A common model explaining flexible decision making, grid fields and cognitive control

Payam Piray* and Nathaniel D. Daw
Princeton Neuroscience Institute, Princeton University
Correspondence: ppiray@princeton.edu

Abstract

A central difficulty for computational theories of planning is that the value of an action taken now depends on which actions are chosen afterward. Thus, optimal choices are coupled across states. We argue that this interdependence underlies a pattern of challenges for reinforcement learning models to explain both the brain's flexibilities and inflexibilities. Building on advances in control engineering, we propose a model for decision-making in the brain that is more efficient, flexible and biologically realistic than previous attempts. It replaces the classic iterative optimization with a linear approximation that addresses interdependence by softly maximizing around a default policy. This solution exposes connections between seemingly disparate phenomena across neuroscience, notably flexible replanning with biases and cognitive control. It also gives new insight into how the brain can represent maps of long-distance contingencies stably and componentially, as in entorhinal response fields, and exploit them to guide choice even under changing goals.

Introduction

In the past two decades, computational ideas from reinforcement learning (RL) have provided an indispensable framework for understanding a wide range of cognitive faculties and their neural substrates, such as decision making, planning and cognitive control. Despite their success, however, these models have difficulty explaining the full range of behavior of humans and other animals: both in terms of their flexibility, and also their inflexibility. For example, in contrast to RL agents, animals are adept at making decisions effortlessly based on little data and experience by generalizing their knowledge efficiently across tasks and situations. On the other hand, unlike RL agents, human performance suffers from odd inflexibilities, such as biases in cognitive control tasks and Pavlovian response biases.

We suggest that these apparently opposite issues, and a number of other seemingly disparate phenomena in decision neuroscience, have common roots in how the brain addresses a core issue in choice. In particular, the central computational challenge in sequential decision tasks is that the optimal decision at every timepoint depends on the optimal decision at the next timepoint, and so on. In a maze, for instance, the value of going left or right now depends on which turn you make at the subsequent junction, and similarly thereafter; so, figuring out what is the best action now requires, simultaneously, also figuring out what are the best choices at all possible steps down the line. This interdependence between actions is a direct consequence of the natural definition of the objective function in this setting (i.e., the Bellman equation[1]), and this greatly complicates planning, replanning, task transfer, and temporal abstraction in both artificial intelligence and biological settings[2].

How, then, can the brain produce flexible behavior? Humans and animals can solve certain replanning tasks, such as reward devaluation and shortcuts, which require generating new action plans on the fly.[3–7] It has been argued that the brain does so by some variant of model-based planning; that is, solving the Bellman equation directly by iterative search[8,9]. However, we lack a biologically realistic account how this is implemented in the brain[10]; indeed, because of the interdependence of optimal actions, exhaustive search (e.g., implemented by neural replay[11]) seems infeasible for most real-world tasks due to the exponentially growing number of future actions that must each be, iteratively and nonlinearly optimized. It has thus also been suggested that the brain employs various shortcuts that rely on reusing previously computed ("cached") quantities, e.g. model-free long-run values[12,13].

One such proposal, which is perhaps the most promising step toward a neurally realistic planning algorithm is the successor representation (SR)[14], which by leveraging cached expectations about which states will be visited in future, can efficiently solve a subset of tasks traditionally associated with model-based planning[5,15]. However, it simply assumes away the key interdependent optimization problem by evaluating actions under a fixed choice policy (implied by the stored state expectancies) for future steps. This policy-dependence makes the model incapable of explaining how the brain can solve other replanning tasks, in which manipulations also affect future choices[15,16]. In general, the precomputed information stored by the SR is only useful for replanning when the newly replanned policy remains similar to the old one: For instance, a change in goals implies a new optimal policy that visits a different set of states, and a different SR is then required to compute it. This is just one instance of a general problem that plagues attempts to simplify planning by temporal abstraction (e.g., chunking steps[17,18]), again due to the interdependence of optimal actions: if my goals change, the optimal action at future steps (and, hence, the relevant chunked long-run trajectories) likely also change.

Here, we adopt and build on recent advances in the field of control engineering[19–21] to propose a new model for decision making in the brain that can efficiently solve for an approximation to the optimal policy, jointly across all choices at once. It does so by relying on a precomputed, temporally abstract map of long-run state expectancies similar to the SR, but one which is, crucially, stable and useful even under changes in the current goals and the decision policy they imply. The model, termed linear RL, provides a common framework for understanding different aspects of animals' cognitive abilities, particularly flexible planning and replanning using temporally abstract maps, but also biases in cognitive control and Pavlovian influences on decision making.

The model is based on a reformulation of the classical decision problem, which makes "soft" assumptions about the future policy (in the form of a stochastic action distribution), and introduces an additional cost for decision policies which deviate from this baseline. This can be viewed as an approximation to the classic problem, where soft, cost-dependent optimization around the default policy stands in for exact optimization of the action at each successor state. Crucially, the form of the costs allows the modified value function to be solved analytically using inexpensive and biologically plausible linear operations. In particular, the optimal value of any state under any set of goals depends on a weighted average of the long-run occupancies of future states that are expected under the default policy. Therefore, we propose that the brain stores a map of these long-run state expectancies across all states (the default representation, or DR), which gives a metric of closeness of states under the default policy. Because the DR depends only on the default policy, and can be used to compute a new optimal policy for arbitrary goals, the model can solve a large class of replanning tasks, including ones that defeat the SR.

Our novel modeling approach also addresses a number of seemingly distinct questions. First, the stability of the DR across tasks makes it a candidate for understanding the role in decision-making of multiscale, temporally abstract representations in the brain, notably grid cells in the medial entorhinal cortex. These cells show regular grid-like firing patterns over space, at a range of frequencies, and have been argued to represent something akin to a Fourier-domain map of task space (e.g., the eigenvalues of the SR, equivalent to the graph Laplacian[22,23]), and could provide some sort of mechanism for spatial[24] and mental navigation[7,25,26]. However, it has been unclear how this and similar long-run temporal abstractions are actually useful for planning or navigation, because as mentioned long-run (low-frequency) expectancies over task space are not stable across tasks due to the interdependence of policy, goals, and trajectories[27,28]. For instance, because the SR only predicts accurately under the training policy, to be even marginally useful for replanning the SR theory predicts grid fields must continually change to reflect updated successor state predictions as the animal's choice policy evolves, which is inconsistent with evidence[29–31]. The linear RL theory clarifies how the DR, a stable and globally useful long-run map under a fixed default policy, can serve flexible planning. Our theory also provides a new account for updating maps in situations which actually do require modification – notably, the introduction of barriers. We show how these give rise to additional, separable basis functions in the corresponding DR, which we associate with a distinct class of entorhinal response fields, the border cells. This aspect of the work goes some way toward delivering on the promise of such response as part of a reusable, componential code for cognitive maps[7,25].

Also, linear RL addresses the flip side of how the brain can be so flexible: why, in some cases it is inflexible. We suggest that this is simply another aspect of the same mechanisms used to enable flexible planning. While it has long been suggested that fully model-free learning in the brain might account for extreme cases of goal-inconsistent habits (e.g., animals persistently working for food when not hungry[8]), there are many other phenomena which appear as more graded biases, such as Stroop effects, Pavlovian tendencies, slips of action[32], and more sporadic failures of participants to solve replanning tasks[5]. The default policy and cost term introduced to make linear RL tractable offers a natural explanation for these types of phenomena, and in particular gives a novel rationale and explanation for a classic problem in cognitive control: the source of the apparent costs of "control-demanding" actions.

Despite its simplicity, the linear RL model accounts for a diverse range of problems across different areas of cognitive neuroscience. In the reminder of this article, we present a series of simulation experiments that demonstrate that the theory provides i) a biologically-realistic, efficient and flexible account of decision making; ii) a novel understanding of entorhinal grid code that explains its role in flexible planning, navigation and inference; iii) an understanding of cognitive control that naturally links it to other aspects of decision systems; and iv) a normative understanding of Pavlovian-instrumental transfer (PIT).

Results

The Model

In Markov decision tasks, like mazes or video games, the agent visits a series of states s, and at each they receive some reward or punishment r and choose among a set of available actions a, which then affects which state they visit next[2]. The objective in this setting is typically to maximize the expected sum of future rewards, called the 'value' function. Formally, the optimal value $\bar{v}^*$ of some state is given by the sum of future rewards, as a series of nested expectations:

$$\bar{v}^*(s_t) = r(s_t) + \max_{a_t} \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \left[r(s_{t+1}) + \max_{a_{t+1}} \sum_{s_{t+2}} P(s_{t+2}|s_{t+1}, a_{t+1}) [r(s_{t+2}) + \cdots] \right]$$

or equivalently in recursive form by the Bellman equation (Bellman, 1957):

$$\bar{v}^*(s_t) = r(s_t) + \max_{a_t} \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \bar{v}^*(s_{t+1}) \tag{1}$$

Across all states, this results in a set of interdependent optimization problems, which can be solved, for instance, by iterative search through the tree of future states, computing the maximizing action at each step[2]. However, in realistic tasks with large state spaces, this iterative, nonlinear computation may be intractable.

Note that prediction can be used for action choice or computing an action selection policy: once we have computed $\bar{v}^*$ (the optimal future reward available from each state), we can just compare it across actions to find the best action in any particular state and form a policy, π^*; for instance, we can evaluate the max in equation (1) for any state, plugging in the optimal values of successor states without further iteration. However, note also that this depends on having already found the maximizing action at other states down the line, since $\bar{v}^*$ depends, recursively, on which actions are taken later, and this in turn depends on the assignment of rewards to states (e.g., the agent's goals).

If we instead assumed that we were going to follow some given, not necessarily optimal, action selection policy π at each subsequent state (say, choosing randomly), then equation (1) would be replaced by a simple set of linear equations (eliminating the nonlinear function "max" at each step) and relatively easily solvable. This observation is the basis of the SR model[5,14,15,22], which computes values as

$$\bar{\mathbf{v}}^{\pi} = \mathbf{S}^{\pi}\mathbf{r}, \tag{2}$$

where (in matrix-vector form) $\bar{\mathbf{v}}^{\pi}$ is a vector of long-run state values under the policy π; $\mathbf{r}$ a vector of state rewards; and $\mathbf{S}^{\pi}$ a matrix measuring which subsequent states one is likely to visit in the long run following a visit to any starting state: importantly, assuming that all choices are made following policy π. However, although this allows us to find the value of following policy π, this does not directly reveal how to choose optimally. For instance, plugging these values into equation (1) won't produce optimal choices, since $\bar{\mathbf{v}}^{\pi}$ (the value of choosing according to π in the future) in general does not equal the value, $\bar{\mathbf{v}}^*$, of choosing optimally. The only way to find the latter using equation (2) is by iteratively re-solving the equation to repeatedly update π and $\mathbf{S}$ until they eventually converge to π^*, i.e., the classic policy iteration algorithm.

