

Reinforcement Learning and Know-How

Paul Talma

March 11, 2025

1 Introduction

The nature of know-how has occupied a prominent place in contemporary philosophy of mind and action. Much of the literature focuses on a question raised by [CITE: Ryle]’s criticisms of *intellectualism*, the view that know-how centrally involves a kind of propositional knowledge. *Anti-intellectualists* (such as Ryle) deny this.

My goal in this paper is to demonstrate that computational reinforcement learning provides a range of tools for probing the nature of knowledge-how. I will also argue that it provides compelling counterexamples to the intellectualist thesis. But I think that the former contribution is more important: our practical capacities exhibit rich and variegated structures; it matters more to understand these structures on their own terms than to decide whether know-how is, always and everywhere, a form of know-that. So, at any rate, I will argue.

Still, the debate between intellectualists and anti-intellectualists provides a nice place to start, so I will begin by laying out the two positions. Then, I will introduce the basics of reinforcement learning, sharing technical details on a need-to-know basis. With those basics on the table, I will illustrate some central uses of reinforcement learning in the cognitive and neural sciences, with an eye to their bearing on the intellectualist position. After explaining why I take these cases to refute intellectualism and considering rejoinders, I go on to illustrate how computational concepts from reinforcement learning can shed light on various aspects of know-how, abilities, and skills, such as their relation to chunking and motor schemata, the distinction between habitual and goal-directed behavior, the phenomenon of automatization, and ...

2 Intellectualism and Anti-Intellectualism

In this section, I will endeavor to clarify the two central positions. This is not a trivial task. The literature has revolved around three concepts: *skill*, *ability*, and *know-how*, and their relation to propositional knowledge. Various authors seek to identify one or more of these phenomena. For example, [CITE: Noe], following [CITE: Ryle], maintains that know-how and ability are one. [CITE: Stanley and Williamson] deny this, and maintain instead that know-how is a species of propositional knowledge (which anti-intellectualists deny). Some, like [CITE: Stanley and Krakauer], identify skill and

know-how but divorce them from abilities, holding that one can know how to do something without being able to do it. And so on. It is therefore difficult to characterize the subject matter in a theory-neutral way (that is, in a way which wouldn't draw objections from at least one party). As such, I will talk of know-how, skills, and abilities more or less interchangeably, but without wishing to imply anything about their identity or distinctness. Let us proceed by way of examples.

Examples of the phenomena in question include my knowing how to swim, how to write a philosophy paper, how to get to the grocery store, how to play a video game, how to count to 10, and how to play the bass. To these we can add that Alva Noë's dog knows how to catch a Frisbee [CITE: Noe, 289] and that trained rats know how to navigate their way out of a maze. (Note that even Stanley and Williamson, the most prominent contemporary proponents of intellectualism, are happy to ascribe know-how to dogs [CITE: Stanley and Williamson].)

The first thing that should strike us about this collection of examples is its diversity: it is *ex ante* implausible, I think, that all of these examples fall under the same mental kind. Indeed, as I shall argue below, they do not: although there are good reasons to lump them together in ordinary talk, they are underwritten by importantly different cognitive mechanisms. Ordinary talk is (here as elsewhere) no guide to mental organization.

Be that as it may, here is the intellectualist thesis (φ ranges over actions):

INTELLECTUALISM: To know how to φ is to know, for some way w of doing φ , that w is a way of doing φ .

Note that knowing that w is a way of doing φ is knowledge of a proposition: it entails holding a propositional attitude (belief) toward the proposition that w is a way to do φ . Many intellectualists add that the proposition in question must feature the way w under a *practical mode of presentation* [CITE: Stanley and Williamson, Pavese]. Practical modes of presentation are supposed to be a species of Fregean modes of presentation. Modes of presentation support a fine-grained notion of mental content of the kind suitable for psychological and rational explanation [CITE: Fodor, Burge, Rescorla]. Practical modes of presentation index representational content to the exercise of practical capacities [CITE: Pavese]. Although (unlike some anti-intellectualists [CITE: Noe]) we do not consider the notion of a practical mode of presentation irredeemably obscure, our argument will not turn on it, and so we omit it from our discussion.

