REINFORCEMENT LEARNING: COMPUTATIONAL MODELING OF LEARNING AND DECISION-MAKING

ANALYTICAL CONNECTIONISM 2023
LECTURES BY DR. MARIA ECKSTEIN

Zach Cohen*
Department of Neurobiology
Kempner Institute
Harvard University
Cambridge MA, 02138
zcohen1@g.harvard.edu

Akshay K. Jagadish*
Princeton AI Lab
Princeton University
Princeton, NJ, 08540
akshaykjagadish@gmail.com

Eghbal A. Hosseini*
Deparment of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA, 02139
ehoseini@mit.edu

Maria K. Eckstein
Google DeepMind
London, UK
mariaeckstein@deepmind.com

ABSTRACT

Reinforcement learning (RL), as a computational modeling framework, is a formal approach to understanding and building agents, natural or artificial, that learn to make decisions based on rewards they receive from the environment. In this Lecture Notes, we begin by exploring how RL has historically been used in psychology and neuroscience to investigate reward-driven learning, before introducing it more formally from a machine learning perspective. We then demonstrate its utility in building cognitive models that explain the processes and mechanisms underlying human learning and decision-making at both the behavioral and neural levels. Finally, we discuss recent work that brings together the theory-driven approach taken by RL and the data-driven approach taken by artificial neural networks to build more predictive, yet interpretable, models of human behavior. Together, this work highlights the value of RL as a computational modeling framework for cognitive neuroscience.

Keywords Reinforcement Learning · Cognitive Modeling · Learning · Decision-making

1 Introduction

Reinforcement learning (RL) is the process by which an agent learns to maximize its cumulative reward in a given environment. Rapid theoretical development of algorithms for solving RL problems Sutton and Barto [1998] has offered increasingly rich and complex frameworks and models of how the brain may solve a range of naturalistic tasks. RL's success as a framework in cognitive neuroscience is due in part to its broad-reaching predictions at various levels of study in the field. Indeed, RL makes predictions about the computational goals of a system (i.e., that modeled organisms seek to maximize cumulative reward), about the algorithms that are best suited for doing so under various conditions (e.g., algorithms that deploy world knowledge for deliberation versus those that simply bootstrap from experience), and about what neural activity may subserve the implementation of these algorithms. These predictions have been successful in explaining many aspects of animal and human behavior (Daw et al. [2011], Sutton and Barto [1998], Otto et al. [2015], and evidence suggests that the substrates necessary to solve RL tasks may be directly implemented by specialized neurons in the brain [Padoa-Schioppa and Assad, 2006, McClure et al., 2003, Oyama et al., 2010, Glimcher,

*Equal contribution. Author order chosen randomly.

2011, Schultz et al., 1997, Luo and Huang, 2016, Wise, 2004, Watabe-Uchida et al., 2017]. Beyond offering useful predictions for cognitive neuroscience, RL theory is a fast-evolving branch of machine learning that is particularly well-developed in tabular settings. As such, the theoretical apparatus available to modelers is highly flexible and well understood, lending both expressive power and interpretability to RL models used to explain cognitive function.

The following explores RL as a cognitive model from a variety of perspectives, following the structure of Dr. Maria Eckstein's lecture in the Analytical Connectionism Summer School, held at the University College London in 2023. We briefly survey the fundamentals of RL from a machine learning perspective. We also survey RL's historical use as a model from psychological and neuroscience perspectives. Finally, we summarize recent developments in RL's use as a cognitive model, which combine theory- and data-driven approaches to bridge insights in psychological and neuroscientific domains to comprehensively shed light on the algorithms used by the brain and how they may be implemented.

2 Reinforcement learning from psychological perspective

The question of how rewards shape animal behavior has intrigued psychologists for centuries [Thorndike, 1998, Jones and Skinner, 1939, Pavlov, 2010, Rescorla and Wagner, 1972]. Animals have been shown to learn complex behaviors through rewards alone, commonly referred to as reinforcement learning. For instance, a dog can be trained to perform acts of different levels of difficulty, from simple acts, like shaking your hand, to complex sequence of actions, such as running through an entire obstacle course performing different stunts. Remarkably, all of these feats can be directly attributed to reward-based learning.

Psychologists have posited many theories to explain how animals learn different behaviors through reinforcement. One such theory in psychology is classical conditioning [Pavlov, 2010]. Although its discovery was accidental, it is considered a major breakthrough in psychology, culminating in a Nobel prize in physiology and medicine in 1904. Classical conditioning describes how a previously neutral stimulus can be conditioned to elicit an automated, unconditioned response by repeatedly presenting stimuli together. To illustrate it, let us assume that a dog is presented with a piece of meat. Before implementing any conditioning procedure, the dog immediately salivates (Behavior in Figure 1; top) when food is presented. This is the dog's reflexive response built in by evolution. The piece of meat shown to the dog is the unconditioned stimulus (US) and the salivation evoked in the dog prior to conditioning is called the unconditioned response (UR). The goal of conditioning is to induce an association between a neutral stimulus (NS), such as the sound of a bell, that does not evoke an UR on its own, and the US, the piece of meat. Pavlov demonstrated that this can be achieved simply by repeatedly presenting the NS along with the US. There are constraints, such as timing between stimulus, intensity relationship, etc. that are necessary to form the association, but this is beyond the scope of this paper. However, assuming that they adhere to these constraints, the dog learns to associate the sound of the bell (Situation in Figure 1; top) with the presentation of meat (Consequence in Figure 1; top). After conditioning, the salivary response evoked by the NS, which is renamed the conditioned stimulus (CS), is called the conditioned response (CR).

Classical conditioning, despite its usefulness, is still a descriptive theory of associative learning. It does not explain the processes and mechanisms underlying associative learning. Rescorla and Wagner tried to address this by formalizing classical conditioning with the help of a computational model, known as the Rescorla-Wagner (RW) model [Rescorla and Wagner, 1972]. This model attempts to explain how the association between CS and CR is learned over the course of the experiment. Specifically, its goal is to learn to accurately predict the US given the CS.

More concretely, the RW model assumes that each CS has a specific association strength, called the value (V_{CS}), with the US (or reward) and treats the US, prior to conditioning, as the reward (r). V_{CS} of each CS is updated after each trial based on the trial-specific reward prediction error (RPE), which is defined below:

$$\mathrm{RPE} = \mathrm{r} - \Sigma[V_{\mathrm{CS}}] \tag{1}$$

where Σ denotes sum over all CS. As there can be many CS, the sum plays a crucial role as it indicates that the reward (or the US) is predicted jointly by all CS.

The computed RPE helps trigger learning by adjusting the value of the CS until it is good at predicting the reward in a given trial. The learning rule used for the update is as follows:

$$V_{\mathrm{CS}}^{new} \leftarrow V_{\mathrm{CS}} + \alpha_{\mathrm{CS}} * \beta_{\mathrm{US}} * \mathrm{RPE} \tag{2}$$

where the rate of learning depends on the salience of the CS, $\alpha_{\mathrm{CS}} \in (0, 1)$, and association strength of US, $\beta_{\mathrm{US}} \in (0, 1)$. This learning rule ensures that learning occurs gradually, adjusting V_{CS} according to RPE. When the RPE is positive, the strength of the association increases. When the RPE is negative, the strength of the association decreases. If the actual reward matches the expected value, no learning occurs because the animal's expectations are already accurate.