It has recently been shown that a change in the formulation of this problem, which we refer to as *linear RL*, greatly simplifies the Bellman equation[19–21]. To see this, we first assume a one-to-one, deterministic correspondence between actions and successor states (i.e., for every state s' reachable in one step from some s, assume there is a corresponding action a for which $P(s'|s,a) = 1$, which is simply denoted by its destination, s'). This fits many problems with fully controllable, deterministic dynamics, such as spatial navigation (where for each adjacent location, there is a corresponding action taking you there). Second, linear RL seeks to optimize not a discrete choice of successor state (action), but a stochastic probability distribution π over it[20,21]. Finally, it redefines the value function to include not just the one-step rewards r but also at each step a new penalty[19–21], called a "control cost," $\mathrm{KL}(\pi||\pi^d)$, which is increasing in the dissimilarity (KL divergence) between the chosen distribution π and some *default* distribution, π^d.

Linear RL is most naturally a formalism for modeling tasks in which there are some default dynamics (e.g., a rocket in a gravitational field) and costly actions to modify them (e.g., firing thrusters burning different amounts of fuel). Alternatively, here we view it as an approximation to the original value function, where the additional penalty terms modify the original problem to a related one that can be more efficiently solved. This is because linear RL deals with the problem of the interdependence of the optimal actions across states[19–21]: the default policy π^d represents a set of soft assumptions about which actions will be taken later, which are optimized into an optimal stochastic distribution π^* that is approximately representative of the optimal (deterministic) subsequent choices in the original problem.

Efficient solution is possible because, substituting the penalized rewards into the Bellman equation, the optimal value function is now given by a non-recursive, linear equation[20,21]:

$$\exp(\mathbf{v}^*) = \mathbf{MP}\exp(\mathbf{r}), \tag{3}$$

such as can be computed by a single layer of a simple, linear neural network. Here, $\mathbf{v}^*$ is a vector of the optimal values (now defined as maximizing cumulative reward minus control cost) for each state; $\mathbf{r}$ is a vector of rewards at a set of "terminal" states (i.e., various possible goals); $\mathbf{P}$ is a matrix containing the probability of reaching each goal state from each other, nonterminal, state; and the key matrix $\mathbf{M}$, which we call the default representation (DR), measures the closeness of each nonterminal state to each other nonterminal state (in terms of expected aggregate cost to all future visits) under the default policy. This is similar to the SR ($\mathbf{S}^{\pi}$, equation (2)), except that it is for the optimal values $\mathbf{v}^*$ (not the on-policy values $\mathbf{v}^{\pi}$), and $\mathbf{v}^*$ is systematically related to optimal values as defined in the original problem ($\bar{\mathbf{v}}^*$, Eq. 1), with the difference being the additional penalties for deviation from the default policy. But these exert only a soft bias in π^* toward π^d, which furthermore vanishes altogether in an appropriate limit (see Methods). Thus, while $\mathbf{M}$ does depend on the default policy π^d, it is stable over changes in goals and independent from π^* in the sense that it can usefully find optimized policies π^* even when these are far from π^d; in comparison, $\mathbf{v}^{\pi}$ (computed from the SR: $\mathbf{S}^{\pi}$) is only a useful aproximation to $\mathbf{v}^*$ (and thus only helpful in finding a new π^*) when the SR's learned policy π is near the target policy π^*. Effectively, linear RL works by introducing a smooth approximation of the "max" in equation (1), since the log-average-exp (with the average here taken with respect to the default distribution, π^d) of a set of values approximates the maximum. The control costs, then, simply capture the difference between the original solution and the smooth approximate one.

Model Performance

The optimized policy in this model balances expected reward with control cost, and is generally stochastic rather than deterministic, like a softmax function (Fig 1a-b). We evaluated the performance of linear RL as an approximation to exact solution by considering a difficult, 7-level decision tree task in which each state has two possible successors, a set of costs are assigned randomly at each state, and the goal is to find the cheapest path to the bottom. We conducted a series of simulations, comparing linear RL with a set of benchmarks: exact (model-based) solution, and a set of approximate model-based RL agents[12] that optimally evaluate the tree up to a certain depth, then "prune" the recursion at that leaf by substituting the exact average value over the remaining subtree (Fig 1c; in the one-step case this is equivalent to the SR under the random walk policy). For linear RL, the default policy was taken as a uniform distribution over possible successor states. Linear RL achieved near-optimal average costs (Fig 1d).

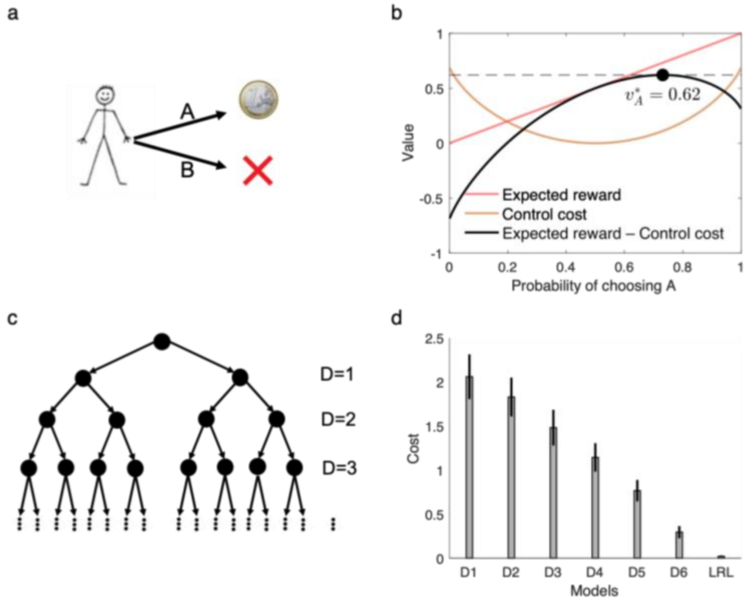


Fig 1. The linear RL model. a-b) the model optimizes the decision policy by considering the reward and the control cost, which is defined as the KL divergence between the decision policy and a default policy. Assuming an unbiased (uniform) distribution as the default policy, the optimal decision policy balances the expected reward with the control cost. Although the expected reward is maximum when probability of choosing A is close to 1 (and therefore probability of choosing B is about zero), this decision policy has maximum control cost due to its substantial deviation from the default policy. The optimal value instead maximized expected reward minus the control cost, which here occurs when probability of choosing A is 0.73. c-d) The model accurately approximates optimal choice. We compared its performance on a 7-level decision tree task (with random one-step costs at each state) to 6 pruned model-based RL algorithms, which evaluate the task to a certain depth (D = 1,..,6; D7 is optimal; D1 is equivalent to the successor representation for the random walk policy) and use average values at the leaves. Linear RL (LRL) achieved near-optimal average costs (y-axis is additional cost relative to the optimum). Local costs of all states were randomly chosen in the range of 0 to 10, and simulations were repeated 100 times. Mean and standard error across all simulations are plotted.

An important aspect of linear RL is that the DR, $\mathbf{M}$, reflects the structure of the task (including the distances between all the nonterminal states under the default policy) in a way that facilitates finding the optimal values, but is independent of the goal values $\mathbf{r}$, and the resulting optimized value and policy (Fig 2). Therefore, by computing or learning the DR once, the model is able to re-plan under any change in the value of the goals (see below) and also (with some additional computation to efficiently add an additional terminal goal state, see *Methods*), plan toward any new goal with minimal further computation (Fig 2b-c). In the case of spatial tasks, this corresponds to finding the shortest path from any state to any goal state. In fact, our simulation analysis in a maze environment revealed that linear RL efficiently finds the shortest path between every two states in the maze (Fig 2d).

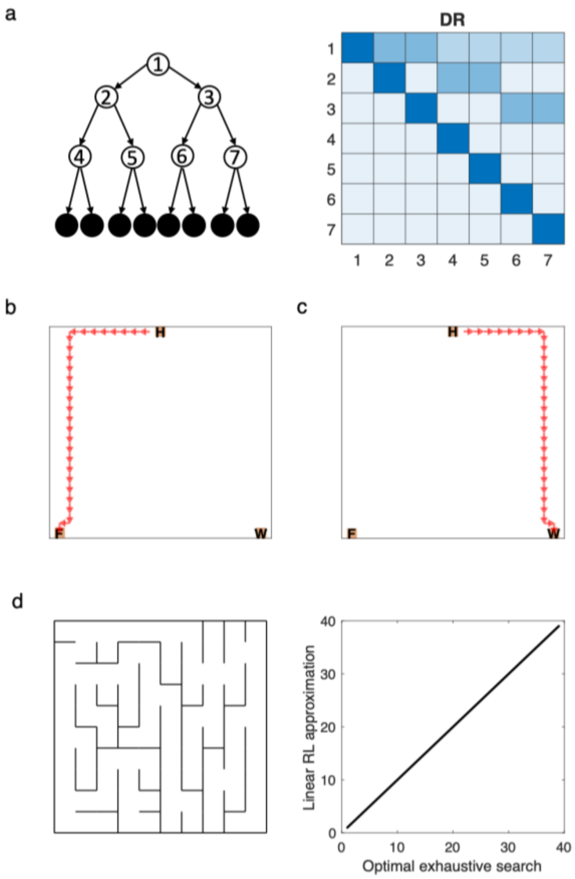


Fig 2. Default representation. a) The DR corresponding to a three-level decision tree task is shown. Each row of the DR represents weighted future expectancies starting from the corresponding state and following the default policy. Therefore, the DR is independent of the goals and optimized policy. b-c) The optimized path for planning from home (H) to the food (F) state is computed based on the DR. The linear RL model is efficient because the same DR is sufficient for planning towrds a new goal, such as the water (W) state. d) The path between every two states in a 10-by-10 maze environment (d) computed by linear RL exactly matches the optimal (shortest) path computed by exhaustive search. The DR has been computed once and reused (in combination with techinuqes from matrix identities) to compute each optimal path.

Replanning

In both artificial intelligence, and psychology and biology, a key test of efficient decision making is how an agent is able to transfer knowledge from one task to another. For instance, many tasks from neuroscience test whether organisms are able, without extensive retraining, to adjust their choices following a change in the rewards or goals ("revaluation," "devaluation," "latent learning") or transition map ("shortcut," "detour") of a previously learned task[3–6]. We explored the ability of linear RL for solving these types of replanning problems (Fig 3). Importantly, the model is able to solve one class of these problems – those involving revaluation of goals – efficiently, as the DR can be used, unmodified, to solve any new problem. This corresponds to simply changing $\mathbf{r}$ in Eq. 3, and computing new values.

First, we confirmed that linear RL is able to solve a version of Tolman's latent learning task (Fig 3a), a revaluation task in which rats were first trained to forage freely in a maze with two rewarding end-boxes, but then were shocked in one of the end-boxes to reduce its value[33]. This manipulation defeats model-free RL algorithms like temporal difference learning, because they must experience trajectories leading from the choice to the devalued box to update previously learned long-run value or policy estimates[8]. In contrast, rats are able to avoid the path leading to the devalued end-box on the first trial after revaluation, even though they had never experienced the trajectory following the devaluation. Linear RL is also able to correctly update its plans using the DR computed in the learning phase (Fig 3b-c). In particular, during the revaluation phase, the reward associated with one of the end-boxes changes but the structure of the environment remains the same: the revaluation corresponds to a change in $\mathbf{r}$ but not $\mathbf{M}$. Therefore, the agent is able to use the DR computed during the learning phase in the test phase and update its policy according to revalued reward function.