INTELLECTUALISM is, as stated, a very strong thesis. It implies not only that knowing how to do something requires a capacity for propositional thought, but also the possession of propositional *knowledge*. Moreover, having such knowledge is not merely necessary for possessing the relevant know how. It is constitutive of (and hence sufficient for) know-how. Not all who call themselves "intellectualists" will sign on to the thesis in its full strength ([CITE: XXXX], for example, requires mere belief rather than knowledge, and at times [CITE: Stanley and Krakauer] seem to think that propositional knowledge is merely necessary for, but perhaps not constitutive of, know-how). But, again, our arguments below will not turn on these features of the view, so we can let them stand.

It is important, however, to distinguish INTELLECTUALISM from a weaker thesis:

WAY-REPRESENTATIONALISM: knowing how to φ requires (or consists in) having some representation, whether propositionally structured or not, of a way to do φ .

This weaker thesis does not require knowers-how to be capable of propositional thought, though we may add that the way in question must be represented via a practical mode of presentation. Although I will argue this view should also be rejected, not all considerations against INTELLECTUALISM will apply to WAY-REPRESENTATIONALISM.

Finally, we must also distinguish the two foregoing views from

REPRESENTATIONALISM: knowledge-how requires some representations or other.

This view is much weaker. I do not know whether it is true, but I will suggest some tools for assessing it below.

Let me foreshadow my argument before introducing the necessary background. Drawing on case studies from computational cognitive science, I will argue that a wide range practical capacities in humans and animals—in particular, motor control and navigation—are explained by reinforcement learning. Although such explanations are, in principle, compatible with INTELLECTUALISM, in practice they are not. As I will argue, in cases of *model-based* reinforcement learning, the representations invoked to explain human and animal know-how are not propositionally structured. Hence, this know-how does not consist in propositional knowledge. These examples will suffice to refute INTELLECTUALISM. In addition, I will argue that the representations invoked to explain this know-how are not representations of ways of doing the relevant actions. Thus, these examples also serve to refute WAY-REPRESENTATIONALISM. In a later, more speculative section, I consider the bearing of *model-free* reinforcement learning explanations on REPRESENTATIONALISM. [Not sure what I'll say about that yet.] Before getting there, however, I will need to present the theoretical framework underlying the cognitive capacities in question. To this task we now turn.

3 Reinforcement Learning

Reinforcement learning is a set of mathematical and computational tools for handling sequential decision problems. As a branch of economics and operations research, it is closely related to the study of rational choice behavior and to theories of optimal control. As a branch of computer science, it sits alongside supervised and unsupervised learning as one of the methodological pillars of machine machine learning, and lies behind fundamental advances in robotics. And as a branch of cognitive science, it stands as one of the best-confirmed computational analyses of human and animal behavior, unifying a wide range of observations and enjoying precisely identified neural correlates.

In this section, I will provide an overview of the reinforcement learning framework and of its use in cognitive science.

3.1 Reinforcement Learning: the Basics

I begin with an informal gloss before introducing some technical notions. The following example does not illustrate all relevant features of reinforcement learning (no single example could), but should serve as a relatively concrete scenario onto which the technical concepts introduced later can be mapped. There will, of course, be further examples below.

Suppose that you are trapped in a cage and desirous to escape. Around you are various levers, buttons, ropes, springs, and other bells and whistles (your captor turns out to be a certain Rube Goldberg). For lack of an obvious way out of the cage, you haphazardly press some buttons and pull some ropes. Nothing happens. Then, you notice that one of the levers is connected to a latch; you press the lever before twisting a knob coupled to the latch, releasing a marble that rolls down a slide and dislodges an iron bar, and you make your escape. Unfortunately, a trapdoor opens beneath you, and after a short fall, you find yourself trapped once more in an identical cage. This time, however, you waste no time fiddling with the buttons and ropes: you go straight for the lever and the knob, and watch as the marble secures your escape once more. Unfortunately, an elaborate contraction of ropes, pulleys, and elastic bands catches you as you step out of the cage and transports you to yet another identical cage. This time, curious, you decide to twist the knob without first pressing the lever. To your surprise, this suffices to release the marble, which you now watch warily as it rolls down to unlock the door. Careful to avoid further any further traps, you make your way out and go on to confront Prof. Goldberg.