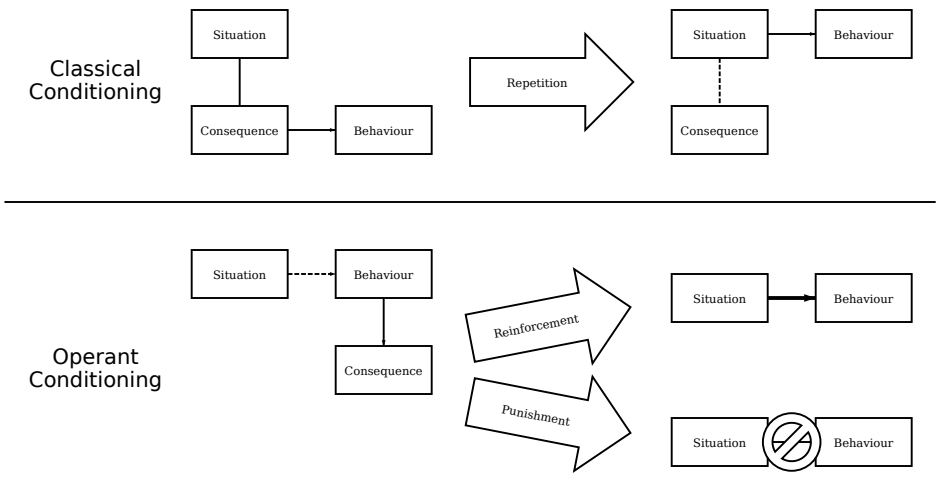

Figure 1: Schematic for reinforcement learning theories from psychology. Classical conditioning (top), where a Situation and a Consequence are associated through Repetition to eventually elicit a Behaviour. Operant conditioning (bottom), where a Behaviour performed in a specific Situation leads to a Consequence that acts as either Reinforcement (strengthening the link) or Punishment (breaking the link). Image by Perey, licensed under CC BY-SA 4.0 via Wikimedia Commons.

To demonstrate how the RW model works in practice, let us carry on with Pavlov's dog example. Before conditioning (trial = 0), the value of the bell, V_{bell}, is assumed to be zero, since no rewarding stimulus (r = 0) has been paired with the bell. During the conditioning phase, the US is presented along with the sound of the bell in each trial. Let us assume that the reward takes a value of 1 upon presentation of the US and that the learning rate $\alpha = \alpha_{\text{CS}} * \beta_{\text{US}} = 0.5$. After the first trial, RPE can be computed using Eq. 1, which is 1. Then we can use the computed RPE to update V_{bell} following Eq. 2. This results in a new estimate for V_{bell} (0.5). Assuming that everything else remains the same, the steps are repeated after each trial. For instance, after trial 5, V_{bell} = 0.97, which closely approximates the reward value for the US (see Table 1). After the conditioning phase is completed, V_{bell} can be assumed to be equal to the reward value, that is, 1. This high V_{bell} for the bell explains why after conditioning the dog immediately salivates after the bell ring.

In addition, the RW model can explain other phenomena connected with classical conditioning, such as blocking [Kamin, 1967] and second-order conditioning [Pavlov, 2010]. In the case of blocking, it has been shown that when a new neutral stimulus is presented together with the initial CS, the new neural stimulus fails to elicit a conditioned response even after it is repeatedly coupled with a conditioned stimulus. This is because, after conditioning, the CS fully predicts the US and the new stimulus has nothing to predict. This phenomenon is known as blocking, as the first stimulus essentially blocks the second stimulus from being associated with the reward. To demonstrate this through the RW formulation, let us assume that V_{bell} is 1 (equal to reward) after the conditioning phase, and the value for the second stimulus, let us say light, V_{light} is zero before the start of the conditioning phase. When a second stimulus is co-presented with the bell during the conditioning phase, its value does not update at all, as the RPE is 0. This is because the expected value for all CS, ΣV_{CS}, is already 1 (as in Eq. 1) which matches the reward magnitude. As a result, $V_{\text{light}}^{\text{new}}$ remains 0 throughout. These computations lend support to the empirical observation that the second stimulus is blocked by an already conditioned stimulus.

Trial	V_{bell}	r	RPE	V_{bell}^{new}
0	0	0	-	-
1	0	1	1	0.50
2	0.50	1	0.50	0.75
3	0.75	1	0.25	0.87
4	0.87	1	0.12	0.94
5	0.94	1	0.062	0.97
∞	1	1	0	1

Table 1: Rescorla-Wagner updates across five trials with $\alpha = 0.5$ and reward = 1.0.

Furthermore, a variant of the RW model has been used to explain operant conditioning. In this case, an animal has to perform a physical action, like pressing a lever, to get a reward when the CS is presented as opposed to only passively observing the CS-US pairs. We need to slightly adapt the RW formulation from before to explain how the animal learns to associate performing an action in a given state with reward. Specifically, the equations of RPE and value update undergo a slight change to incorporate action-reward contingency:

$$\begin{aligned} \text{RPE} &= \text{r} - V_{a|s} \\ V_{a|s}^{new} &\leftarrow V_{a|s} + \alpha * \text{RPE} \end{aligned} \tag{3}$$

where a corresponds to action, like pressing a lever, and s is state under which the action is performed, like presentation of the light stimulus.

To illustrate this, let us consider an experiment in which a rat needs to press a lever (Behaviour in Figure 1; bottom) upon presentation of a light stimulus (Situation) to get a reward. (Consequence) The goal is to model how the rat learns to associate pressing the lever on the presentation of light with getting a reward. Let $V_{\text{press|lever}}$ be the value of pressing given the presence of the lever, the learning rate, α, be 0.5, and the reward r be 1. Before any learning, $V_{\text{press|lever}}$ takes a value of 0. In the first trial in which the rat presses the lever to get a reward, the RPE will be 1. This is then used to compute the $V_{\text{press|lever}}^{new}$, which is equal to 0.5. The two steps are repeated until the task is complete, assuming that everything else remains constant. After trial 3, $V_{\text{press|lever}}$ = 0.875, which closely approximates the reward value for pressing the lever (see Table 2 for details). After the conditioning phase is completed, $V_{\text{press|lever}}$ can be assumed to be equal to the reward value, that is, 1. Higher learned value estimates have been found to be linked not only with the likelihood of pressing the lever, but also with the rate at which the lever is pressed. Therefore, this term serves as an effective proxy for measuring the strength of learned associations.

Furthermore, the estimate of the learned value, $V_{\text{press|lever}}$, at the end of operant conditioning has also been found to be useful in explaining the habitual responses made by both animals and humans [Dickinson, 1985, Wood and Neal, 2007]. When the rat is presented with the lever after the conditioning phase, it was found to press the lever even when it was not hungry. This form of response constitutes habitual responses, which is different from goal-directed responses, where the rat has an internal model and presses the lever only when hungry.

Trial	$V_{\text{press\|lever}}$	r	RPE	$V_{\text{press\|lever}}^{new}$
0	0	0	-	-
1	0	1	1	0.5
2	0.5	1	0.5	0.75
3	0.75	1	0.25	0.875
∞	1	1	0	1

Table 2: Operant conditioning updates across three trials with $\alpha = 0.5$ and reward = 1.0.

In this section, we have explored how two rudimentary forms of reinforcement learning, namely, classical and operant conditioning, can shape behavior. We have illustrated with examples how rewards lead an animal to learn different types of response, either passively or actively. In the next section, we outline how reinforcement learning might be realized in the brain.

3 Reinforcement learning from neuroscience perspective

A substantial body of work in the neural substrate of reinforcement learning has centered on the study of the neurotransmitter dopamine. Traditionally, research has highlighted three major dopamine pathways: mesolimbic, mesocortical, and nigrostriatal. Each pathway originates from specific subcortical regions (e.g., the ventral tegmental area (VTA) or substantia nigra) and projects to cortical or subcortical targets [Luo and Huang, 2016].