The SR is also capable of solving the latent learning task (and similar reward devaluation tasks with only a single step of actions widely used in neuroscience[3]), because the SR, $\mathbf{S}^{\pi}$, even though learned under the original policy π, is good enough to compute usable new values from the new reward vector[15]. However, there are many other, structurally similar revaluation tasks – in particular, those with several stages of choices – that defeat the SR. We considered a slightly different revaluation task, which Russek et al.[5,15] termed "policy revaluation" that has this property. Here human subjects were first trained to navigate a three-stage sequential task leading to one of the three terminal states (Fig 3d[5]). The training phase was followed by a revaluation phase, in which participants experienced the terminal states with potentially new reward. In particular, a new large reward was introduced at a previously disfavored terminal state. In the final test, participants were often able to change their behavioral policy at the starting state of the task, even though they had never experienced the new terminal state contingent on their choices in the task[5].

Importantly, this is not possible for the SR without relearning or recomputing the successor matrix $\mathbf{S}^{\pi}$, because under the original training policy, the cached successor matrix does not predict visits to the previously low-valued state[15,16]. That is, it computes values for the top-level state (1 in Fig 3d) under the assumption of outdated choices at the successor state (2), neglecting the fact that the new rewards, by occasioning a change in choice policy at 2 also imply a change in choice policy at 1. This task then, directly probes the agent's ability to re-plan respecting the interdependence of optimal choices across states. Unlike the SR, linear RL can successfully solve this task using the DR that has been computed in the training phase, because the DR is independent of the decision policy in the learning phase (Fig 3e).

We finally considered a different class of replanning tasks, in which the *transition* structure of the environment changes, for example by placing a barrier onto the maze as to block the previously preferred path[6]. These tasks pose a challenge for both the SR and DR, since the environmental transition graph is cached inside both $\mathbf{S}^{\pi}$ and $\mathbf{M}$[5,15], and these must thus be updated by relearning or recomputation in order to re-plan. However, people are again often able to solve this class of revaluations[5]. We introduce an elaboration to linear RL to permit efficient solution of these tasks: in particular, we exploit matrix identities that allow us to efficiently update $\mathbf{M}$ in place to take account of local changes in the transition graph, then re-plan as before (see *Methods*). With these in place, the linear RL model can solve this task efficiently and computes the modified values and optimized policy using the old DR after updating it with simple operations (Fig 3h).

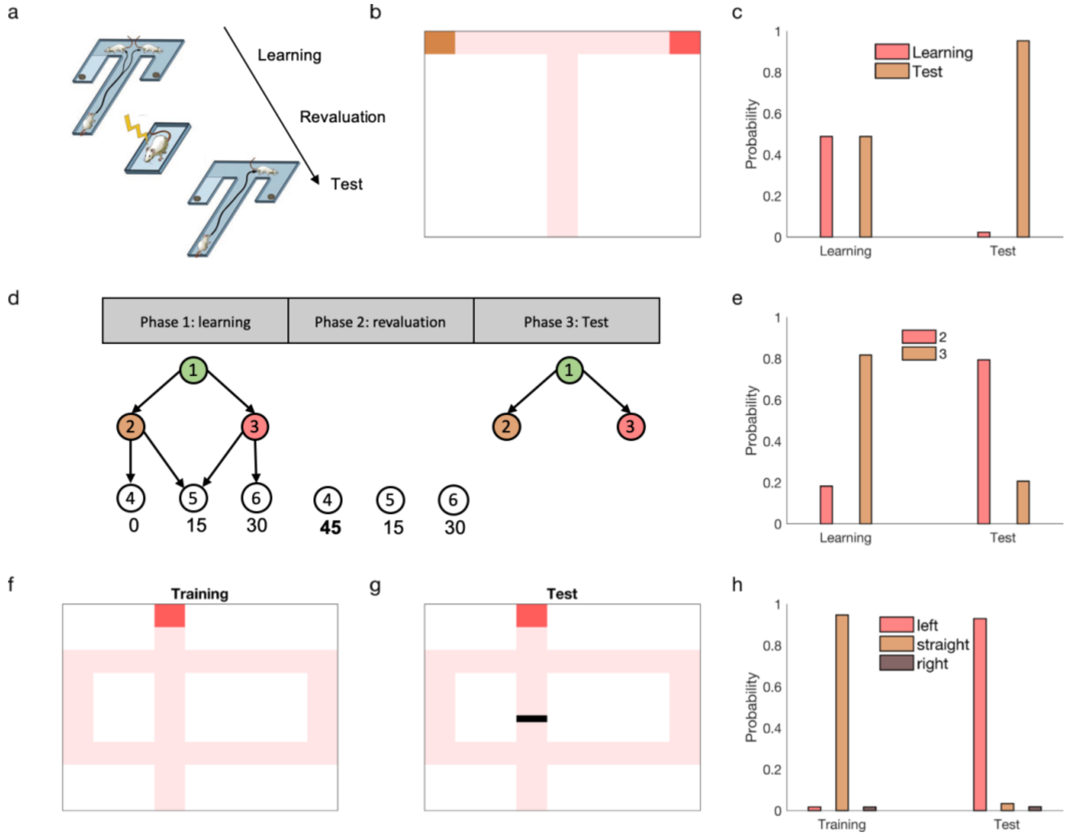


Fig 3. Linear RL can explain flexible replanning. a-c) Performance of linear RL on a version of Tolman's latent learning task (a). We simulated the model in a maze representing this task (b) and plotted the probability of choosing each end-box during the learning and test phases. The model correctly (c) reallocates choices away from the devalued option. d-e) Performance of linear RL in another reward revaluation task (Momennejad et al., 2017), termed policy revaluation (d). Choices from state 1: During the learning phase, the prefers to go to state 3 rather than state 2. Revaluation of the bottom level state reverses this preference (e) similar to human subjects (Momennejad et al., 2017). f-h) Performance of the model in Tolman's detour task. The structure of the environment changes in this task due the barrier placed into the maze (g), which blocks the straight path. The model is able to compute the optimized policy using the old DR (following a single, inexpensive update to it) and correctly choose the left path in the test phase (h).

Grid fields

The linear RL model also highlights, and resolves, a central puzzle about the neural representation of cognitive maps or world models. It has long been argued that the brain represents a task's structure in order to support planning and flexible decision making[6]. This is straightforward for maximally local information: e.g., the one-step transition map $P(s_{t+1}|s_t, a_t)$ from Eq. 1, might plausibly be represented by connections between place fields in hippocampus, and combined with local-state reward mappings $r(s_t)$ that could be stored in hippocampal-stratial projections. But using this information for planning requires exhaustive evaluation, e.g. by replay[11], and strongly suggesting a role for map-like representations of longer-scale relationships (aggregating multiple steps) to simplify planning[18,34].

Indeed, grid cells in entorhinal cortex represent long-range (low-frequency) periodic relationships over space, and theoretical and experimental work has suggested that they play a key role in representation of the cognitive map and support navigation in both physical[24] and abstract[7,25] state spaces. However, the specific computational role of these representations in flexible planning is still unclear. A key concept is that they represent a set of basis functions for quickly building up other functions over the state space, including future value predictions like $\bar{v}^{*}$[23] and also future state occupancy predictions like the SR[22,35]. By capturing longer range relationships over the map, such basis functions could facilitate estimating or learning these functions[23]. In particular, the graph Laplacian (given by the eigenvectors of the on-policy, random walk transition matrix or, equivalently the eigenvectors of the SR for the random walk policy) generalizes Fourier analysis to an arbitrary state transition graph, and produces a set of periodic functions similar to grid fields[22,28], including potentially useful low-frequency ones.

The puzzle with this framework is that, as mentioned repeatedly, the long-range transition map is not actually stable under changes in goals, since it depends on action choices ("max") at each step of Eq. 1: in effect, the spatial distribution of goals biases what would otherwise be a pure map of space, since those affect choice policy, which in turn affects experienced long-run location-location contingencies. Conversely, basis functions built on some fixed choice policy (like the SR for a particular π) are of limited utilty for transferring to new tasks[15,16]. Accordingly, algorithms building on these ideas in computer science (such as "representation policy iteration,"[27]), iteratively update basis functions to reflect changing policies and values as each new task is learned. It has been unclear how or whether representations like this can usefully support more one-shot task transfer, as in the experiments discussed in the previous section.

As shown in the previous section, linear RL resolves this problem, since the DR is similar to the SR but stably useful across different reward functions and resulting choice policies. In particular, the comparison between Eqs. 2 and 3 shows that the DR is a stable linear basis for the (approximate) optimal value function regardless of the reward function, but the SR is not. Accordingly, we suggest that grid cells encode an eigenvector basis for the DR, functions which are also periodic and have grid-like properties in 2D environments (Fig 4d). Empirically, because both the SR and DR represent relationships under the objective transition graph (e.g., barrier locations in space), both theories that grid fields should be affected by changes in the objective transition contingencies of the environment (e.g., barrier locations in space; though see the next section for another way to address this). This is indeed the case experimentally[30,31] (Fig. 4abc). However, the key experimental prediction is that grid fields based on the DR should be stable under changes in the choice policy, whereas the SR (and its eigenvectors) are strongly policy-dependent, so grid fields based on it should change to reflect the animal's tendency to follow particular trajectories[22]. Experimental data strongly support the DR's prediction that grid fields are robust to behavioral changes; for instance, grid cells are affected by walls producing a "hairpin maze" but in rats trained to run an equivalent hairpin pattern without barriers[31] (Fig. 4ab); grid cells are also affected by the presence or absence of a set of walls the same shape as the animal's home cage, but whether or not it is the actual home cage (which strongly affects behavioral patterns) does not change the responses[29] (Fig. 4c). Similar results have been reported in humans using functional neuroimaging[36]. A second difference between the SR and the DR is that the DR (and its eigenvectors) include information about local costs along a path, so we predict that environmental features that make locomotion difficult, like rough terrain or hills, should modulate grid responses (see Discussion).