This example illustrates several features of the reinforcement learning problem. You are an agent, interacting with an environment. You have a goal, which is to escape. Achieving this goal requires performing specific sequences of actions. Each action may have some effect on your environment. However, the contribution of any individual action to your goal need not be clear. Initially, you try out various actions more or less at random. Having found, through trial and error or through careful observation of the cage mechanism, that a particular sequence of actions achieves your goal, you learn to reproduce that sequence in your next escape attempt: you do not need go through a trial-and-error process again. However, as your third stint in the cage shows, trial and error still has a role to play. By turning the knob without first pressing the lever, you discover that pressing the lever is not necessary to release the marble, thereby finding a more efficient route to your goal.

Abstracting from the details of this example, the core components of an reinforcement learning problem are an environment, an agent acting in that environment, potentially changing the state of and receiving feedback from the environment while working toward a goal. Pretty clearly, if the agent is to reliably do well in its pursuit of its goal, it must also be capable of learning from its experiences. The reinforcement learning *problem* is to do well in an environment, and *solutions* to this problem come in the form of algorithms for (efficiently) learning from experience within an environment (cf. [CITE: Sutton and Barto, 2]). In the remainder of this subsection, I will introduce some formalism for characterizing the problem and discuss some solutions to it.

Modern computational reinforcement learning is built upon the notion of a *Markov Decision Process* (MDP). An MDP is given by specifying a set \mathcal{S} of states that the

environment can be in, a set \mathcal{A} of actions that the agent can undertake, a set of rewards \mathcal{R} , and a probabilistic structure dictating how rewards and next states depend on actions and current states. A *trajectory* is a discrete (temporal) sequence of states, actions, and subsequent rewards. Moreover, we assume that \mathcal{A} and \mathcal{R} are at most countably infinite; in almost all cases, they are finite. We do not, however, assume that \mathcal{X} is finite or even countable. Some of the most difficult and interesting problems in reinforcement learning arise in the context of vast state spaces.

In general, an agent's actions depends (perhaps probabilistically) on the current state of the environment, and the subsequent reward and environmental state depend (again probabilistically) on the agent's action. Thus, state, action, and reward at a given time step are represented by random variables. We represent a trajectory as follows:

$$S_0, A_0, R_1, S_1, A_1, R_2, \dots$$

That is, S_0 denotes the initial state of the environment, A_0 denotes the first action taken by the agent, R_1 denotes the reward received as a result of the agent's first action, S_1 denotes the resulting environmental state, and so on.

The environment also specifies the conditional probabilities

$$P(S_{t+1}, R_{t+1} | S_t, A_t)$$

which is the probability distribution over next states and rewards given the agent's action in the current state. Crucially, the eponymous Markov property ensures that the effects of an action depend only on the current state of the environment. That is,

$$P(S_{t+1}, R_{t+1} | S_t, A_t, S_{t-1}, A_{t-1}, \dots) = P(S_{t+1}, R_{t+1} | S_t, A_t)$$

The Markov assumption enforces a kind of locality for the probabilistic (and hence causal) influence of states and actions upon one another. Markov processes are thus memoryless.

To illustrate the foregoing, consider the classic gridworld environment [CITE: Sutton].

[insert gridworld pic here]

The states of the environment correspond to the cells on a grid. In each state, the available actions correspond to the four cardinal directions. The effect of each action is to move to the adjacent cell in the chosen direction. The agent receives a large reward for reaching the goal state, and a small penalty for all other transitions. These consequences of the agent's actions are captured in the transition probabilities, which in this case are deterministic (i.e. if the agent chooses to go up, they'll end up in the northern adjacent cell and get negative reward—unless the cell contains the goal—with probability 1). In this example, there is a clear goal state: the only state that rewards the agent for reaching it. In other cases, rewards may be more dispersed or continuing [CITE: Abel et al. 2023].