Dopamine has been implicated in a wide range of functions, including sensorimotor learning, memory, attention, emotion, movement, and motivation (often termed "reward" or "pleasure seeking"; Bromberg-Martin et al., 2010, Wise, 2004). In the context of reinforcement learning, the mesolimbic pathway (from the VTA to the nucleus accumbens and the cortical targets) has been a central focus in both animal and human studies, as it is critically involved in reward-related learning and prediction errors [Watabe-Uchida et al., 2017].

Our contemporary understanding of dopamine in reward-related learning has originated from neurophysiological studies of the dopamine neuron firing pattern in response to rewards and predictive cues. The seminal study by Wolfram Schultz and colleagues provided a foundational account of dopamine activity in VTA [Schultz et al., 1997]. In their

experiment, monkeys were trained in a Pavlovian conditioning task, in which a CS was followed by a reward r after a random inter-trial interval (ITI). Schultz and colleagues made three critical observations about dopamine neuron activity. First, before learning, when no conditioned stimulus was present, dopamine neurons showed a sustained (tonic) level of activity, but the delivery of reward produced a transient (phasic) increase in dopamine firing (see Fig. 2). Second, after the monkeys learned the association between the CS and reward, the onset of the CS itself elicited the phasic increase in dopamine activity, and the reward no longer caused any additional change. Finally, when an expected reward was omitted following the CS, the dopamine neurons exhibited a phasic suppression (decrease) in their firing rate.

What exactly are dopamine neurons encoding about the task? Their activity suggests that they do more than simply report rewards, or movements. After learning, when a reward consistently follows a CS, dopamine neurons show no additional change upon reward delivery, yet they exhibit a depressed response when the expected reward is omitted. Instead the dopamine neurons activity appears to encode expectations over external events, and their violations, consistent with the reinforcement learning (RL) framework[Sutton and Barto, 1981, 1998]. In this framework, the agent learns a value function $Q(S)$, representing the expected discounted sum of future rewards from state S. Schultz et al., 1997 employed temporal-difference (TD) learning algorithm to explain dopamine activity, positing that these neurons encode a *reward prediction error (RPE)*.

$$\delta = r + \gamma Q(S_{t+1}) - Q(S_t)$$

where γ is a discount factor for future rewards. Before learning, the value of the inter-trial interval is $Q(S_{\text{ITI}}) = 0$, so an unexpected reward elicits a large positive RPE (Fig. 2D▲). As learning proceeds, the value estimate $Q(S_{\text{CS}})$ is updated via a TD error. Once the CS acquires a high value $\text{Q}\big(S_{\text{CS}}\big)$, the onset of S_{CS} itself triggers the positive RPE (Fig. 2E▲), and the reward delivery then adds little or no additional RPE (Fig. 2E▲). Finally, if the reward is omitted when expected, the difference between the predicted and actual outcome becomes negative, resulting in a suppression of dopamine neuron activity (Fig. 2F▲).

This formulation relies on two key assumptions. First, learning is framed as predicting the expected discounted sum of future rewards given a sensory cue. Second, it assumes the Markovian property, meaning that future states and rewards depend only on the present state and not on past states or rewards.

Extensive research across species indicates that dopamine neurons—especially those projecting to the striatum—encode a reward prediction error signal in classical and operant conditioning paradigms [Glimcher, 2011, Oyama et al., 2010, McClure et al., 2003, Daw et al., 2011]. Alongside this dopaminergic input, the orbitofrontal and ventromedial prefrontal cortices represent value, and together these areas appear to implement computations akin to temporal difference learning within cortico-basal ganglia loops. This framework is further supported by human neuro-imaging studies showing increases in the BOLD signal in the ventromedial / orbitofrontal cortex and striatum when rewards or other appetitive stimuli are anticipated or received, highlighting a more general appetitive function [O'Doherty et al., 2003, Levy and Glimcher, 2012, Knutson et al., 2001, Oyama et al., 2010]. While the reward prediction error hypothesis is robust, ongoing research continues to refine our understanding of how these signals vary with context, salience, and other motivational factors.

In summary, the discovery of dopamine encoding of reward prediction errors represents a key insight into the neural basis of reinforcement learning, and a computational grounding for understanding how experience shapes behavior.

4 Reinforcement learning from the AI perspective

In the machine learning context, reinforcement learning is a general framework for describing reward-based learning. It is often considered a third pillar of machine learning paradigms, alongside supervised and unsupervised learning.

A reinforcement learning problem is composed of four central components: (1) a decision-making agent, (2) an environment imbued with various deterministic or stochastic characteristics, (3) a notion of reward extrinsic in the environment (or intrinsic to the agent), and (4) a goal for the agent, which is to utilize its experience in order to learn how to navigate the environment in a way that maximizes cumulative reward. Solving an RL problem is tantamount to achieving the goal in (4), and entails learning a policy to achieve the goal. The task of navigating a maze with treasure at the end of it, for example, is well captured by the RL framework: The environment (maze) may not be trivially navigable. In order to find a way through the environment, the agent can try out a few trajectories and learn from interactions with the environment over these attempts. If the agent learns how to navigate the maze, the agent will be rewarded (with the treasure). (Fig. 3).

In order to render RL problems tractable, the environment (and an agent's interaction with it) is assumed to be Markovian. Under the assumption of a Markovian environment, transitions from one state to another are only dependent on the current state of the agent and the action it chooses, rather than its full state and action history. Formally, the interaction between an agent and its environment is modeled as a Markov decision process (MDP). An MDP, M, is

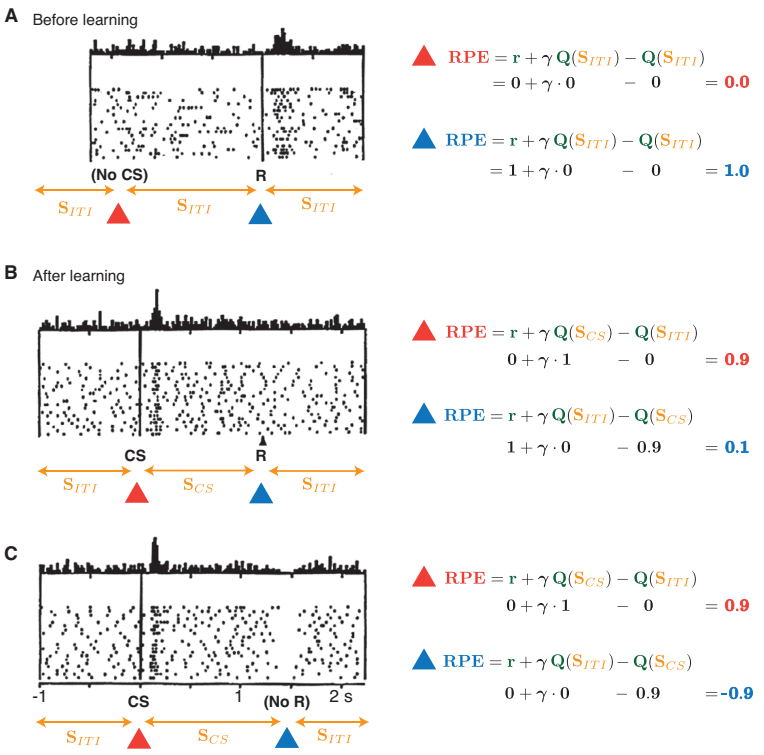

Figure 2: (A) Before learning, neuronal firing does not change in the absence of a conditioned stimulus (CS) and shows an increase in activity in response to reward (R). (B) After learning, a prominent firing response occurs at the CS onset, and (C) when reward is omitted, firing decreases at the expected reward time. (D–F) Corresponding reward prediction error (RPE) equations illustrate how RPE values shift with the introduction of the CS and the omission of reward. Modified with permission from Schultz et al., "A Neural Substrate of Prediction and Reward," *Science*, DOI: 10.1126/science.275.5306.1593 1997, AAAS.