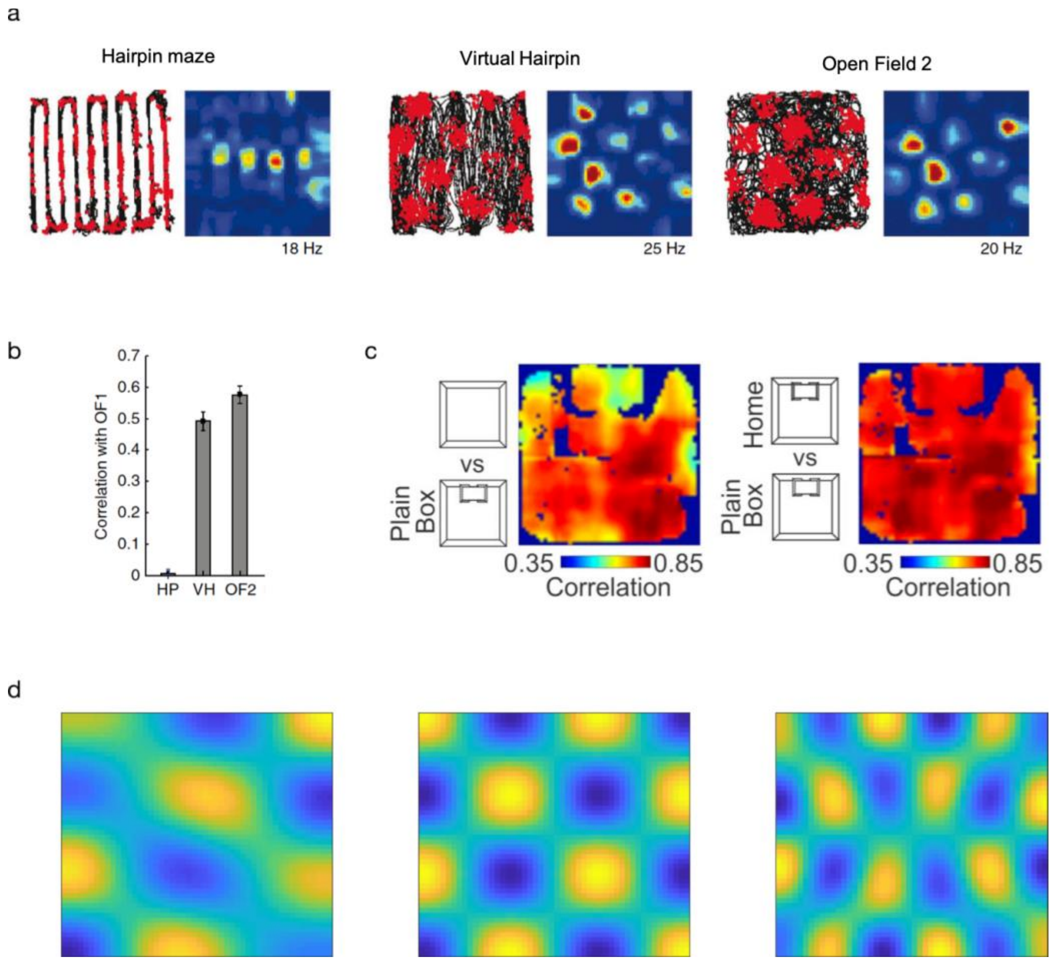


Fig 4. The DR as a model of grid fields. a-b) (adapted from Derdikman et al., 2009) Grid fields are sensitive to the geometry of the environment, but are stable with respect to behavior. Derdikman et al.[31] tested grid fields in a hairpin maze formed by actual barriers, and compared them to those recorded in a “virtual” hairpin maze, in which rats were trained to show hairpin-like behavior in an open field without constraining side walls. Grid fields in the virtual hairpin differ from those in the hairpin maze but are similar to the open field. b) This similarity is quantified by the correlation between grid fields in a baseline from an initial open field test (OF1) and those from the three tasks (HP: hairpin maze; VH: virtual hairpin; OF2: the second control open field). c) (Adapted from Sanguinetti-Scheck and Brecht[29]) Grid fields are sensitive to the presence of the home cage only insofar as it introduces new barriers in space, but not through the changes it produces in behavior. In particular, introducing a plain box (the same shape as the home cage) affects grid fields compared to the open field (left); but substuting the home cage for the box (right) does not further affect the grid code, although it changes behavior. d) Eigenvectors of the DR are independent from behavioral policies and periodic, similar to grid fields. Three example eigenvectors from a 50-by-50 maze are plotted.

Border cells

As we have already shown, one aspect of the environment that does require updating the DR if it changes is the transition structure of the environment, such as barriers. In simulating the Tolman detour task (Fig 3c) we solved this problem using a matrix inversion identity, which rather than expensively recomputing the entire DR with respect to the new transition graph, expresses the new DR as the sum of the original DR plus a low-rank correction matrix reflecting, for each pair of states, the map change due to the barrier. This suggests a novel, componential way to build up spatial distance maps, such as the DR, by summing basis functions that correspond to generic components, like walls. In this case, grid cells could represent a low-rank (e.g. eigenvector) representation for a baseline map, and other cells could represent the contribution of additional environmental features. Here, we highlight the relevance and importance of this computational approach in the context of entorhinal border cells (Fig 5a). This is another principal family of neurons in the medial entorhinal cortex that fire exclusively when the animal is close to a salient border of the environment[37], such as the wall; and are generic in the sense that they retain this tuning at least across changes in the environment's geometry. Assuming that the DR has been represented using a combination of features from a low-rank basis set, such as its eigenvectors, the columns of the matrix term for updating the DR show remarkable similarity to the border cells (Fig 5b). This brings the border cells and grid cells under a common understanding (both as basis functions for representing the map), and helps to express this map in terms of more componential features, like walls.

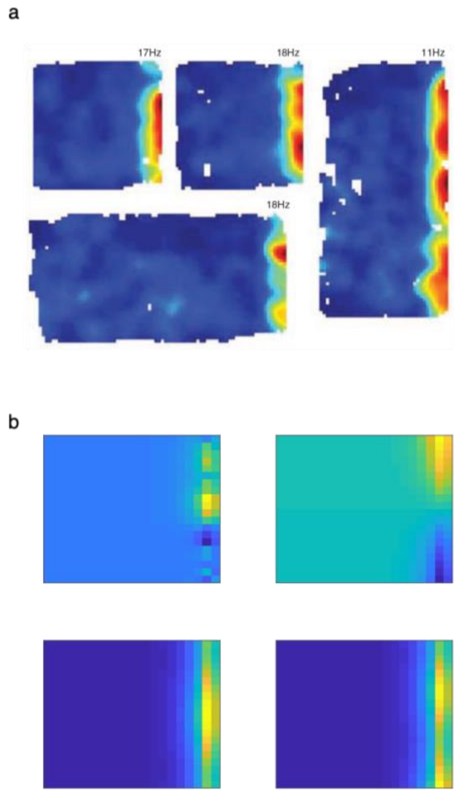


Fig 5. The model explains boundary cells. a) Adapted from Solstad et al.[37], which shows rate maps for a representative border cell in different boxes. b) Columns of the matrix required to update the DR matrix to account for the wall resemble border cells. Four example columns from a 20-by-20 maze are plotted.

Cognitive control

We have stressed the usefulness of linear RL for enabling flexible behavior, but because of the inclusion of the default policy, the model also offers a natural framework for understanding biases and inflexibilities in behavior, and phenomena of cognitive control for overcoming them – as necessary consequences of the very same computational mechanisms that permit flexibility. The default policy represents soft, baseline assumptions about action preferences, which (on this view) are introduced because they help efficiently to solve the problem of forecasting the set of optimal future choices during planning. So far, we have simulated it as unbiased (uniform over successors), which works well; but in situations with stable, clear regularities in behavior, it can be an even better approximation to build these in via a nonuniform default. If the default policy is not uniform, it softly biases the model towards actions that are common under the default policy. This aspect of the model naturally captures biases in human behavior, such as Stroop effects and Pavlovian biases (next section), and suggests a novel rationale for them in terms of the default policy's role in facilitating efficient planning.

Cognitive control has been defined as the ability to direct behavior toward achieving internally maintained goals and away from responses that are in some sense more automatic but not helpful in achieving those goals[38,39]. Two classic puzzles in this area are, first, why are some behaviors favored in this way; and second, why do people treat it as costly to overcome them[40–42]? For instance, is there some rivalrous resource or energetic cost that makes some behaviors feel more difficult or effortful than others[42,43]? Such "control costs" arise naturally in the current framework, since actions are penalized if they are more unlikely under the default policy. Such deviations from default are literally charged in the objective function, in units of reward: though for computational reasons of facilitating planning, rather than energetic ones like consuming a resource.

These control costs trade off in planning against the rewards for different actions, and lead (through the stochastic resulting policy) to biased patterns of errors. Fig 6a,b plots the control cost as a function of the decision policy, showing that the cost is substantially larger for choosing the action that is less likely under the default policy. For instance, action A in this simulation could be the color-naming response in the classic Stroop task, in which participants must read the name of a color that it is printed in a differently colored ink. People are faster and make fewer errors in word reading compared to color naming, presumably because the former is a more common task. For the same reason, we would expect color naming to be less likely under the default policy, and incur a larger control cost to execute reliably (Fig 6b). For any particular reward function (utility for correct and incorrect responses), this results in a larger chance of making errors for this action: a classic Stroop effect. Furthermore, since the optimal policy in the linear RL model balances the expected reward with the control cost, the model correctly predicts that these Stroop biases can be offset by increasing the rewards for correct performance[44] (Fig 6c). In other words, the prospect of reward can enhance performance even when the task is very difficult, as has been shown experimentally[44,45].