The foregoing concludes the formal definition of an MDP. The MDP characterizes the decision problem which the agent must solve. Before discussing solutions to this problem, it is worth reflecting on the scope and nature of this modeling choice. As Sutton and Barto write in their reference textbook,

The MDP framework is abstract and flexible and can be applied to many different problems in many different ways. For example, the time steps need not refer to fixed intervals of real time; they can refer to arbitrary successive stages of decision making and acting. The actions can be low-level controls, such as the voltages applied to the motors of a robot arm, or high-level decisions, such as whether or not to have lunch or to go to graduate school. Similarly, the states can take a wide variety of forms. They can be completely determined by low-level sensations, such as direct sensor readings, or they can be more high-level and abstract, such as symbolic descriptions of objects in a room.

[CITE: Sutton and Barto, 50]

[say something about the range of application of RL, the reward hypothesis, and the markov assumption.]

We now introduce the building blocks of the agent's solution to the sequential decision problem posed by an MDP. An agent's actions in an environment are guided by a policy π . A policy is a function from states and actions to probabilities: $\pi(s, a) = p$ if the probability of choosing action a in state s is p .

INSERT GRIDWORLD WITH POLICY

Figure 2 displays an example of a (deterministic) policy for the gridworld (arrows determine which action the agent will take in each cell). Other policies are obviously possible.

How should policies be evaluated? Policies are evaluated with respect to the agent's goals. The agent's goal is to maximize its *return*. The return is a technical notion, but it is intuitively related to utility. It is defined as the agent's cumulative discounted reward. Consider an agent interacting with an MDP and generating a sequence of rewards

$$R_1, R_2, \dots$$

The cumulative reward is simply the sum of these individual rewards:

$$\sum_{t=0}^{\infty} R_{t+1}$$

For various reasons, it is advisable to discount later rewards.¹ Thus, we consider the return:

$$G := \sum_{t=0}^{\infty} \gamma^t R_{t+1}$$

¹There are mathematical reasons to discount: doing so ensures that the cumulative discounted reward is finite, and hence mathematically tractable. There are also ecological considerations: discounting by a fixed factor γ is equivalent to terminating the learning episode with probability $1 - \gamma$ at each step. Since the possibility of termination is always present in natural environments (a hawk might swoop down at any moment, or the environment might shift, so as to introduce a new learning situation) it is reasonable to build it into the framework. Note that the undiscounted case (which can be tractable in certain cases) is a special case of the discounted case with $\gamma = 1$.

Now, the actual return depends on the actual sequence of rewards which the agent experiences. Different choices of actions will in general lead to different sequences of reward. The same action taken in the same state may result in a different next state, due to the stochastic nature of the environment. And even when the same action in the same state leads to the same next state, the rewards might be different. Thus, the actual return depends probabilistically on both the environmental dynamics and on the agent's policy (as we would expect). To account for this dependence, we consider a policy's *expected return*.²

$$\mathbb{E}_\pi[G] = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} | \pi \right]$$

The agent's goal is to do as well as possible for itself in its environment. Given the environment's ineliminable stochasticity, we formalize the agent's goal as that of finding a policy with maximal expected return. And in general, we consider policies with higher expected returns to be better than ones with lower expected return.

A crucial tool in evaluating policies are *value functions*. There are two sorts of value function: state value functions and action value functions. In both cases, the value of an action or state is closely tied to return: it is the return that can be expected, in the long-run, when starting in that state or taking that action. Note that since the return depends on the agent's policy, so do value functions. We denote the value of a state s under policy π by $v_\pi(s)$ and the value of taking action a in state s by $q_\pi(s, a)$. Note that the value of an action is *not* the reward that results from taking that action (in a given state). As an estimate of return, value is sensitive to long-run consequences, whereas rewards are inherently local, attaching to particular actions in particular states independently of their connection to other states.