defined as a tuple $M := (\mathcal{S}, \mathcal{A}, p, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. The function $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathbb{R} \to [0, 1]$ defines the dynamics of the MDP and assigns a probability of being in state $s \in \mathcal{S}$, taking action $a \in \mathcal{A}$ and arriving in state $s' \in \mathcal{S}$ with associated reward $r \in \mathbb{R}$. In some instances, the dynamics function p is broken down into two functions, $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ and a separate reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ which assigns a number $r \in \mathbb{R}$, the reward, to the sequence in which the agent starts in s and transitions through action a into state s' (accordingly, in this case, $M := (\mathcal{S}, \mathcal{A}, p, R, \gamma)$; additionally, sometimes the reward function is defined as $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which is basically the same as the original definition, except that it does not specify the target state). Finally, $\gamma \in (0, 1)$ is a discounting factor, which describes how much the agent discounts events that happen far into the future.

The agent's goal is to maximize its cumulative discounted reward in the future (here, starting from time t), $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$. One way an agent accomplishes this is through two interleaved processes (though other algorithmic approaches, such as policy gradient methods, can achieve this goal in other ways). One is a predictive process: The agent must learn to predict its expected return G_t *given* its current policy π, which maps states to actions, and reflects what actions an agent takes given a particular state. The other process is one of control: The agent must adapt its behavior, summarized in a new, improved policy π', which can in principle allow the agent to achieve greater reward than if it continued following the old policy π.

The predictive process involves learning a *value function*, which maps states in $s \in \mathcal{S}$ to the expected return, assuming the agent starts in s, and follows the current policy π:

$$v^{\pi}(s) = \mathbb{E}_{\pi}\left[G_t | S_t = s\right] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \tag{4}$$

The optimal value function v^* is the value function of the optimal policy:

$$v^* = \max_{\pi} v^{\pi} \tag{5}$$

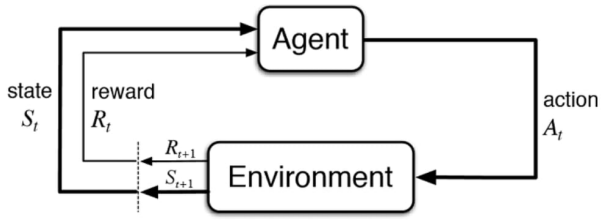

Figure 3: The basic ingredients of an RL problem (taken from Sutton and Barto [2018]). An agent interacts with the environment through actions. The environment in turn emits new observations and rewards to the agent. The agent derives a policy through this feedback loop: The agent observes how its actions furnish (or fail to furnish) reward. Over time, the agent adjusts its policy in order to maximize its cumulative reward.

In contexts where control is also of interest, it is helpful to use the *state-action value function*, which is defined for all (s, a) (where $a \in \mathcal{A}$) pairs, and is the expected return assuming the agent starts in s, takes action a and thereby follows the current policy π:

$$q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[G_t | S_t = s, A_t = a\right] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \tag{6}$$

Similarly, the optimal state-action value function q^* is the state-action value function associated with the optimal policy:

$$q^* = \max_{\pi} q^{\pi} \tag{7}$$

The Markovian assumption on the environment dynamics along with the recursive nature of the value function permits one to write down the Bellman consistency equation (here for v^{π}):

$$\begin{aligned} v^{\pi}(s) &= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] = \mathbb{E}_{\pi}\left[R_{t+1} + \sum_{k=1}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \\ &= \sum_{a} \pi(a|s) \sum_{s',r} p(s', r | s, a)\big[r + \gamma v^{\pi}(s')\big] \end{aligned} \tag{8}$$

where the value of a state s is written in terms of the values of other states s' in the environment. From these consistency equations, it is possible to derive various learning rules to compute the value function for a given policy π.

A popular set of algorithms used to efficiently estimate the value function bootstrap the agent's existing knowledge of the value function for states s' along with direct interaction with the environment to refine the value function estimate for its current state s. These methods are deemed temporal difference (TD) methods, as they generally use the reward prediction error (i.e., the discrepancy between the current reward vs. that anticipated by the bootstrapped estimate of the value function) to adjust the current estimate of the value function. Basic TD learning is an algorithm to estimate v^{π}. Let V^{π} denote the estimate of v^{π}. We can initialize $V(\cdot)$ however, and, when in a state s_t, follow the action prescribed by π, so that we reach the state s_{t+1} and receive a reward R_{t+1}. Then we update our estimate of the value function as follows:

$$V^{\pi}(s_t) \leftarrow V^{\pi}(s_t) + \alpha\big[R_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)\big] = V^{\pi}(s_t) + \alpha\delta_t \tag{9}$$

where α is the learning rate. This update proceeds until $V^{\pi} \to v^{\pi}$, which is guaranteed (over an infinite time horizon and with some assumptions about the learning rate and environment). The term δ_t in the update equation above is called the TD error, and expresses the reward prediction error of the current value function estimate.

We can also utilize the temporal difference error in control, or policy improvement, contexts. Here, we consider the process of learning q with the estimate Q (here, we drop the π notation). Prediction and control are processes that are necessarily intertwined: The agent seeks to *exploit* the information it has already learned in the predictive phase, while also needing to *explore* alternative actions in the control phase in order to possibly improve its policy. Consequently, popular RL algorithms interleave prediction and control in a way that explicitly reflects the exploration-exploitation trade-off.

Two flavors of TD learning—*on-* and *off-policy* control—reflect different approaches to balancing exploration and exploitation. First, we discuss on-policy algorithms. On-policy algorithms update the estimate of Q using actions taken

from the current policy π. We first consider the SARSA algorithm, an on-policy TD learning control algorithm. The agent starts in state s_t, takes action a_t, which lands it in state s_{t+1} (where the policy suggests taking action a_{t+1}) and the agent receives a reward R_{t+1}. Then we update the state-action value function as:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{10}$$

We can then update the policy to select actions that are, say, deemed valuable by the updated state-action function. What makes this algorithm *on-policy* is which bootstrap estimate is used to calculate the TD error: We use $Q(s_{t+1}, a_{t+1})$, where the action a_{t+1} is given by the current policy, to compute the reward prediction error. Under SARSA, because the agent's own policy drives behavior, exploration directly shapes the learned value function: suboptimal exploratory actions can reduce the bootstrap estimate and are therefore reflected in the values themselves. These estimates in turn shape behavior, and can lead to more conservative policies.

In contrast to on-policy algoirthms, off-policy algorithms do not rely on the current policy to derive the reward prediction error. The most straightforward example of an off-policy algorithm is Q-learning. In Q-learning, unlike SARSA, we inspect our current estimate of Q and use the estimate of the *best possible action*. Indeed, as mentioned above, the agent starts in state s_t, takes action a_t, which lands it in state s_{t+1} and the agent receives a reward R_{t+1}. Here, the update rule is now:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)] \tag{11}$$

Notice that instead of following the action prescribed by the policy, our bootstraped estimate is now $\max_{a'} Q(s_{t+1}, a')$, or the Q value associated with the action taken in s_{t+1} that is predicted to maximize the agent's return. Further, in SARSA, exploration (through the policy used to select the next action) partially influences learned behavior by exerting influence on learned values. In contrast, in Q-learning, the agent greedily follows the best action to derive the bootstrapped estimate to update the value estimate. As such, exploration exerts less of an influence on value estimates, and consequently, on the derived policy.

