR-TD and SR-Dyna used a successor-matrix learning rate of $\alpha_{sr} = 0.4$. and SR-MB used a policy learning rate of $\alpha_\pi = 0.1$.

References

1. Daw ND, Niv Y, Dayan P. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. Nat Neurosci. 2005;8: 1704–1711. doi:10.1038/nn1560

2. Houk JC, Adams JL, Barto a C. A model of how the basal ganglia generates and uses neural signals that predict reinforcement. Model Inf Process Basal Ganglia. 1995; 249–270.

3. Montague PR, Dayan P, Sejnowski TJ. A framework for mesencephalic dopamine systems based on predictive Hebbian learning. J Neurosci. 1996;16: 1936–1947. doi:10.1.1.156.635

4. Frank MJ, Seeberger LC, O'Reilly RC. By Carrot or by Stick: Cognitive Reinforcement Learning in Parkinsonism. Science (80-). 2004;306: 1940–1943. doi:10.1126/science.1102941

5. Yin HH, Ostlund SB, Knowlton BJ, Balleine BW. The role of the dorsomedial striatum in instrumental conditioning. Eur J Neurosci. 2005;22: 513–23. doi:10.1111/j.1460-9568.2005.04218.x

6. Daw ND, Gershman SJ, Seymour B, Dayan P, Raymond J. Model-based influences on humans' choices and striatal prediction errors. 2011;69: 1204–1215. doi:10.1016/j.neuron.2011.02.027.Model-based

7. Wunderlich K, Smittenaar P, Dolan RJ. Dopamine Enhances Model-Based over Model-Free Choice Behavior. Neuron. 2012;75: 418–424. doi:10.1016/j.neuron.2012.03.042

8. Doll BB, Bath KG, Daw ND, Frank MJ. Variability in Dopamine Genes Dissociates Model-Based and Model-Free Reinforcement Learning. J Neurosci. 2016;36: 1211–1222. doi:10.1523/JNEUROSCI.1901-15.2016

9. Deserno L, Huys QJM, Boehme R, Buchert R, Heinze H-J, Grace AA, et al. Ventral striatal dopamine reflects behavioral and neural signatures of model-based control during sequential decision making. Proc Natl Acad Sci U S A. 2015;112: 1595–600. doi:10.1073/pnas.1417219112

10. Sharp ME, Foerde K, Daw ND, Shohamy D. Dopamine selectively remediates “model-based” reward learning: A computational approach. Brain. 2015;139: 355–364. doi:10.1093/brain/awv347

11. Sadacca BF, Jones JL, Schoenbaum G. Midbrain dopamine neurons compute inferred and cached value prediction errors in a common framework. Elife. 2016;5: 1–13. doi:10.7554/eLife.13665

12. Glascher J, Daw N, Dayan P, O’Doherty JP. States versus rewards: Dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron. Elsevier Ltd; 2010;66: 585–595. doi:10.1016/j.neuron.2010.04.016

13. Balleine BW, Daw ND, O’Doherty JP. Multiple Forms of Value Learning and the Function of Dopamine BT - Neuroeconomics: Decision Making and the Brain. Neuroeconomics Decision Making and the Brain. 2008. pp. 367–387. Available: http://books.google.com/books?hl=en&lr=&id=g0QPLzBXDEMC&oi=fnd&pg=PA367&dq=balleine+neuroeconomics&ots=i9afuLQDYl&sig=usxp3lfOydDCxVhoXJXa_IFCPLU

14. Daw ND, Dayan P. The algorithmic anatomy of model-based evaluation. Philos Trans R Soc Lond B Biol Sci. 2014;369: 20130478–. doi:10.1098/rstb.2013.0478

15. Dayan P. Improving Generalisation for Temporal Difference Learning: The Successor Representation. Neural Comput. 1993;5: 613–624.

16. Dayan P. Motivated Reinforced Learning. Adv Neural Inf Process Syst. 2002;

17. Sutton RS, Pinette B. The learning of world models by connectionist networks. Proceedings of the Seventh Annual Conference of the Cognitive Science Society. 1985. pp. 54–64.

18. Stachenfeld KL, Botvinick MM, Gershman SJ. Design Principles of the Hippocampal Cognitive Map. Adv Neural Inf Process Syst 27. 2014; 1–9. Available: http://web.mit.edu/sjgershm/www/Stachenfeld14.pdf\nhttp://papers.nips.cc/paper/5340-design-principles-of-the-hippocampal-cognitive-map

19. Gershman SJ, Moore CD, Todd MT, Norman K a., Sederberg PB. The Successor Representation and Temporal Context. Neural Comput. 2012;24: 1553–1568. doi:10.1162/NECO_a_00282

20. Suri RE. Anticipatory responses of dopamine neurons and cortical neurons reproduced by internal model. Exp Brain Res. 2001;140: 234–240. doi:10.1007/s002210100814

21. Barreto A, Munos R, Schaul T, Silver D. Successor Features for Transfer in Reinforcement Learning. arXiv Preprint. 2016;1606.