Intuitively, if the agent's goal is to maximize expected return, then it should seek to perform high-value actions (alternatively, it should seek high-value states, but we will focus on action for simplicity). There are two subtleties with this picture, and they drive the development of the most fundamental reinforcement learning algorithms. The first is that action values are, at least initially, not known to the agent.³ Indeed, action values are expectations over (potentially) infinite sequences of rewards, the calculation of which is anything but trivial and requires information (such as transition probabilities between states) that may in any case not be available to the agent. Thus, if the agent is to select actions based on their value, it must have a way of estimating these values on the basis of its experience in its environment: it needs to learn action values from experience.

Second, action values depend in general on the agent's policy. If the agent modifies its policy, for example after learning that a given action has a high value, this change could in principle modify the value of that very action, rendering the updated policy worse than the original one. In fact, this cannot happen: improving a policy "locally," by swapping a lower-value action for a higher-value one, can only improve the one's

²A policy and an MDP define a probability distribution over returns. Discounting is needed to ensure that this distribution has an expectation.

³I am using "know" in a maximally loose sense here, for ease of exposition.

expected return. This result is known as the *policy improvement theorem*.⁴

Although local improvements to the policy cannot make it worse, such improvements usually change the value of other actions and states. For example, consider the following toy environment:

INSERT SIMPLE DECISION TREE

Clearly, the optimal policy (in a sense to be made precise below) is to first go left in state s_1 , then go right in state s_2 . Suppose that π_1 goes left in state s_2 . The value of going left in state s_1 , conditional on following π_1 , is low, since doing so will be followed by another left turn, and hence a low reward. Suppose now that the agent learns that going right in state s_2 in fact has greater value, and so updates to a policy π_2 , which is like π_1 except that it goes right in state s_2 . By the policy improvement theorem, π_2 is better than π_1 (as can be easily checked manually). Notice also that the value of going left at s_1 is much greater under π_2 than under π_1 . Importantly, however, we have so far left it open whether the agent is sensitive to this fact. Learning the value of going right in s_2 under π_1 (i.e. $q_{\pi_1}(s_2, \text{right})$) is quite distinct from learning the value of going left at s_1 under π_2 (i.e. $q_{\pi_2}(s_1, \text{left})$). And until this new value is learned, the agent's estimate of action values in state s_1 will be mistaken, relative to the new policy π_2 .

Thus, although the first remark above points to the need for learning action values (or related quantities, such as state values), the second shows that such learned values may be invalidated as soon as they are used—as they should be—to improve one's policy. This tension suggests that whatever method we use to learn action values should not be too costly, as we might need to re-learn these values as soon as our estimates are used to improve our policy.

State and action values obey a consistency condition called the *Bellman recurrence*:

$$v_{\pi}(s) = \sum_a \pi(s, a) \sum_{s', r} p(s', r | s, a) [r + \gamma v_{\pi}(s')]$$

This states that the value of state s under policy π is identical to the expected immediate reward associated with being in state s and following π , plus the value of the resulting state s' . Intuitively, present value is one-step reward plus subsequent value. A similar equation holds for action values, according to which the value of a given action in a given state is the expected reward of taking that action plus the value of the expected action in the resulting state.

The Bellman equation is of fundamental importance to reinforcement learning. It enables an iterative approach to learning values and policies that elegantly bypasses the concerns raised in the previous paragraphs. In the next few paragraphs, we present a widely-known and illustrative algorithm that leverages the Bellman recurrence to learn an optimal policy.

TODO:

1. algorithms

(a) explore exploit

⁴A rigorous statement and proof sketch can be found in [CITE: Sutton and Barto: 78].

- (b) policy iteration
- (c) Q learning

3.2 Reinforcement Learning in Cognitive Science

Reinforcement learning has a long and distinguished history in the cognitive sciences. In the early 1980s, computer scientists and cognitive scientists observed that the then-dominant Rescorla-Wagner model of classical conditioning [CITE: Rescorla and Wagner] could be subsumed under the method of temporal differences [CITE: Sutton and Barto 1981].