5 Bringing it all together: RL for cognitive modeling

Having expanded the different components of RL and viewed it from different perspectives, in this section, we focus on its utility for cognitive modeling. Cognitive modeling entails transforming cognitive theories into precise mathematical computations with the goal of better understanding behavior and the processes underlying it. The approach has become increasingly popular, with many recent psychology studies including computational models [Palminteri et al., 2017].

In Fig. 4, we illustrate the different steps involved in cognitive modeling [Wilson and Collins, 2019]. The first step in cognitive modeling is to find a suitable class of models to study the scientific question at hand. The chosen models can span different classes, where each model class provides a different explanation. While a reinforcement learning model has parameters that allow one to study how the valence of the rewards determines how people learn trial-by-trial, drift diffusion models explain how evidence accumulates over trials and leads to a decision. Once chosen, the model is fitted to human choices, which involves finding the parameters that capture human data the best. Traditionally, psychologists use maximum likelihood estimation (MLE) to find the best parameters, where the parameters are chosen to maximize the likelihood of human data. But recently, Bayesian model fitting has become common, where model parameters are fit to each participant's choices separately conditioned on a population-level prior. In contrast to MLE-based approaches, Bayesian model fitting leads to much more stable parameter estimates, provides uncertainty estimates for fitted parameters, allows incorporation of prior knowledge, and is also more reliable for fitting parameters in experiments with only a limited number of trials and participants. Alternatively, when a large dataset is available, it is possible to perform a model comparison using cross-validation [Hastie et al., 2009], as is standard in machine learning. In this approach, the model parameters are fit to a set of participants, and its capacity to generalize to left-out participants is measured. An advantage of this approach is that it circumvents the problem of choosing the number of free parameters, which is tricky when the parameters are strongly correlated as in certain RL models.

Depending on model fit, it may be worthwhile to broaden the class of models considered by incorporating additional hypotheses. For example, in an RL model, it is possible to add a new parameter to capture the differences in learning from reward and punishment. Once researchers are satisfied with the candidate models, they compare them against each other in terms of how well they can account for human data taking into account their complexity. Measures, such as the Akaike Information Criterion [Akaike, 1974] and the Bayesian Information Criterion [Schwarz, 1978], are well suited for this purpose because they penalize the MLE with the complexity of the model. The resulting model, i.e., the model that achieves the best information criterion, can serve as an *in silico* model of human behavior. It offers an explanation of the underlying cognitive process through interpretable parameters and a quantitative description of complex multi-step problems that are hard to verbalize. Additionally, it can be used to make qualitative predictions that can then be compared against humans.

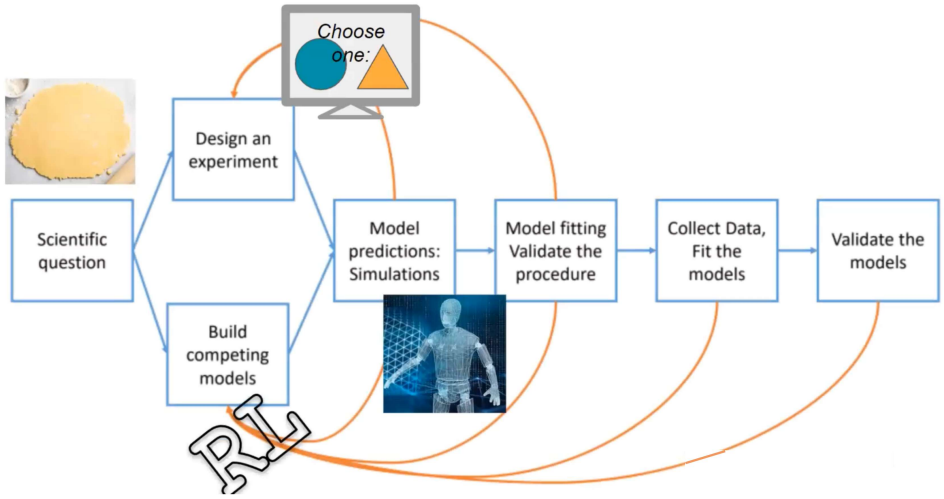

Figure 4: Recipe for cognitive modeling. This flowchart outlines the iterative process of cognitive modeling, beginning with a scientific question about a cognitive process (e.g., how people learn the value of different options based on rewards). To address this question, researchers design an experiment that probes the underlying mechanisms (e.g., a multi-armed bandit task) and formulate computational models that capture the hypothesized processes (e.g., the Rescorla–Wagner model). Model predictions are generated through simulations and validated for identifiability by fitting the models to simulated data. The experiment is then conducted, and human behavior is compared against the model predictions. To quantitatively assess model performance, each candidate model is fitted to the human data, and model comparison techniques are used to evaluate their relative fit. The winning model, which is the model that best accounts for the human data, is interpreted as providing the most plausible computational account of the cognitive process under investigation, relative to the alternatives considered. Figure reproduced with permission from Maria Eckstein.

Reversal learning

To illustrate the steps involved in cognitive modeling, let us consider a bandit task setting that has previously been used to study the developmental trajectory of reversal learning [Eckstein et al., 2022]. In this study, the authors used a two-armed bandit task, in which participants had two boxes to choose from (see Fig. 5 A). The action participants take in this setting corresponds to the two boxes, states are their positions on the screen, and the rewards, provided after each choice, are either 0 or 1. The reward probability for the boxes are assigned apriori, 0.7 if the chosen box was correct and 0.0 if it is incorrect. This means that even if one box is more rewarding than the other, feedback is not always rewarding, as there is some stochasticity in reward delivery. In the reversal learning setting, the experimenter abruptly changes the reward probability assigned to the boxes every few trials, unbeknown to the participant. The participants have to infer the change of reward probability from the sampled rewards and reverse their strategy. Hence, it is called reversal learning. Eckstein et al. ran this task on participants whose age continuously varied between 8 and 30 and to their surprise, found that teenagers, approximately 13-15 years old, were better than participants in any other age group, as shown in Fig. 5 B. To better understand the algorithm teens use to learn the task and how it differs from other age groups, they turned to RL for cognitive modeling, following the steps shown in Fig. 4. Specifically, they used a variant of the Q-learning algorithm discussed in Section 4 that includes a perseverance parameter, which makes it likely to repeat the option chosen before, and two different learning rates for positive reward and no reward. Inspecting the parameters of the fitted model, they first found that the perseverance parameter increased with age, indicating that people show a strong tendency to pick the same option as before with age, but plateau after age 15 (see Fig. 5 C). Second, they found that the learning rates for negative outcomes were the lowest for adolescents (aged 13-15) (see Fig. 5 D). This suggests that adolescents take negative outcomes less into account than adults or younger children. Combined with their lower perseverance rate, this means that their learned value estimates for the two boxes reflect the ground-truth reward probability much more than adults. Together, these results provide an explanation, at an algorithmic level, of why teenagers perform much better than adults or children in the reversal learning task. The authors also compared the explanations derived from RL modeling with those from Bayesian inference and found the modeling results to be highly corroborated; see [Eckstein et al., 2022] for details.