22. Tolman EC. Cognitive maps in rats and men. Psychol Rev. 1948;55: 189–208. doi:10.1037/h0061626

23. Simon DA, Daw ND. Neural correlates of forward planning in a spatial decision task in humans. J Neurosci. 2011;31: 5526–5539. doi:10.1523/JNEUROSCI.4647-10.2011

24. Sutton RS. Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bull. 1991;2: 160–163. doi:10.1145/122344.122377

25. Sutton RS, Barto AG. Reinforcement learning : an introduction. MIT Press; 1998.

26. Daw ND, Tobler PN. Value Learning through Reinforcement. In: Glimcher PW, Fehr E, editors. Neuroeconomics. 2nd ed. London: Elsevier; 2014. pp. 283–298. doi:10.1016/B978-0-12-416008-8.00015-2

27. Gershman SJ, Markman AB, Otto AR. Retrospective revaluation in sequential decision making: a tale of two systems. J Exp Psychol Gen. 2014;143: 182–94. doi:10.1037/a0030844

28. Glimcher PW. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. Proc Natl Acad Sci U S A. 2011;108 Suppl : 15647–54. doi:10.1073/pnas.1014269108

29. Thorndike EL, Jelliffe. Animal Intelligence. Experimental Studies. The Journal of Nervous and Mental Disease. Transaction Publishers; 1912. p. 357. doi:10.1097/00005053-191205000-00016

30. Camerer C, Ho T-H. Experience-Weighted Atttraction in Normal Form Games. Econometrica. 1999;67: 827–874.

31. Dickinson A, Balleine BW. The role of learning in the operation of motivational systems. In: Gallistel CR, editor. Steven's handbook of experimental psychology: Learning, motivation and emotion. New York: John Wiley & Sons; 2002. pp. 497–534. doi:10.1002/0471214426.pas0312

32. Wimmer GE, Shohamy D. Preference by association: how memory mechanisms in the hippocampus bias decisions. Science (80-). American Association for the Advancement of Science; 2012;338: 270–3. doi:10.1126/science.1223252

33. Yin HH, Knowlton BJ, Balleine BW. Lesions of dorsolateral striatum preserve outcome expectancy but disrupt habit formation in instrumental learning. Eur J Neurosci. 2004;19: 181–189. doi:10.1111/j.1460-9568.2004.03095.x

34. Balleine BW, O'Doherty JP. Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action. Neuropsychopharmacology. 2010;35: 48–69. doi:10.1038/npp.2009.131

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

36. Balleine BW, Daw ND, O'Doherty JP. Neuroeconomics. Neuroeconomics. 1st ed. 2009. pp. 367–387. doi:10.1016/B978-0-12-374176-9.00024-5

37. Alexander GE, Crutchner MD. Functional architecture of basal ganglia circuits: neural substrated of parallel processing. Trends Neurosci. 1990;13: 266–271. doi:10.1016/0166-2236(90)90107-L

38. Dezfouli A, Balleine BW. Habits, action sequences and reinforcement learning. Eur J Neurosci. 2012;35: 1036–1051. doi:10.1111/j.1460-9568.2012.08050.x

39. Haber SN. The primate basal ganglia: Parallel and integrative networks. J Chem Neuroanat. 2003;26: 317–330. doi:10.1016/j.jchemneu.2003.10.003

40. Faure A, Haberland U, Massioui N El. Lesion to the Nigrostriatal Dopamine System Disrupts Stimulus – Response Habit Formation. J Neurosci. 2005;25: 2771–2780. doi:10.1523/JNEUROSCI.3894-04.2005

41. Schultz W, Dayan P, Montague PR. A neural substrate of prediction and reward. Science (80-). 1997;275: 1593–1599. doi:10.1126/science.275.5306.1593

42. Tolman EC, Honzik CH. Introduction and removal of reward, and maze performance in rats. Univ Calif Publ Psychol. 1930;

43. Wang T, Bowlingm M, Schuurmans D. Dual representations for dynamic programming and reinforcement learning. Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning, ADPRL 2007. 2007. pp. 44–51. doi:10.1109/ADPRL.2007.368168

44. White LM. Temporal Difference Learning: Eligibility Traces and the Successor Representation for Actions [Internet]. University of Toronto. 1995. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.4525&amp;rep=rep1&amp;type=pdf

45. Parker NF, Cameron CM, Taliaferro JP, Lee J, Choi JY, Davidson TJ, et al. Reward and choice encoding in terminals of midbrain dopamine neurons depends on striatal target. Nat Neurosci. 2016;19. doi:10.1038/nn.4287