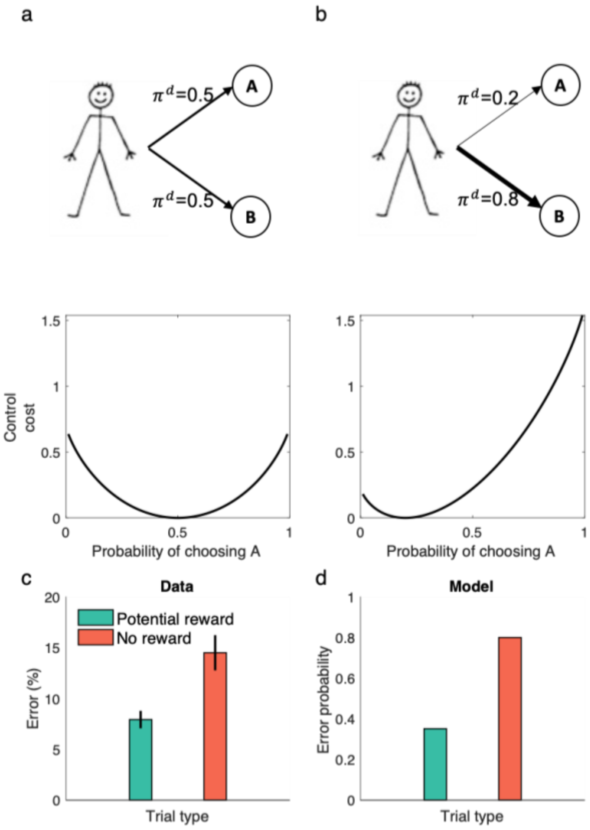


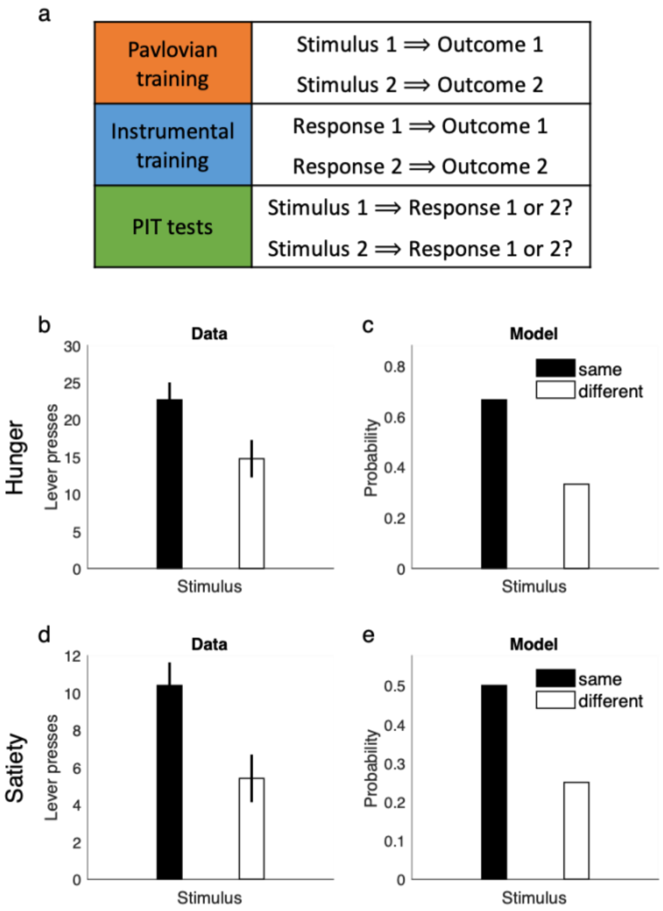
Fig 6. Linear RL captures prepotent actions and costs of cognitive control. a-b) The control cost is plotted as a function of the decision policy. For a uniform distribution (a) as the default policy, the control cost is a symmetric function of the decision policy. When the default policy is skewed toward a more likely response (b), the control cost is higher for reliably performing the action that is more unlikely under the default. c) People show classical Stroop effect in a color-naming Stroop task in which the name of colors are printed in the same or different color (Krebs, Boehler, & Woldorff, 2010). These errors, however, are reduced in potential reward trials, in which correct response is associated with monetary reward. d) The linear RL model shows the same behavior, because the default probability is larger for the automatic response (i.e. word reading). Promising reward reduces this effect because the agent balances expected reward against the control cost to determine the optimized policy.

Pavlovian-instrumental transfer

A second example of response biases in the linear RL model arises in Pavlovian effects. In particular, PIT is a phenomenon by which previously learned Pavlovian stimulus-outcome relationships influence later instrumental decisions (action choices). Puzzlingly, this happens even though the Pavlovian cues are objectively irrelevant to the actions' outcomes[46,47]. PIT – in this case, associations between drug-associated cues and drugs triggering drug-seeking actions – has been argued to play a key role in the development of addiction and cue-induced relapse[48].

In a typical PIT task (Fig 7a), animals first learn that a neutral stimulus, such as a light, predicts some rewarding outcome in a Pavlovian phase. Later, in an instrumental phase, they learn to press a lever to get the same outcome. In the final testing phase, the presentation of the conditioned stimulus biases responding toward the action for the associated reward, even though the stimulus has never been presented during instrumental phase and the stimulus is objectively irrelevant as the action produces the outcome either way (Fig 7b). Existing RL models typically fail to explain this result, instead predicting that the presence of the stimulus should not influence behavior in the test phase, because actions predict the same outcome contingencies regardless of the stimulus.

Linear RL explains PIT as another example of biases arising from a learned default policy, because during the Pavlovian phase the agent learns that the reward outcome occurs more often in the presence of the conditioned stimulus, which is reflected in the default contingencies. Therefore, during the test phase, the presentation of a conditioned stimulus elicits a default policy biased toward the corresponding outcome occurring, which favors choosing the corresponding action (Fig 7c). Furthermore, this effect is carried by the sensory (state) aspects of the outcome, not its rewarding properties per se. In particular, since in the absence of reward, the decision policy is equal to the default policy, the theory predicts that PIT effects persist even in the absence of reward, which is consistent with experimental work showing that PIT biases survive even under reward devaluation (e.g. for food outcomes tested under satiety) (Fig 7d-e). This finding that PIT effects reflect some sort of sensory cuing, and not reward or motivational properties of the stimulus per se, is central to the hypothesis that they underlie some phenomena in drug abuse such as cue-elicited relapse following extinction[48].



Pavlovian training	Stimulus 1 ⟹ Outcome 1 Stimulus 2 ⟹ Outcome 2
Instrumental training	Response 1 ⟹ Outcome 1 Response 2 ⟹ Outcome 2
PIT tests	Stimulus 1 ⟹ Response 1 or 2? Stimulus 2 ⟹ Response 1 or 2?

Fig 7. Linear RL explains Pavlovian-instrumental transfer. a) the task testing outcome-specific PIT consists of three phases: a Pavlovian training phase, an instrumental training phase and the PIT test. Outcomes 1 and 2 are both rewarding. During PIT test, both stimuli are presented in succession, and "same" responses denote the one whose associated outcome matches that associated with the presented stimulus, e.g. Response 1 chosen following presentation of Stimulus 1. The other response is "different." b-c) Data from Corbit et al.[49] when rats are hungry (b) and simulated behavior of the model (c). The model learns the default policy during the Pavlovian phase, which biases performance during the PIT test. d-e) Outcome-specific PIT persists even when rats are sated on both outcomes[49] (d). The model shows the same behavior (e) because default state probabilities learned during Pavlovian training influence responses even in absence of reward. Mean and standard error are plotted in b and c.

Discussion

A central question in decision neuroscience is how the brain can store cognitive maps or internal models of task contingencies and use them to make flexible choices, and more particularly how this can be done efficiently in a way that facilitates re-use of previous computations and leverages long-run, temporally abstract predictions without compromising flexibility. To help answer this question, we identify a core issue underlying many difficulties in planning, replanning, and reuse, which is the interdependence of optimal actions across states in a sequential decision task. To solve this problem, we import from control theory[20,50] to neuroscience a novel computational model of decision making in the brain, called linear RL, which enables efficient (though approximate) global policy optimization by relying on soft relaxation away from default, stochastic policy expectations. This leverages the DR, a stored, long-run predictive map of state and cost expectancies under the default policy. The DR is closely related to the SR, and inherits many of the appealing features that have generated current excitement for it as a neuroscientific model[5,15,22,51]. However, linear RL corrects serious problems that hobble the practical applicability of the SR. The DR, unlike the SR, exerts only a weak bias toward the default policy, and so delivers on the promise of a stable cognitive map[6] that can reuse substantial computation to transfer learning across contexts without sacrificing flexibility. This allows the model to explain animals' ability to solve reward and policy revaluation problems that otherwise would require exhaustive, biologically unrealistic model-based search. For the same reason, the model also helps to deliver on the idea that grid cells in entorhinal cortex could provide a broadly useful neural substrate for such a temporally abstract map. And the model's remaining inflexibilities – in general, soft, stochastic biases rather than hard failures – connect naturally with phenomena of cognitive control and Pavlovian biases and provide a strong theoretical framework for understanding the role of many such biases in both healthy and disordered choice.

The basic planning operation in linear RL is matrix-vector multiplication, which is easily implemented in a single neural network layer. The theory offers new insights into the systems-level organization of this computation. In particular, the model realizes the promise of a representation that factors a map representing the structure of environment, separate from an enumeration of the current value of the goals in the environment. This facilitates transfer by allowing update of either of these representations while reusing the other. Previous models, like the SR, nominally exhibit this separation, but the hard policy dependence of the SR's state expectancies means that goal information, in practice, inseparably infects the map and interferes with flexible transfer[15,16].

In fact, in order to facilitate efficient planning, the linear RL model actually factors the map into three rather than two pieces, distinguishing between terminal states (representing goals), and nonterminal states (those that may be traversed on the way to goals); and dividing the map into one matrix encoding long-run interconnectivity between nonterminal states (the DR, $\mathbf{M}$) and a second matrix representing one-step connections from nonterminal states to goals ($\mathbf{P}$). This is a necessary restriction, in that only for this type of finite decision problem are the optimal values linearly computable. However, this classification is not inflexible, because we also introduce novel techniques (based on matrix inversion lemmas) that allow dynamically changing which states are classed as goals. This allows the model (for example) to plan the best route to any arbitrary location in a maze (Fig 2d). Representing goals as terminal states also means that the model does not directly solve problems that require figuring out how best to visit multiple goals in sequence. However, this restriction has little effect in practice because these can either be treated as a series of episodes, re-starting at each goal, or by including multiple goals within the nonterminal states, since the model does optimize aggregate rewards over trajectories through nonterminal states as well.

This last point raises several interesting directions for future work. First, although there is evidence that humans choose their goal and plan towards that goal[52], there is some empirically underconstrained theoretical flexibility in specifying how a task's state space should be partitioned into terminal and nonterminal states. For the simulations here, we have tentatively adopted the principle that all discrete, punctate outcomes (like food or shock) are represented as terminal goal states with corresponding value in $\mathbf{r}$, and the rest of the (nonterminal) states contain only costs, constant everywhere, meant to capture the cost of locomotion. But, in general, state-dependent costs (or indeed rewards) can be included for nonterminal states as well. These in effect modulate the "distance" between states represented within the DR (see *Methods*). Nevertheless, this leads to the testable prediction that to whatever extent state-specific rewards or costs are accounted for within nonterminal states, they should affect hypothetical neural representations of the DR, such as grid cells. For instance, unlike for the SR, the DR predicts that by increasing locomotion cost, hills or rough terrain should increase "distance" as measured in the grid map. This aspect of the DR may be relevant for explaining recent evidence that grid cells have some subtle sensitivities to reward[53,54] which cannot be explained, as the SR-eigenvector account would predict, as secondary to changes in behavioral policy (e.g., not due to occupancy around rewarding locations[54], nor variations in trajectories or speed[53]).

Linear RL requires one other formal restriction on tasks, compared to standard Markov decision processes as often assumed by other RL theories in theoretical neuroscience. This is that the task is deterministically controllable. This is a good fit for many important sequential tasks, such as spatial navigation (I can reliably get from location A to location B by taking a step forward) and instrumental lever-pressing, but does not directly or exactly map to tasks that include irreducibly stochastic state transitions, such as two-step noisy Markov decision tasks that we and others have used to study model-based planning[55]. Such tasks can also be addressed via a further step of approximation[21], but it remains for future work to explore how far this can be pushed.

We have stressed how the DR can be used for planning, and also how it embodies substantial, reusable computation (specifically, predictions of long-run future state occupancy and cost-to-go), relative to simpler, easy-to-learn map representations like the one-step state adjacency model $P(s_{t+1}|s_t)$. We have not, so far, discussed how the DR can itself be learned or computed. There are several possibilities: two inherited from previous work on the SR[15] and one newly introduced here. First, like the SR, the DR can be learned gradually by actual or replay-based sampling of the environment, using a temporal difference rule[14,15]. Second, again like the SR, the DR can be constructed from the one-step transition matrix and costs (which can themselves be learned directly by Hebbian learning) by a matrix inversion, or equivalently a sum over a series of powers of a matrix. The latter form motivates attractor methods for computing the inverse iteratively by a simple recurrent network[15,56,57].