The development of the temporal difference model of classical conditioning throughout the eighties met with great empirical success. The model elegantly unified a variety of puzzling phenomena related to learning. For example, while the Rescorla-Wagner model could account for blocking, it did not have the resources to capture higher-order conditioning.⁵ By contrast, both blocking and higher-order conditioning are easily seen to be consequences of the same prediction error mechanism at the heart of the temporal difference model.

In addition, the temporal difference model allowed for much greater temporal resolution than existing models. Indeed, the basic unit of temporal organization in the Rescorla-Wagner model is the *trial*: during a trial, the animal may be presented with any number of stimuli, separated by various intervals; the model sees learning as updating parameters from one trial to the next. As such, it is blind to the finer temporal structure of trials, and cannot model within-trial learning. The temporal difference model, by contrast, affords experimenters a fine-grained view into the temporal structure of a single trial. As a result, a variety of factors that could not even be expressed in the Rescorla-Wagner model—such as the temporal distance between stimuli (the *interstimulus interval*), temporal overlap and adjacency of stimuli, and various subtle

⁵Blocking occurs when previously learned associations prevent the formation of new associations. For example, suppose an animal has been trained to associate a tone with the delivery of food. If the tone is then combined with another stimulus (such as a light) while the rest of the learning setup remains unchanged, the animal will fail to learn an association between the light and the food. The prior tone-food association blocks learning a light-food association. One of the great successes of the Rescorla-Wagner model was its elegant explanation of blocking. Roughly, the model posits that learning occurs only when something surprising happens. Since in blocking cases, the reward is fully predicted by the tone, its delivery is not surprising. There is no surprise “left over” to fuel learning of a light-food association, and so the model correctly predicts that learning will not occur.

Higher-order conditioning occurs when an animal forms associations between two stimuli that have not been presented together. For example, suppose that an animal is taught a tone-food association and is then repeatedly exposed to a light-tone association (without food delivery). The animal exhibits higher-order conditioning if it learns a light-food association. Note that the animal has never experienced any (immediate or delayed) connection between light and food. However, it has learned that the light is predictive of a tone, which is in turn predictive of food. (There are subtleties of experimental design that necessitate great care in setting up higher-order conditioning experiments: since a correlation between light and food would undermine the logic of the experiment, food cannot be presented during the light-tone trials. But if tones are presented without being followed by food, the tone-food association undergoes extinction, and becomes unable to support higher-order conditioning. Thus, the light-tone trials must be interspersed with tone-food trials. But this interleaving now risks introducing some degree of correlation between the light and food, again jeopardizing any inference to true higher-order conditioning. Fortunately, statistical methods can be used to confirm that animals indeed undergo higher-order conditioning.)

manifestations of blocking—were successfully modeled [CITE: Kehoe, Schreurs, and Graham 1987, Sutton 1984, 1988, Sutton and Barto 1987, 1990]. In addition, the increased (temporal and conceptual) resolution of the temporal difference model allowed researchers to frame several novel questions (a mark of good science, according to [CITE: Laudan/Lakatos?]): how is the presence or absence of a stimulus across a period of time registered by the animal? How are the model parameters (such as learning and decay rates) set? And, perhaps most importantly, how is the temporal difference error at the heart of the model computed?

This last question was the focus of a burst of activity in the nineties, when researchers observed that midbrain dopaminergic neural activity precisely matched the reward-prediction error associated with a given task [CITE: Montague et al. 1993, Montague et al. 1995, Montague et al. 1994, 1996, Niv 2009]. The details of this correspondence are not relevant for our purposes. It will suffice to note that many core components of temporal difference algorithms were seen to be implemented in the brain: state- and action-value estimates, prediction errors, actor and critic structures, and so on. Contemporary research has even found neural support for more advanced forms of reinforcement learning, such as hierarchical reinforcement learning (HRL). HRL enriches the basic MDP setup with *options*, which are temporally extended action sequences that the agent can select. Implementing an HRL model requires tracking several prediction errors at once, on distinct time scales. Empirical support for these relatively sophisticated error signals has been found [CITE: Botvinick et al. 2009, Botvinick 2012, Diuk et al. 2013].