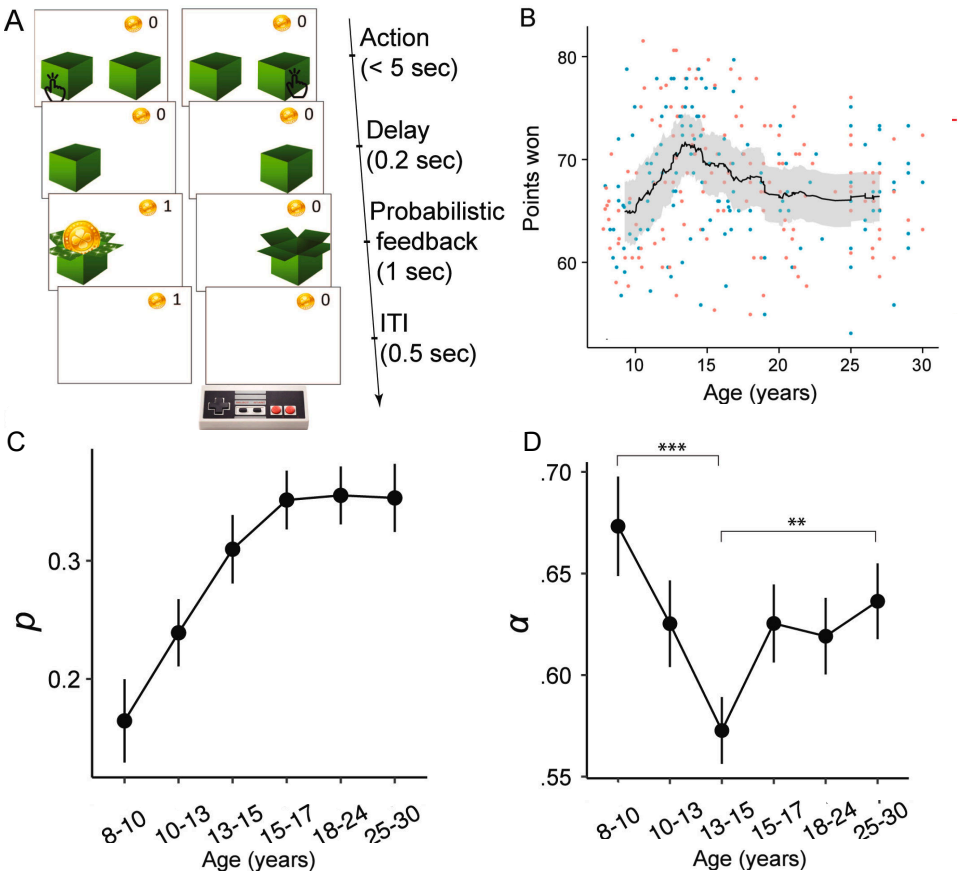

Figure 5: Developmental trends in reversal learning (A) Structure of the probabilistic reward learning task trial, including action, delay, probabilistic feedback, and inter-trial interval (ITI). (B) Total points won across different age groups. Each dot shows one participant, color denotes sex (Red: Female; Blue: Male). Lines show rolling averages, shades the standard error of the mean. (C, D) Fitted model parameters for the winning RL model across different age groups, with (C) showing perseverance rate, p and (D) showing learning rate for negative outcomes. Stars significant effects of age on model parameters. Stars on top of brackets show differences between groups as revealed by t-tests. Dots (means) and error bars (standard errors) show the results of the age-less fitting model, providing an unbiased representation of individual fits. Figures adapted from Eckstein et al. [2022] under the CC BY-NC-ND 4.0 license.

Model-based vs. model-free

Do humans accumulate reward habitually or deliberatively? In other words, do humans rehearse learned strategies that they know are going to yield at least some reward (habitual behavior) or do they consider all possible contingencies of each step in a behavioral sequence and update behavior flexibly (deliberative behavior)? Using RL as a cognitive model offers a convenient language for disambiguating habitual versus deliberative behavior.

Habitual behavior is captured by the model-free approach. Model-free algorithms, unlike model-based algorithms, do not rely on the agent's knowledge (or estimate) of the world dynamics. Instead, the agent (still iteratively) updates its estimate of the value function using just its interactions with the environment. TD learning (outlined above) is a classic model-free algorithm.

Conversely, model-based approaches capture the notion of deliberative behavior. Unlike model-free methods, which bootstrap knowledge of immediate successive states, model-based approaches incorporate additional prior (or learned) knowledge of the environment. The model-based approach to finding the optimal policy involves estimating the optimal value function by representing an explicit running calculation of the world dynamics of the MDP given by p. In other words, the agent can "predict," given a state s and a candidate action a, what the resultant state s' will be, and what reward r it will get. Value and policy iteration, which iteratively determine the optimal value functions by turning the

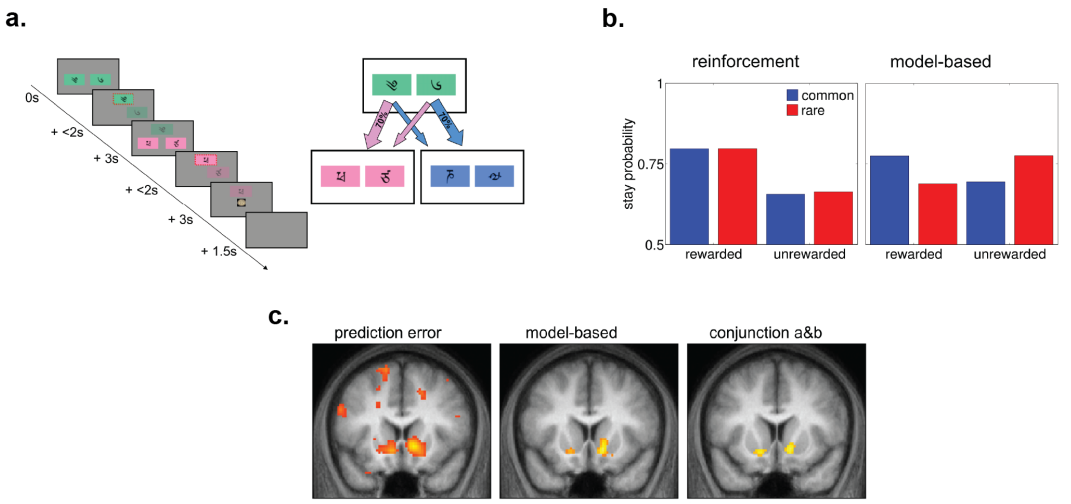

Figure 6: The two step task (a., figure adapted in lecture slides, taken from Daw et al. [2011]), and different predictions furnished by different RL algorithms (model-based vs. model-free), b., (figure adapted in lecture slides, taken from Daw et al. [2011]). A hybrid model that arbitrates between model-based and model-free algorithms best explains human decision-making, and finds correlates in the brain (c., figure adapted in lecture slides, taken from Daw et al. [2011]).

right hand side of the Bellman equation into an update rule, implicitly assume the agent's knowledge of the world dynamics, p, and thus are model-based algorithms.

A landmark experimental paradigm for probing whether humans rely more on habitual or deliberative reasoning is the two-step task Daw et al. [2011] (Fig. 6a). In this task, subjects start in stage 1 (which is comprised of two states; for now state $1A$ and state $1B$), and take an action that will lead them (with some stochasticity) towards two possible outcomes in stage 2, each with two states itself (so for outcome 1: states $2A_1$ and $2B_1$, and for outcome 2: states $2A_2$ and $2B_2$). In stage 2, one state in each outcome is rewarded. The transition probabilities from states 1A and 1B to either of the outcomes are symmetric: $p(2A_1|2B_1||1A) = q$ and $p(2A_2|2B_2||1A) = 1 - q$, whereas $p(2A_2|2B_2||1B) = q$ and $p(2A_1|2B_1||1B) = 1 - q$. Model-free and model-based algorithms make different predictions about how subjects' preference over actions will change as a function of the stochasticity that governs state transitions. In the model-free case, the probability of transitioning to either one of the stage 2 states is not considered; thus, the reward obtained in either one of the stage 2 states will result in identical changes to the estimated value of stage 1. Conversely, in the model-based case, the value updates are weighted by the probabilities of transitioning to either one of the stage 2 states. For example, if the agent starts in $1A$ and makes a relatively unlikely transition to one of the stage 2 outcomes and gets rewarded, it will factor in that its transition was relatively unlikely when updating the value of state $1A$. In Fig. 6b, it is easy to see how these different algorithms make different predictions about how the rewards obtained under common and rare transitions differentially impact the value estimations of starting states.