A third possibility for learning the DR follows from the novel method we introduce for using matrix inversion identities to efficiently update the DR in place to add additional goals, barriers, or shortcuts (see *Methods*). This works by expressing the inverse matrix in terms of the inverses of simpler component matrices (one of which is the pre-update DR), rather than for instance by updating the transition matrix and then, expensively, re-inverting the whole thing. For instance, we used this to solve tasks, such as Tolman's detour task, in which the transition structure of the environment changes. It could also be used, state by state or barrier by barrier, as a learning rule for building up the DR from scratch. Suggestively, this insight that the Woodbury matrix inversion identity can be used to decompose a DR map (an inverse matrix) into the sum of component maps, each associated with different sub-graphs of the transition space, offers a very promising direction for a direct neural implementation for representing and constructing maps componentially: via summing basis functions. This idea dovetails with – and may help to formalize and extend – the emerging idea that maps in the brain are built up by composing basis functions, such as those putatively represented in the grid cells[7,22,26,35,58]. Here, we showed that the term required to update the DR when encountering a wall remarkably resembles entorhinal border cells[37]. Therefore, our theory unifies the functional roles of entorhinal grid and border cells in planning and navigation, both as basis functions for building up the map. It remains for future work to explore the extent that this technique can be used to account for other aspects of neural coding in the entorhinal cortex. With respect to the grid cells, we also note that just as for the graph Laplacian and SR[22,23], the eigenvectors of the DR capture the periodicity and multiscale aspect of the grid cell code, but only a subset of them exhibit hexagonal symmetry. Additional constraints, such as nonnegativity[58] are likely required for a more detailed model.

Our model is based on the notion of the default policy, which is a map of expected state-to-state transition probability regardless of the current goals. Unlike previous RL models, such as the SR, linear RL does not entirely rely on the default policy and instead optimizes the decision policy around the default policy. This means that the final optimized policy is between the exact, deterministic optimized policy, and the default. The degree of this bias is controlled by a free parameter that scales the control costs relative to rewards and corresponds to the temperature in the softmax approximation to the optimization. In the limit of zero, or respectively infinite, control cost scaling, the approximation to the optimum becomes exact, or the default policy dominates completely. How should this parameter be set, and why not always take it near zero to improve the fidelity of the approximation? Linear RL works by multiplying small numbers (future occupancies) times large numbers (exponentiated, scaled rewards) to approximate the maximum expected value; just as with numerical precision in computers, there are issues of bandwidth (e.g. maximum spike rate and quantization) and gain control for making this work effectively across different decision situations in the brain. This suggests fruitful connections (for future work) with gain control and normalization[59], and rational models for choice using noisy representations[60,61]. The same tradeoff can also be understood from a Bayesian planning as inference perspective[62], in which the default policy plays the role of prior over policy space and rewards play the role of the likelihood function. In this case, the decision policy is the posterior that optimally combines them[50]. Then, how much the decision policy should be influenced by the default depends on how informative a prior it is (e.g. how reliable or uncertain it has been previously). This also suggests another distinct perspective on the default policy's role, in the model, in producing prepotent biases that can be overcome by cognitive control[38,42]. On this view, it serves to regularize behavior toward policies that have worked reliably in the past; and deviations from this baseline are costly.

Indeed, our framework leaves open not just how strongly the default policy is emphasized, but also how it is learned or chosen. In general, while the model provides a good approximation to the true optimal values independent of which default policy is used (so long as its cost is scaled appropriately relative to the rewards), we can also ask the converse question – which default policy should be chosen to allow for the best approximation and thereby obtain the most (actual) reward? The answer is of course, that the cost term (measuring the divergence between true and approximate $\mathbf{v}^*$) is minimized whenever the future π^* is equal to the default π^d. Any algorithm for learning policies might be appropriate, then, for finding a π^d that is likely to be near-optimal in the future, including in particular previous habit learning models, including model-free actor-critic learning[63] or even non-reward-driven memorization of previous policies[64]. A related idea has also been recently proposed in the context of a more explicitly hierarchical model of policy learning: that a default policy (and control-like charges for deviation form it) can be useful in the context of multitask learning to extract useful, reusable policies[65,66]. Separately, an analogous principle of identification of task structure that generalizes across tasks in a hierarchical generative model has also been proposed as a model of grid and place cell responses[7,26]. Future work remains to understand the relationship between the considerations in both of these models – which involve identifying shared structure – and ours, which are motivated instead more by efficiently reusing computation.

The role of the default policy, finally, points at how the linear RL framework provides a richer, more nuanced view of habits and pathologies of decision making than previous computational theories. Although a learned default policy biases behavior, and may modulate accuracy or speed of performance, it trades off against rewards in the optimization. This give and take stands in contrast to much previous work, especially in computational psychiatry, which has often assumed a binary model of evaluation: either flexible values are computed (model-based, goal-directed) or they are not (model-free, habits). The latter, acting rather than thinking, has been taken as a model of both healthy and unhealthy habits, and especially of compulsive symptoms such as in drug abuse[48] and obsessive compulsive disorder[67]. Although such outright stimulus-response behaviors may exist, the present framework allows for a much broader range of biases and tendencies, and may help to understand a greater range of symptomatology, such as excessive avoidance in anxiety[68], craving and cue-induced relapse in drug abuse, and the ability to effortfully suppress compulsive behaviors across many disorders. Finally, and relatedly, the possibility of a dynamic and situation-dependent default policy also offers a way to capture some aspects of emotion that have been resistant to RL modeling. In particular, one important aspect of emotion is its ability to elicit a pattern of congruent response tendencies, such as a greater tendency toward aggression when angry. Complementing recent work suggesting these might arise due to a hard bias on planning (via pruning context-inappropriate actions)[69], the default policy offers a clear and normative lever for influencing behavior on the basis of emotional (and other) context.

Methods

Model description

In this work, we focus on Markov decision processes with two conditions. First, we assume that there is one or a set of terminal states, s_T; Second, we only consider deterministic environments, such as mazes, in which there is a one-to-one map between actions and successor states.

The linear RL model is then based on a modification to the value function for this setting[20,21], in which the agent controls the probabilistic distribution over successor states (i.e., actions) and pays an additional control cost quantified as the dissimilarity (in the form of KL divergence) between the controlled dynamics (i.e. decision policy), $\pi(.\,|s_t)$ and a default dynamics, $\pi^d(.\,|s_t)$. In particular, the objective of this MDP is to optimize a "gain" function, $g(s_t)$, defined as

$$g(s_t) = r(s_t) - \lambda \mathrm{KL}(\pi||\pi^d) \tag{4}$$

where $\lambda > 0$ is a constant and $\mathrm{KL}(\pi||\pi^d)$ is the KL divergence between the two probability distributions; it is only zero if the two distributions are the same, i.e. $\pi = \pi^d$ and otherwise is positive. We also require that $\pi = 0$ if $\pi^d = 0$. Note that in the limit of zero, or respectively infinite, λ, the gain converges to pure reward (i.e. a standard MDP), or pure cost. Here, λ scales the relative strength of control costs in units of reward (and is equivalent to rescaling the units of reward while holding the cost fixed).

It is easy then to show that the optimal value function for this new problem, $\mathbf{v}^*$, is analytically solvable[20,21] (see formal derivation below). We first define the one-step state transition matrix $\mathbf{T}$, whose (i,j) element is equal to the probability of transitioning from state i to state j under the default policy (i.e. probability of the action under the default policy that makes $i \rightarrow j$ transition). This contains subblocks, $\mathbf{T}_{NN}$, the transition probability between nonterminal states, and $\mathbf{T}_{NT} = \mathbf{P}$, the transition probabilities from terminal to nonterminal states. Then:

$$\exp(\mathbf{v}^*/\lambda) = \mathbf{MP}\exp(\mathbf{r}/\lambda), \tag{5}$$

where $\mathbf{v}^*$ is the vector of optimal values at nonterminal states, $\mathbf{r}$ is the vector of rewards at terminal states, and $\mathbf{M}$ is a matrix defined below. Note that equation (3) is the case of this equation for $\lambda = 1$.

The DR matrix $\mathbf{M}$ is defined as:

$$\mathbf{M} = (\mathrm{diag}(\exp(-\mathbf{r}_N/\lambda)) - \mathbf{T}_{NN})^{-1},$$

where $\mathbf{r}_N$ is the vector of rewards at nonterminal states (which we take as a uniform cost of -1 in most of our simulations).

For flexibility in updating which states are viewed as goal states, it is helpful to define a second, more general version of the DR matrix, $\mathbf{D}$, defined over all states (not just nonterminal states) as:

$$\mathbf{D} = (\mathrm{diag}(\exp(-\mathbf{r}_A/\lambda)) - \mathbf{T})^{-1},$$

where $\mathbf{r}_A$ is the reward vector across all states. Note that since matrix $\mathbf{M}$ can be easily computed from $\mathbf{D}$ (in particular, $\mathbf{M}$ is a subblock of **D** corresponding to the nonterminal states only), we refer to both of them as the DR unless specified otherwise. Also note that for defining $\mathbf{D}$, we assumed, without loss of generality (since this assumption does not affect $\mathbf{M}$), that reward at terminal states are not 0.

This solution for $\mathbf{v}^*$ further implies that the policy takes the form of a weighted softmax, where the weights are given by the default policy

$$\pi(a|s_t) = \frac{\pi^d(a|s_t)\exp(v^*(s_a)/\lambda)}{\sum_{a'} \pi^d(a'|s_t)\exp(v^*(s_{a'})/\lambda)} \tag{6}$$

where s_a is the successor state associated with action a. Thus, for a uniform default policy, the optimal policy is simply given by the softmax over optimal values with the temperature parameter λ. Note also that in the limit of $\lambda = 0$, the problem becomes the classical MDP (because $g(s_t) = r(s_t)$ in equation (4)) and the decision policy in equation (6) also reflects the optimum policy (i.e. greedy) exactly. In the limit of infinite λ, the influence of the rewards vanishes and the decision policy converges to the default policy.

Planning toward a new goal and transfer revaluation

Consider an environment with $\mathbf{T_0}$ and $\mathbf{D_0}$ as the transition matrix under the default policy and the associated DR, respectively. Now suppose that the agent's goal is to plan toward state j (or equivalently computing the distance between any state and j), i.e., we wish to add j to the set of terminal states. Here, we aim to develop an efficient method to plan towards j by using the cached $\mathbf{D_0}$, without re-inverting the matrix.