46. Howe MW, Dombeck DA. Rapid signalling in distinct dopaminergic axons during locomotion and reward. Nature. Nature Publishing Group; 2016;27: 1–22. doi:10.1038/nature18942

47. Thorn CA, Atallah H, Howe M, Graybiel AM. Differential Dynamics of Activity Changes in Dorsolateral and Dorsomedial Striatal Loops during Learning. Neuron. Elsevier Ltd; 2010;66: 781–795. doi:10.1016/j.neuron.2010.04.036

48. Barron HC, Dolan RJ, Behrens TEJ. Online evaluation of novel choices by simultaneous representation of multiple memories. Nat Neurosci. Nature Publishing Group; 2013;16: 1492–8. doi:10.1038/nn.3515

49. Lee SW, Shimojo S, O'Doherty JP. Neural Computations Underlying Arbitration between Model-Based and Model-free Learning. Neuron. Elsevier Inc.; 2014;81: 687–699. doi:10.1016/j.neuron.2013.11.028

50. Ciancia F. Tolman and Honzik (1930) revisited: or The mazes of psychology (1930-1980). Psychol Rec. 1991;41: 461–472. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Tolman+and+honzik+(1930)+revisited+or+the+mazes+of+psychology+(1930-1980)#0

51. Poucet B, Thinus-Blanc C, Chapuis N. Route planning in cats, in relation to the visibility of the goal. Anim Behav. 1983;31: 594–599. doi:10.1016/S0003-3472(83)80083-9

52. Winocur G, Moscovitch M, Rosenbaum RS, Sekeres M. An investigation of the effects of hippocampal lesions in rats on pre- and postoperatively acquired spatial memory in a complex

environment. Hippocampus. 2010;20: 1350–1365. doi:10.1002/hipo.20721

53. Jovalekic A, Hayman R, Becares N, Reid H, Thomas G, Wilson J, et al. Horizontal biases in rats' use of three-dimensional space. Behav Brain Res. 2011;222: 279–288. doi:10.1016/j.bbr.2011.02.035

54. Chapuis N, Durup M, Thinus-Blanc C. The role of exploratory experience in a shortcut task by golden hamsters (&lt;i&gt;Mesocricetus auratus&lt;/i&gt;). Learn Behav. 1987;15: 174–178. doi:10.3758/BF03204960

55. Alvernhe A, Van Cauter T, Save E, Poucet B. Different CA1 and CA3 representations of novel routes in a shortcut situation. J Neurosci. 2008;28: 7324–33. doi:10.1523/JNEUROSCI.1909-08.2008

56. Spiers HJ, Gilbert SJ. Solving the detour problem in navigation: a model of prefrontal and hippocampal interactions. Front Hum Neurosci. 2015;9: 1–15. doi:10.3389/fnhum.2015.00125

57. Corbit LH, Ostlund SB, Balleine BW. Sensitivity to instrumental contingency degradation is mediated by the entorhinal cortex and its efferents via the dorsal hippocampus. J Neurosci. 2002;22: 10976–84. doi:22/24/10976 [pii]

58. 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. doi:10.1037/0033-295X.102.3.419

59. Buzsáki G. Two-stage model of memory trace formation: A role for "noisy" brain states. Neuroscience. 1989;31: 551–570. doi:10.1016/0306-4522(89)90423-5

60. Schapiro AC, Kustner L V., Turk-Browne NB. Shaping of object representations in the human medial temporal lobe based on temporal regularities. Curr Biol. Elsevier; 2012;22: 1622–1627. doi:10.1016/j.cub.2012.06.056

61. Schapiro AC, Turk-Browne NB, Norman KA, Botvinick MM. Statistical learning of temporal community structure in the hippocampus. Hippocampus. 2016;26: 3–8. doi:10.1002/hipo.22523

62. Schapiro AC, Rogers TT, Cordova NI, Turk-Browne NB, Botvinick MM. Neural representations of events arise from temporal community structure. Nat Neurosci. Nature Research; 2013;16: 486–92. doi:10.1038/nn.3331

63. Botvinick MM, Niv Y, Barto AC. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. Cognition. 2009;113: 262–280. doi:10.1016/j.cognition.2008.08.011

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

65. Schapiro AC, Rogers TT, Cordova NI, Turk-Browne NB, Botvinick MM. Neural representations of events arise from temporal community structure. Nat Publ Gr. 2013;16. doi:10.1038/nn.3331

66. Boorman ED, Rajendran VG, O'Reilly JX, Behrens TE. Two Anatomically and Computationally Distinct Learning Signals Predict Changes to Stimulus-Outcome Associations in Hippocampus. Neuron. The Authors; 2016;89: 1343–1354. doi:10.1016/j.neuron.2016.02.014

67. Doll BB, Duncan KD, Simon DA, Shohamy D, Daw ND. Model-based choices involve prospective neural activity. Nat Neurosci. 2015;18: 767–772. doi:10.1038/nn.3981