[contemporary questions in neuroscientific RL research: locating the neural substrates of various components of RL algorithms (actor and critic, value functions, model-based vs model-free adjudication, information gain)]

Reinforcement learning outgrew its behaviorist roots through the development of *model-based* reinforcement learners. As Sutton and Barto put it, a model

is something that mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave. For example, given a state and action, the model might predict the resultant next state and next reward. Models are used for *planning*, by which we mean any way of deciding on a course of action by considering possible future situations before they are actually experienced.

[CITE: Sutton and Barto: 7]

Minimally, a model must track the environmental dynamics, and be usable in deciding what to do. Two important classes of models are *distribution models* and *sampling models*. A distribution model takes the form of an algorithm, which when given a state s , an action a , a next state s' , and a reward r , outputs a probability p of transitioning to s' and receiving reward r upon taking action a in s . That is, a distribution model is a way of computing some $p(s', r | s, a)$ for all relevant values of s', r, s , and a . By contrast a sampling model is an algorithm that on input s and a outputs a next state s' and reward r with some probability p . Intuitively, a sampling model encodes a distribution, but only implicitly: it does not make the transition probabilities available for further computation [CITE: Rescorla, "Neural Implementation of Bayesian Inference"]. The

distribution encoded by a sampling model is the obvious one: it assigns probability p to the transition $s, a \rightarrow s', r$. Either kind of model can be used to plan, though they necessitate different algorithms.

For a simple example, consider [Example of planning with a model].

Model-based reinforcement learning supports a kind of behavioral flexibility unavailable to model-free learners. To see this, consider the following environment.

[INSERT 3-WAY PATH HERE]

This environment consists of three paths to the goal, with a reward of 1 for reaching the goal and a reward of -1 for each other time step. Thus, the agent does best by taking the shortest path (namely A) to the goal. Now suppose that some obstruction is introduced at the end of path A, blocking the path to the goal. Note that this obstruction also blocks the intermediate path B. Suppose a model-based learner encounters this obstruction while traveling down path A and updates its environmental model to reflect the change. If the agent is placed in the starting state again, it will directly go for the longest path C. This is because, when deciding which path to take, the learner can look ahead at the consequences of taking each path. Doing so reveals that both path A and B are now very inefficient ways to reach the goal, and that path C is now the most direct route to the goal.

By contrast, a model-free learner, having encountered the obstruction at the end of path A, would take path B if placed in the starting state. This is because the model-free learner can only update its estimate of action values by taking these actions and observing their consequences. Since it did not yet encounter the obstacle after choosing path B, the learner cannot update its estimate of the value of that path. Thus, path B will now appear to be better than path A and path C. Only once path B has been taken and found to be obstructed will the agent learn the correct value for B, and ultimately choose C.

This example illustrates that model-based learners can respond flexibly to changes in the causal dynamics of their environment. Similar examples show that the same is true of changes in the rewards associated with outcomes. Model-based learning is freed from the need to experience transitions in order to learn their values. The model supports a kind of generalization, whereby information acquired through one experiential trajectory bears on decisions independent of that trajectory.

Models have been associated with cognitive maps: representations of the agent's environment whose format mirrors the spatial structure of the environment [CITE: Tolman 1948, Daw et al. 2005, Rescorla 2009, Chrisippus]. We shall examine in detail the representational credentials of model-based reinforcement learning below. For now, we detail some uses of the distinction between model-based and model-free learning in the cognitive literature.