This differing prediction provides a hypothesis about the kind of valuation behavior we should see if humans deploy model-based (deliberative) or model-free (habitual) evaluation. Such experiments revealed that human decision-making is not well captured as solely model-based or model-free, but is well-captured as a hybrid model of the two. Using such models, we can also correlate brain activity (using fMRI) that is responsible for implementing the model-based and model-free components that subserve human decision making (Fig. 6c).

6 Theory driven vs data driven models

In the effort to predict and reproduce human learning behavior (Fig.7A), we observe a contrast between two modeling paradigms: structured reinforcement learning (RL) models, which rely on explicitly defined internal states such as value (Fig.7B), and artificial neural networks (ANNs), which operate as flexible function approximators that learn distributed representations to maximize task performance (Fig.7C).

Consider a bandit task in which humans must choose among four targets to maximize cumulative reward, where the reward probabilities drift over time. A theory-driven model of this behavior typically includes a set of theoretically motivated variables. For instance, an action a has an internal value $Q_a(t)$, with initial value of Q_{init}, and approximates

the expected reward. These Q-values are updated via a delta learning rule. In addition, an RL module might include a context variable $c_t(a_t)$ to capture choice nuances such as perseverance (Fig.7D).

Alternatively, recurrent neural networks (RNN) can be employed to model these dynamics. While standard RL models often assume a Markovian state representation, humans may rely on complex historical dependencies, effectively treating the task as partially observable. An RNN thus replaces explicit parameters with latent state s_t. The latent state is encoded in high-dimensional hidden-layer activations, and updated by observations of reward (r_t) and action (a_t) to predict subsequent actions (a_{t+1}) (Fig.7F). While the theory-driven RL model offers a specified mechanism with few interpretable parameters, the RNN flexibly integrates inputs and hidden states to generate behavior, albeit with less transparency.

When trained to predict human choice behavior, RNN models consistently outperform classical RL models (Fig.7I, "Best RL vs. Vanilla RNN"). As illustrated in the performance comparison, the unconstrained RNN achieves a substantially higher predictive accuracy than the best-performing RL baseline, highlighting the gap between simple cognitive variables and high-dimensional distributed representations. However, this success comes at the cost of interpretability. One strategy to introduce interpretability into an ANN-based framework is to progressively replace components of a standard RL model with RNN modules. These modules compute conceptual parameters in a distributed manner, allowing us to compare the resulting "hybrid" models against fully distributed networks.

To explore this architecture, one might initially replace explicit RL computations with RNN modules that update Q_t and c_t (Fig.7E). In practice, however, this modification alone yields no significant advantage over classic RL (see Fig.7E,I, "RL-ANN"), suggesting that standard RL formulations may miss fundamental learning processes. Behavioral data can guide further improvements. Learning depends on both chosen and unchosen available actions. Consequently, we can enhance the RNN-RL model with additional inputs for the value estimates of all actions and context values, making learning a function of (r_t, a_t, Q_t, c_t). Although these "context-ANN" models predict behavior more accurately than basic RL models, visual inspection of Fig.7G,I reveals they still fail to capture the full variance explained by the vanilla RNN.

Building on this, we can investigate other facets of human behavior that might enhance predictive performance while retaining interpretability. Because a vanilla RNN intrinsically encodes memory in its hidden state, we can enable an RL model to tap into this capacity by providing RNN modules with a history of previous latent states ($s_t^{(r)}$, $s_t^{(a)}$) alongside r_t and a_t. This "memory ANN" model matches the vanilla RNN in predicting human behavior despite its more constrained architecture (Fig.7H,I).

Finally, the structural constraints of the "memory ANN" architecture allow us to probe learning mechanisms. By examining the reward module, we can construct an input-output mapping between the reward r_t and the updated value Q_{t+1}. Notably, because the reward module lacks direct access to prior values or previous actions, it updates values directly from the observed reward. This contrasts with incremental updating via an explicit delta rule, suggesting the model captures human-like learning dynamics through high-dimensional, memory-based processes rather than simple incremental updates.

7 Conclusion

In this Lecture Notes, we have explored reinforcement learning (RL) as a powerful computational theory that bridges fields of psychology, neuroscience, and artificial intelligence. We examined its roots in psychological theories of associative learning, its neural correlates in dopamine-based reward prediction error signals, and its formalization within machine learning using frameworks like Markov Decision Processes and algorithms such as Temporal Difference Learning.

Building on this foundation, we demonstrated the utility of RL for cognitive modeling. By fitting specific RL models to behavioral data, researchers can dissect the mechanisms underlying decision-making behavior across individuals, as illustrated by studies on reversal learning. Different classes of RL models can be utilized to distinguish between model-free (habitual) and model-based (deliberative) control systems. Finally, the integration of theory-driven RL principles with data-driven artificial neural networks represents a promising frontier, potentially yielding models that are both highly predictive and interpretable.

Together, this work highlights RL's unique capacity to link computational mechanisms, neural processes, and behavior. While current models continue to evolve, particularly through hybrid RL-ANN designs, the RL framework remains central to understanding adaptive behavior in complex, naturalistic settings. Furthermore, its principles continue to guide the design of artificial agents capable of flexibly adapting to new environments, tasks, and goals.