If we define $\mathbf{L_0} = \mathrm{diag}(\exp(-\mathbf{r}_A/\lambda)) - \mathbf{T}_0$ and $\mathbf{L} = \mathrm{diag}(\exp(-\mathbf{r}_A/\lambda)) - \mathbf{T}$, then $\mathbf{L}$ and $\mathbf{L_0}$ are only different in their jth row (because $\mathbf{T}$ and $\mathbf{T_0}$ are only different in their jth row). We define $\mathbf{d}$, a row-vector corresponding to the difference in jth row of the two matrices:

$$\mathbf{d} = \mathbf{L}(j,:) - \mathbf{L}_0(j,:),$$

and therefore, we can write:

$$\mathbf{L} = \mathbf{L_0} + \mathbf{de},$$

where $\mathbf{e}$ is a binary column-vector that is one only on jth element. Using the Woodbury matrix identity, $\mathbf{L}^{-1}$ is given by

$$\mathbf{L}^{-1} = \mathbf{L_0^{-1}} - \frac{1}{1 + \mathbf{dL_0^{-1}e}} \mathbf{L_0^{-1}edL_0^{-1}},$$

in which we exploited the fact that $\mathbf{d}$ and $\mathbf{e}$ are row- and column- vectors, respectively, and therefore $\mathbf{dL_0^{-1}e}$ is a scalar. Since $\mathbf{D_0} = \mathbf{L_0^{-1}}$ and $\mathbf{D} = \mathbf{L}^{-1}$, we obtain

$$\mathbf{D} = \mathbf{D_0} - \frac{1}{1 + \mathbf{dm_0}} \mathbf{m_0 d D_0}, \quad (7)$$

where $\mathbf{m_0}$ is the jth column of $\mathbf{D_0}$.

The above equation represents an efficient, low-rank update to the DR itself. However, for the purpose of this single planning problem (e.g. if, we do not intend further modifications to the matrix later), we may also further simplify the computation by focusing only on the product $\mathbf{z} = \mathbf{MP}$, which is what is needed for planning using equation (5) in the new environment. We find $\mathbf{z}$ in terms of an intermediate vector $\hat{\mathbf{z}} = \mathbf{D}\widehat{\mathbf{P}}$, where $\widehat{\mathbf{P}}$ is a subblock of $\mathbf{T}$ from all states to terminal states, in which all elements of rows corresponding to terminal states are set to 0. Therefore, $\hat{\mathbf{z}}$ is given by

$$\hat{\mathbf{z}} = \mathbf{z_0} - \frac{1}{1 + \mathbf{dm_0}} \mathbf{m_0 d z_0}, \quad (8)$$

where

$$\mathbf{z_0} = \mathbf{D_0}\widehat{\mathbf{P}}. \quad (9)$$

Finally, $\mathbf{z}$ is given by the submatrix of $\hat{\mathbf{z}}$ corresponding to nonterminal rows.

It is important to note that since $\mathbf{d}$ and $\widehat{\mathbf{P}}$ are very sparse, computations in equations (8-9) are local. In fact, $\mathbf{d}$ is only nonzero on elements associated with immediate state of j (and jth element). If we assume that there is only one terminal state (i.e. j), then $\widehat{\mathbf{P}}$ is a vector that is nonzero on elements associated with immediate state of j.

The same technique can be used to update the DR or re-plan in transfer revaluation problems, such as localized changes in $\mathbf{T}_{NN}$ or $\mathbf{P}$. For example, if transition from state j to i has been blocked, new values for $\mathbf{D}$ and $\mathbf{z}$ can be computed efficiently using equations (7) and (8), respectively. Similarly, $\mathbf{D}$ and $\mathbf{z}$ can be computed efficiently using those equations if the reward value for the nonterminal state changes. Finally, it is also possible to learn the DR matrix, transition by transition, by iteratively computing $\mathbf{D}$ for each update using $\mathbf{D}_0$ in equation (7).

Border cells

We employed a similar approach to account for border cells. Suppose that a wall has been inserted into the environment, which changes the transition matrix $\mathbf{T}_0$ to $\mathbf{T}$. Suppose $\mathbf{L}_\mathbf{0} = \mathrm{diag}(\exp(-\mathbf{r}_A/\lambda)) - \mathbf{T}_0$ and $\mathbf{L} = \mathrm{diag}(\exp(-\mathbf{r}_A/\lambda)) - \mathbf{T}$. We define matrix $\boldsymbol{\Delta}$ using rows of $\mathbf{L}_\mathbf{0}$ and $\mathbf{L}$ corresponding to J:

$$\boldsymbol{\Delta} = \mathbf{L}_J - \mathbf{L}_{0J},$$

where J denotes those states that their transition has been changed, $\mathbf{L}_J$ and $\mathbf{L}_{0J}$, are, respectively, submatrices associated with rows of $\mathbf{L}$ and $\mathbf{L}_\mathbf{0}$ corresponding to J. Using the Woodbury matrix identity (similar to equation (7)), the DR associated with the new environment is given by

$$\mathbf{D} = \mathbf{D}_\mathbf{0} - \mathbf{B},$$

where

$$\mathbf{B} = \mathbf{D}_{\mathbf{0}J}\left(\mathbf{I} + \boldsymbol{\Delta}\mathbf{D}_{\mathbf{0}J}\right)^{-1}\boldsymbol{\Delta}\mathbf{D}_\mathbf{0},$$

in which matrix $\mathbf{D}_{\mathbf{0}J}$ is the submatrix associated with columns of $\mathbf{D}_\mathbf{0}$ corresponding to J, and $\mathbf{I}$ is the identity matrix. Note that although this model requires inverting of a matrix, this computation is substantially easier than inverting matrix $\mathbf{L}$, because this matrix is low-dimensional. For simulating the border cells in Fig 5, we replaced matrix $\mathbf{D}_\mathbf{0}$ by its eigenvectors. Thus, if $\mathbf{u}$ is an eigenvector of $\mathbf{D}_\mathbf{0}$, the corresponding column in $\mathbf{B}$, $\mathbf{b}(\mathbf{u})$ is given by

$$\mathbf{b}(\mathbf{u}) = \mathbf{D}_{\mathbf{0}J}\left(\mathbf{I} + \boldsymbol{\Delta}\mathbf{D}_{\mathbf{0}J}\right)^{-1}\boldsymbol{\Delta}\mathbf{u}.$$

Simulation details

We assumed a uniform default policy in all analyses presented in Figure 1-5. In Fig 1, the cost for all states were randomly generated in the range of 0 to 10 and analysis was repeated 100 times. In Fig 2b-c, a 50x50 maze environment was considered. In Fig 2d-e, a 10x10 maze was considered with 20 blocked states. The DR was computed in this environment with no terminal state, in which the cost for all states was 1. We used equation (8) to compute the shortest path using linear RL. The optimal path between every two states was computed by classic value iteration algorithm. In Fig 3b-c, the reward of all states was –1, except the terminal states, which was +5. In the revaluation phase, the reward of the left terminal state was set to –5. In Fig 3d, the reward of states 1,2 and 3 is 0. In Fig 3e, reward at all states is –1, except for the terminal state, which is +5. In Fig 4d, a 50x50 maze was considered, the cost for all states was assumed to be 0.1. In this figure, 15th, 20th, 32th eigenvectors of the DR have been plotted. In Fig 5b, a 20x20 maze was considered and the cost for all states was assumed to be 0.1. In this figure, 1th, 6th, 11th, 12th eigenvectors of the DR have been considered.

The default policy in Figs 6-7 was not uniform. In Fig 6c, the default probability for the control-demanding action assumed to be 0.2 and reward was assumed to be +2. For simulating PIT in Fig 7, we followed experimental design of Corbit et al.[49] and assumed that the environment contains 4 states, in which state 1 was the choice state, states 2, 3, and 4 were associated with outcomes 1,2 and 3, respectively. In Fig 7c, the reward of outcome 1-3 was +5. In Fig 7e, the reward of all states was assumed to be 0. It was also assumed that during the Pavlovian training, the default probability for Stimulus 1→ Outcome 1 and for Stimulus 2→ Outcome 2 changes from 0.33 (i.e. uniform) to 0.5.

The only parameter of linear RL is λ, which was always assumed to be 1, except for simulating the results presented in Fig 3e, where we set $\lambda = 10$ to avoid overflow of the exponential due to large reward values.

Formal derivation

For completeness, we present derivation of equations (5-6) based on Todorov[20,21]. By substituting the gain defined in equation (4) into the Bellman equation (1), we obtain:

$$v(s_t) = r(s_t) + \max_{\pi}\left\{-\lambda E_{a\sim\pi(a|s_t)}\left[\log\frac{\pi(a|s_t)}{\pi^d(a|s_t)\exp\left(v(s_a)/\lambda\right)}\right]\right\},$$

where s_a denotes the corresponding state (among the set of successor states of s_t) to action a.

Note that the expectation in the Bellman equation is under the dynamics, which we have replaced it with the policy because they are equivalent here. The expression being optimized in this equation is akin to a KL divergence, except that the denominator in the argument of the log function is not normalized. Therefore, we define the normalization term c:

$$c = \sum_a \pi^d(a|s_t)e^{v(s_a)/\lambda},$$

Note that c is independent of the distribution being optimized π. By multiplying and dividing the denominator of the log by c, we obtain:

$$v(s_t) = r(s_t) + \lambda\log c + \max_{\pi}\{-\lambda\mathrm{KL}(\pi(a|s_t)||\pi^d(a|s_t)e^{v(s_a)/\lambda}/c)\},$$

where the maximum value of negative KL divergence is zero, which occurs only if the two distributions are equal, giving rise to equation (6):

$$\pi(a|s_t) = \pi^d(a|s_t)e^{v(s_a)/\lambda}/c.$$

Furthermore, since the KL divergence is zero, optimal values satisfy:

$$v^*(s_t) = r(s_t) + \lambda\log c.$$

Across all states, this gives rise to a system of linear equations in the exponential space. Since at terminal states, $v(s_T) = r(s_T)$, this system can be solved analytically, which can be written in the matrix equation 5.

Acknowledgement

We thank Tim Behrens and Jon Cohen for helpful discussions. This work was supported by grants IIS-1822571 from the National Science Foundation, part of the CRNCS program, and 61454 from the John Templeton Foundation.