The model-based/model-free distinction is used to explain the distinction between *habitual* and *goal-directed* action [CITE: Dayan, Niv, etc.]. The habitual/goal-directed distinction is itself operationalized using the notion of *outcome devaluation sensitivity*. A type of behavior is sensitive to outcome devaluation if information about the value of the consequences of a choice influences that choice. For example, consider the following experiment. [CITE: who did this again?] taught rats that lever presses lead to

food through standard instrumental conditioning protocol. They then fed food to the rats freely, in an environment devoid of levers. At the same time, the test subjects were injected with a nausea-inducing drug. Upon returning to their original lever environment with the levers disconnected to reward, the rats pressed the lever less often (you would expect this anyway, since the lever is no longer connected to anything valuable, but as it turns out, the poisoned rats' rate of lever-pressing decreased faster than that of non-poisoned rats).

In this experiment (and many others like it [CITE: Drummond and Niv, Dolan and Dayan 2013]), the rats' behavior exhibits sensitivity to outcome devaluation: if an outcome (food) is devalued (by associating it with nausea), the rats are less likely to choose actions leading to this outcome. But notice that at no point do the rats experience any association between lever pressing and nausea. Lever presses are only ever followed by either pleasant experiences (food in the first phase) or neutral experiences (nothing in the third phase). For their nauseated states to influence their lever-pressing, the rats would need to associate lever-pressing with the receipt of food, and the consumption of food with nausea. That is, they would need a rudimentary environment model—a mental structure that tracks the transition and reward structure of their environment—and a way to use this model to bring future outcomes to bear on their current decision to press the lever. [how to justify the "goal-directed" terminology? Honestly I'm not sure there's a clear sense in which model-based behavior is goal-directed while model-free is not; rather, they're both goal-directed, but model-based learning is immediately responsive to goal changes, while model-free learning needs to visit the environment to implement the necessary changes. Need to think more about whether the cog sci terminology is just misleading, or reflects a deeper contrast.]

4 RL and know-how

Reinforcement learning provides powerful tools for thinking about many of the questions at the heart of the philosophy of skill and know-how. In this section, we argue that a closer look at cases of model-based learning undermine the intellectualist thesis.

Our argument is as follows:

- (i) Some animal behavior constitutes know-how or skilled action.
- (ii) This behavior can be explained in terms of (model-based) reinforcement learning.
- (iii) These explanations do not warrant the ascription of propositional representations.
- (iv) Therefore, know-how does not require, and hence does not consist in, propositional knowledge.
- (v) Moreover, such representations as are needed to explain this behavior do not represent ways of performing an action.
- (vi) Thus, know-how does not require, much less consist in, representing ways of performing a task, whether propositional or not.

Note that (iv) contradicts INTELLECTUALISM while (vi) contradicts WAY-REPRESENTATIONALISM.

Before assessing each premise, a few points of clarification. First, note that the first and second premises should be read as follows: for some class of animal behavior, that behavior constitutes know-how and can be explained through reinforcement learning. They should not be read as implying that any animal behavior that constitutes know-how can be explained through reinforcement learning (this would be an untenably strong claim).

Second, on the structure of the argument. The significant logical steps occur in the transition from (i)–(iii) to (iv) and from (v) to (vi). These steps are not logically valid: it is logically possible that propositional representations not be required to explain a type of behavior, which nonetheless requires the possession of propositional representations. However, I maintain that explanatory power is the principal reason to posit structured representations [CITE: Fodor, Burge]. If we find that an animal’s behavior can be explained without imputing it propositional representations, we should not attribute it propositional capacities (on the basis of that behavior). Doing so would fly in the face of the fact that representations are explanatory posits of the cognitive sciences [CITE: Fodor 1986]. As such, they must pay their ontological keep in explanatory coin.

Third, this argument is meant to be consistent with the animals in question having propositional capacities. The point at issue is not whether, say, rats are capable of propositional thought. Rather, the question concerns whether their knowing how to navigate a maze depends on their having propositional knowledge. My argument is that since their navigational abilities can be explained without recourse to propositional structures, their possession of these capacities does not depend on propositional knowledge. But this is compatible with the rats nonetheless possessing propositional knowledge.

Let us now defend our premises. Instead of defending each of (i)–(iii) and (v) individually, I will present some case studies in some detail, and then argue that they exemplify the premises.

4.1 Rat navigation

TODO