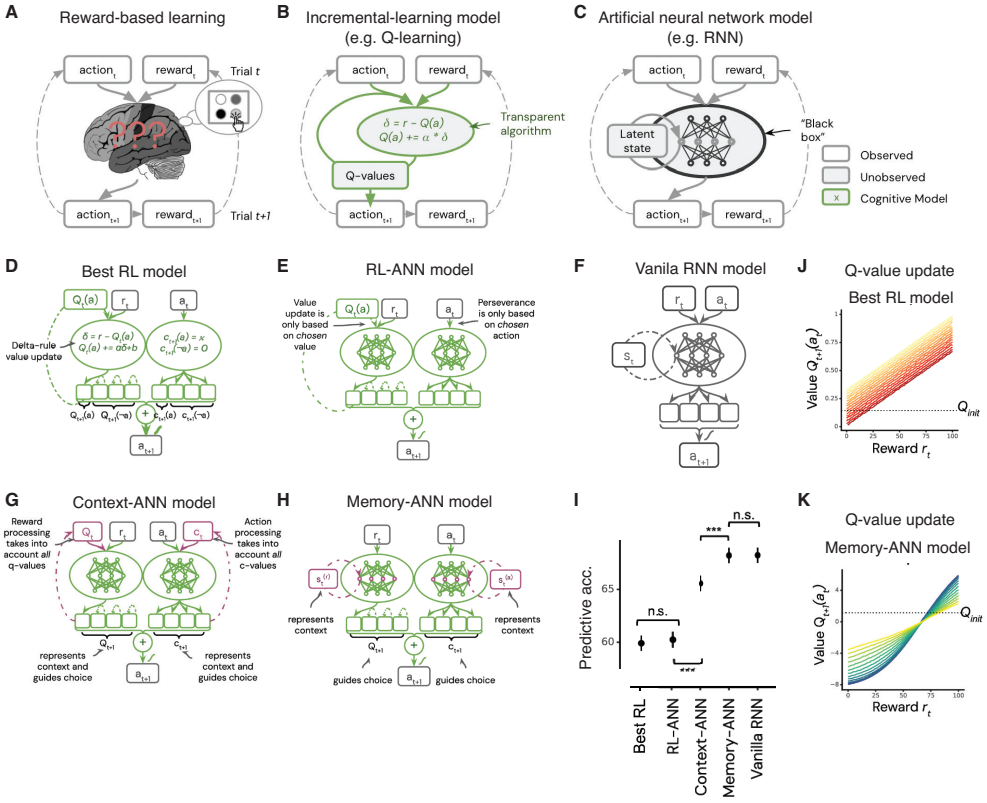

Figure 7: Bridging Reinforcement Learning and Artificial Neural Networks for Modeling Cognition. **(A)** Schematic of a reward-based learning task where choices are informed by past actions and rewards across trials. **(B)** A Q-learning model exemplifies a 'transparent' cognitive model, using explicit variables (e.g., Q-values, prediction errors) and algorithms (e.g., Q-value updating) to describe the decision-making process. **(C)** In contrast, data-driven black box ANN models are trained directly on sequences of actions and rewards without predefined assumptions about the underlying computations or explicit variables. **(D-H)** Architectures bridging cognitive models and ANNs: **(D)** A benchmark 'Best RL' cognitive model. **(E)** A hybrid RL-ANN model incorporating an explicit Q-value representation. **(F)** A standard 'Vanilla RNN' without explicit cognitive or RL variables. **(G)** A Context-ANN model where an RNN integrates value representations with explicit contextual information (c_t) to guide choices. **(H)** A Memory-ANN model where an RNN utilizes an internal memory state (m_t), updated based on past actions and rewards, to guide choices, further abstracting internal computations. **(I)** Comparison of model performance based on predictive accuracy for behavioral choices. Results indicate significant performance differences. The Memory-ANN and Vanilla RNN significantly outperform the Best RL and Context-ANN models. Differences between Memory-ANN and Vanilla RNN, and between Best RL and Context-ANN, are not statistically significant (n.s.) in this depiction (error bars likely represent standard error). **(J, K)** Visualization comparing the learned Q-value update dynamics as a function of reward (r_t) for the Best RL model (J) and the Memory-ANN model (K), demonstrating how different architectures learn to map rewards onto value predictions. Adapted from Eckstein et al., "Hybrid neural–cognitive models reveal how memory shapes human reward learning," *Nature Human Behaviour*, DOI: 10.1038/s41562-025-02324-0 (2026). Licensed under CC BY 4.0.

References

Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. 1(1):9–11, March 1998.

Nathaniel D Daw, Samuel J Gershman, Ben Seymour, Peter Dayan, and Raymond J Dolan. Model-based influences on humans' choices and striatal prediction errors. *Neuron*, 69(6):1204–1215, March 2011.

A. Ross Otto, Anya Skatova, Seth Madlon-Kay, and Nathaniel D. Daw. Cognitive control predicts use of model-based reinforcement learning. *Journal of Cognitive Neuroscience*, 27(2):319–333, February 2015. ISSN 1530-8898. doi:10.1162/jocn_a_00709. URL `http://dx.doi.org/10.1162/jocn_a_00709`.

Camillo Padoa-Schioppa and John A Assad. Neurons in the orbitofrontal cortex encode economic value. *Nature*, 441 (7090):223–226, May 2006.

Samuel M McClure, Gregory S Berns, and P Read Montague. Temporal prediction errors in a passive learning task activate human striatum. *Neuron*, 38(2):339–346, April 2003.

Kei Oyama, István Hernádi, Toshio Iijima, and Ken-Ichiro Tsutsui. Reward prediction error coding in dorsal striatal neurons. *Journal of Neuroscience*, 30(34):11447–11457, August 2010.

Paul W Glimcher. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proc. Natl. Acad. Sci. U. S. A.*, 108 Suppl 3(supplement_3):15647–15654, September 2011.

W Schultz, P Dayan, and P R Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, March 1997.

Sarah X Luo and Eric J Huang. Dopaminergic neurons and brain reward pathways: From neurogenesis to circuit assembly. *Am. J. Pathol.*, 186(3):478–488, March 2016.

Roy A Wise. Dopamine, learning and motivation. *Nat. Rev. Neurosci.*, 5(6):483–494, June 2004.

Mitsuko Watabe-Uchida, Neir Eshel, and Naoshige Uchida. Neural circuitry of reward prediction error. *Annu. Rev. Neurosci.*, 40:373–394, July 2017.

E. L. Thorndike. Animal intelligence: An experimental study of the associate processes in animals. *American Psychologist*, 53(10):1125–1127, 1998. ISSN 1935-990X. doi:10.1037/0003-066X.53.10.1125. URL `http://doi.apa.org/getdoi.cfm?doi=10.1037/0003-066X.53.10.1125`.

F. Nowell Jones and B. F. Skinner. The Behavior of Organisms: An Experimental Analysis. *The American Journal of Psychology*, 52(4):659, 1939. ISSN 00029556. doi:10.2307/1416495. URL `https://psycnet.apa.org/record/1939-00056-000`.

Ivan P. Pavlov. Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex. *Annals of neurosciences*, 17(3), 6 2010. ISSN 09727531. doi:10.5214/ans.0972-7531.1017309. URL `http://www.annalsofneurosciences.org/journal/index.php/annal/article/view/246`.

R A Rescorla and A R Wagner. A theory of Pavlovian conditioning, 1972. ISSN 19416016.

Leon J Kamin. Attention-like processes in classical conditioning. In *SYMP. ON AVERSIVE MOTIVATION MIAMI*, number TR-5, 1967.

Anthony Dickinson. Actions and habits: the development of behavioural autonomy. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 308(1135):67–78, 1985.

Wendy Wood and David T Neal. A new look at habits and the habit-goal interface. *Psychological review*, 114(4):843, 2007.

Ethan S Bromberg-Martin, Masayuki Matsumoto, and Okihide Hikosaka. Dopamine in motivational control: rewarding, aversive, and alerting. *Neuron*, 68(5):815–834, December 2010.

R S Sutton and A G Barto. Toward a modern theory of adaptive networks: expectation and prediction. *Psychol. Rev.*, 88 (2):135–170, March 1981.

John P O'Doherty, Peter Dayan, Karl Friston, Hugo Critchley, and Raymond J Dolan. Temporal difference models and reward-related learning in the human brain. *Neuron*, 38(2):329–337, April 2003.

Dino J Levy and Paul W Glimcher. The root of all value: a neural common currency for choice. *Curr. Opin. Neurobiol.*, 22(6):1027–1038, December 2012.

B Knutson, C M Adams, G W Fong, and D Hommer. Anticipation of increasing monetary reward selectively recruits nucleus accumbens. *J. Neurosci.*, 21(16):RC159, August 2001.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Stefano Palminteri, Valentin Wyart, and Etienne Koechlin. The importance of falsification in computational cognitive modeling. *Trends in cognitive sciences*, 21(6):425–433, 2017.

Robert C Wilson and Anne GE Collins. Ten simple rules for the computational modeling of behavioral data. *Elife*, 8: e49547, 2019.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6): 716–723, 1974.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

Maria K Eckstein, Sarah L Master, Ronald E Dahl, Linda Wilbrecht, and Anne GE Collins. Reinforcement learning and bayesian inference provide complementary models for the unique advantage of adolescents in stochastic reversal. *Developmental Cognitive Neuroscience*, 55:101106, 2022.