References

1. Bellman, R. E. *Dynamic Programming*. (Princeton University Press, 1957).
2. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction*. (MIT Press, 2018).
3. Dickinson, A. & Balleine, B. W. The role of learning in motivation. in *Volume 3 of Steven's Handbook of Experimental Psychology: Learning, Motivation, and Emotion* (ed. Gallistel, C. R.) 497–533 (Wiley, 2002).
4. Wimmer, G. E. & Shohamy, D. Preference by association: how memory mechanisms in the hippocampus bias decisions. *Science* **338**, 270–273 (2012).
5. Momennejad, I. *et al.* The successor representation in human reinforcement learning. *Nat. Hum. Behav.* **1**, 680–692 (2017).
6. Tolman, E. C. Cognitive maps in rats and men. *Psychol. Rev.* **55**, 189–208 (1948).
7. Behrens, T. E. J. *et al.* What Is a Cognitive Map? Organizing Knowledge for Flexible Behavior. *Neuron* **100**, 490–509 (2018).
8. Daw, N. D., Niv, Y. & Dayan, P. Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nat. Neurosci.* **8**, 1704–1711 (2005).
9. Keramati, M., Dezfouli, A. & Piray, P. Speed/accuracy trade-off between the habitual and the goal-directed processes. *PLoS Comput. Biol.* **7**, e1002055 (2011).
10. Daw, N. D. & Dayan, P. The algorithmic anatomy of model-based evaluation. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* **369**, (2014).
11. Mattar, M. G. & Daw, N. D. Prioritized memory access explains planning and hippocampal replay. *Nat. Neurosci.* **21**, 1609–1617 (2018).
12. Keramati, M., Smittenaar, P., Dolan, R. J. & Dayan, P. Adaptive integration of habits into depth-limited planning defines a habitual-goal-directed spectrum. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 12868–12873 (2016).
13. Huys, Q. J. M. *et al.* Interplay of approximate planning strategies. *Proc. Natl. Acad. Sci. U. S. A.* **112**, 3098–3103 (2015).
14. Dayan, P. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Comput.* **5**, 613–624 (1993).
15. Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J. & Daw, N. D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Comput. Biol.* **13**, e1005768 (2017).
16. Lehnert, L., Tellex, S. & Littman, M. L. Advantages and Limitations of using Successor Features for Transfer in Reinforcement Learning. *ArXiv170800102 Cs Stat* (2017).
17. Dezfouli, A. & Balleine, B. W. Habits, action sequences and reinforcement learning. *Eur. J. Neurosci.* **35**, 1036–1051 (2012).
18. Botvinick, M. M., Niv, Y. & Barto, A. C. Hierarchically organized behavior and its neural foundations: A reinforcement-learning perspective. *Cognition* **113**, 262–280 (2009).
19. Kappen, H. J. Linear theory for control of nonlinear stochastic systems. *Phys. Rev. Lett.* **95**, 200201 (2005).
20. Todorov, E. Linearly-solvable Markov decision problems. in *Advances in Neural Information Processing Systems 19* (eds. Schölkopf, B., Platt, J. C. & Hoffman, T.) 1369–1376 (MIT Press, 2007).
21. Todorov, E. Efficient computation of optimal actions. *Proc. Natl. Acad. Sci. U. S. A.* **106**, 11478–11483 (2009).
22. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20**, 1643–1653 (2017).
23. Gustafson, N. J. & Daw, N. D. Grid Cells, Place Cells, and Geodesic Generalization for Spatial Reinforcement Learning. *PLOS Comput. Biol.* **7**, e1002235 (2011).
24. Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E. I. Microstructure of a spatial map in the entorhinal cortex. *Nature* **436**, 801–806 (2005).
25. Constantinescu, A. O., O'Reilly, J. X. & Behrens, T. E. J. Organizing conceptual knowledge in humans with a gridlike code. *Science* **352**, 1464–1468 (2016).
26. Whittington, J. C. *et al.* The Tolman-Eichenbaum Machine: Unifying space and relational memory through generalisation in the hippocampal formation. *bioRxiv* 770495 (2019) doi:10.1101/770495.
27. Mahadevan, S. Representation Policy Iteration. *ArXiv12071408 Cs* (2012).
28. Mahadevan, S. & Maggioni, M. Proto-value Functions: A Laplacian Framework for Learning Representation and Control in Markov Decision Processes. *J. Mach. Learn. Res.* **8**, 2169–2231 (2007).
29. Sanguinetti-Scheck, J. I. & Brecht, M. Home, head direction stability and grid cell distortion. *bioRxiv* 602771 (2019) doi:10.1101/602771.
30. Carpenter, F., Manson, D., Jeffery, K., Burgess, N. & Barry, C. Grid cells form a global representation of connected environments. *Curr. Biol. CB* **25**, 1176–1182 (2015).

31. Derdikman, D. *et al.* Fragmentation of grid cell maps in a multicompartment environment. *Nat. Neurosci.* **12**, 1325–1332 (2009).
32. de Wit, S., Niry, D., Wariyar, R., Aitken, M. R. F. & Dickinson, A. Stimulus-outcome interactions during instrumental discrimination learning by rats and humans. *J. Exp. Psychol. Anim. Behav. Process.* **33**, 1–11 (2007).
33. Tolman, E. C. & Gleitman, H. Studies in learning and motivation; equal reinforcements in both end-boxes; followed by shock in one end-box. *J. Exp. Psychol.* **39**, 810–819 (1949).
34. Sutton, R. S. TD Models: Modeling the World at a Mixture of Time Scales. in *Machine Learning Proceedings 1995* (eds. Prieditis, A. & Russell, S.) 531–539 (Morgan Kaufmann, 1995). doi:10.1016/B978-1-55860-377-6.50072-4.
35. Baram, A. B., Muller, T. H., Whittington, J. C. R. & Behrens, T. E. J. Intuitive planning: global navigation through cognitive maps based on grid-like codes. *bioRxiv* 421461 (2018) doi:10.1101/421461.
36. He, Q. & Brown, T. I. Environmental Barriers Disrupt Grid-like Representations in Humans during Navigation. *Curr. Biol. CB* **29**, 2718-2722.e3 (2019).
37. Solstad, T., Boccara, C. N., Kropff, E., Moser, M.-B. & Moser, E. I. Representation of geometric borders in the entorhinal cortex. *Science* **322**, 1865–1868 (2008).
38. Botvinick, M. M. & Cohen, J. D. The computational and neural basis of cognitive control: charted territory and new frontiers. *Cogn. Sci.* **38**, 1249–1285 (2014).
39. Cohen, J. D., Dunbar, K. & McClelland, J. L. On the control of automatic processes: a parallel distributed processing account of the Stroop effect. *Psychol. Rev.* **97**, 332–361 (1990).
40. Kool, W., McGuire, J. T., Rosen, Z. B. & Botvinick, M. M. Decision making and the avoidance of cognitive demand. *J. Exp. Psychol. Gen.* **139**, 665–682 (2010).
41. Westbrook, A., Kester, D. & Braver, T. S. What Is the Subjective Cost of Cognitive Effort? Load, Trait, and Aging Effects Revealed by Economic Preference. *PLOS ONE* **8**, e68210 (2013).
42. Shenhav, A. *et al.* Toward a Rational and Mechanistic Account of Mental Effort. *Annu. Rev. Neurosci.* **40**, 99–124 (2017).
43. Kurzban, R., Duckworth, A., Kable, J. W. & Myers, J. An opportunity cost model of subjective effort and task performance. *Behav. Brain Sci.* **36**, (2013).
44. Botvinick, M. & Braver, T. Motivation and cognitive control: from behavior to neural mechanism. *Annu. Rev. Psychol.* **66**, 83–113 (2015).
45. Krebs, R. M., Boehler, C. N. & Woldorff, M. G. The influence of reward associations on conflict processing in the Stroop task. *Cognition* **117**, 341–347 (2010).
46. Dickinson, A. & Balleine, B. Motivational control of goal-directed action. *Anim. Learn. Behav.* **22**, 1–18 (1994).
47. Corbit, L. H. & Balleine, B. W. Learning and Motivational Processes Contributing to Pavlovian–Instrumental Transfer and Their Neural Bases: Dopamine and Beyond. in *Behavioral Neuroscience of Motivation* (eds. Simpson, E. H. & Balsam, P. D.) 259–289 (Springer International Publishing, 2016). doi:10.1007/7854_2015_388.
48. Everitt, B. J. & Robbins, T. W. Drug Addiction: Updating Actions to Habits to Compulsions Ten Years On. *Annu. Rev. Psychol.* **67**, 23–50 (2016).
49. Corbit, L. H., Janak, P. H. & Balleine, B. W. General and outcome-specific forms of Pavlovian-instrumental transfer: the effect of shifts in motivational state and inactivation of the ventral tegmental area. *Eur. J. Neurosci.* **26**, 3141–3149 (2007).
50. Todorov, E. General duality between optimal control and estimation. in *2008 47th IEEE Conference on Decision and Control* 4286–4292 (2008). doi:10.1109/CDC.2008.4739438.
51. Gershman, S. J. The Successor Representation: Its Computational Logic and Neural Substrates. *J. Neurosci.* **38**, 7193–7200 (2018).
52. Cushman, F. & Morris, A. Habitual control of goal selection in humans. *Proc. Natl. Acad. Sci. U. S. A.* **112**, 13817–13822 (2015).
53. Boccara, C. N., Nardin, M., Stella, F., O'Neill, J. & Csicsvari, J. The entorhinal cognitive map is attracted to goals. *Science* **363**, 1443–1447 (2019).
54. Butler, W. N., Hardcastle, K. & Giocomo, L. M. Remembered reward locations restructure entorhinal spatial maps. *Science* **363**, 1447–1452 (2019).
55. Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P. & Dolan, R. J. Model-based influences on humans' choices and striatal prediction errors. *Neuron* **69**, 1204–1215 (2011).
56. Sutton, R. S. & Pinette, B. The learning of world models by connectionist networks. in *Seventh Annual Conference of the Cognitive Science Society* 54–64 (1985).
57. Jang, J.-S., Lee, S.-Y. & Shin, S.-Y. An Optimization Network for Matrix Inversion. in 397–401 (1988).

58. Dordek, Y., Soudry, D., Meir, R. & Derdikman, D. Extracting grid cell characteristics from place cell inputs using non-negative principal component analysis. *eLife* **5**, e10094 (2016).
59. Louie, K., Grattan, L. E. & Glimcher, P. W. Reward value-based gain control: divisive normalization in parietal cortex. *J. Neurosci. Off. J. Soc. Neurosci.* **31**, 10627–10639 (2011).
60. Gershman, S. & Wilson, R. The Neural Costs of Optimal Control. in *Advances in Neural Information Processing Systems 23* (eds. Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S. & Culotta, A.) 712–720 (Curran Associates, Inc., 2010).
61. Woodford, M. Prospect Theory as Efficient Perceptual Distortion. *Am. Econ. Rev.* **102**, 41–46 (2012).
62. Botvinick, M. & Toussaint, M. Planning as inference. *Trends Cogn. Sci.* **16**, 485–488 (2012).
63. Barto, A. G. Adaptive critic and the basal ganglia. in *Models of information processing in the basal ganglia* (eds. Houk, J. C., Davis, J. L. & Beiser, D. G.) 215–232 (MIT Press, 1995).
64. Miller, K. J., Shenhav, A. & Ludvig, E. A. Habits without values. *Psychol. Rev.* **126**, 292–311 (2019).
65. Teh, Y. *et al.* Distral: Robust multitask reinforcement learning. in *Advances in Neural Information Processing Systems 30* (eds. Guyon, I. et al.) 4496–4506 (Curran Associates, Inc., 2017).
66. Kool, W. & Botvinick, M. Mental labour. *Nat. Hum. Behav.* **2**, 899–908 (2018).
67. Gillan, C. M., Kosinski, M., Whelan, R., Phelps, E. A. & Daw, N. D. Characterizing a psychiatric symptom dimension related to deficits in goal-directed control. *eLife* **5**, (2016).
68. Zorowitz, S., Momennejad, I. & Daw, N. D. Anxiety, avoidance, and sequential evaluation. *bioRxiv* 724492 (2019) doi:10.1101/724492.
69. Huys, Q. J. M. & Renz, D. A Formal Valuation Framework for Emotions and Their Control. *Biol. Psychiatry* **82**, 413–420 (2017).