A flexible and generalizable model of online latent-state learning

Amy L Cochran[1*] **and Josh M Cisler**[2]

***For correspondence:** cochran4@wisc.edu (ALC)

[1]Department of Biostatistics and Medical Informatics, University of Wisconsin - Madison; [2]Department of Psychiatry, University of Wisconsin - Madison

Abstract Emerging computational models of learning and brain function hypothesize that learning agents use latent-state inferences, i.e. an ability to maintain beliefs over competing hypotheses. Current models of latent-state inferences, however, are limited by a narrow scope or unrealistic demands on human computation. Here, we present a model of latent-state inferences, combining the popular Rescorla-Wagner model with an approximate Bayesian filter. In simulation, we demonstrate the model's ability to reproduce diverse learning experiments. We then show the model explains participant behavior better than the Rescorla-Wagner model using a novel dataset (N=57 and N=24) of two experiments (average increase in log-likelihood: 6.1 [95% CI: 3.1–9.1] and 4.6 [95% CI: 2.2–6.9]). Lastly, we derive our model as an online algorithm to maximization likelihood estimation, establishing it as an efficient approach to outcome prediction. Establishing such a framework is a key step towards quantifying normative and pathological ranges of latent-state inferences in various contexts.

Introduction

Learning and decision-making are fundamental aspects of day-to-day human life. Indeed, many mental health disorders can be conceptualized from the perspective of biases or errors in learning and decision-making (***Huys et al., 2015***). Accordingly, the study of how humans learn and make decisions is an important topic of inquiry and the past two decades has witnessed a significant surge in the application of computational modeling approaches to the problem of human learning and decision-making (***Sutton et al., 1998***; ***Daw et al., 2006***; ***Doll et al., 2012***). One significant insight this literature has made is the differentiation of model-free and model-based learning (***Dayan and Berridge, 2014***; ***Gläscher et al., 2010***). Model-free learning refers to relatively simple updating of cached values based on trial-and-error experience. An early and influential example of model free learning is the Rescorla-Wagner (RW) model (***Rescorla et al., 1972***), which proposed that associative strength (i.e. the degree to which a cue predicted an outcome) updates in response to new experiences in proportion to the magnitude of a prediction error (i.e., how wrong the current prediction was). The RW model, and similar model-free formulations that it inspired (***Pearce and Hall, 1980***; ***Le Pelley, 2004***), powerfully explained many aspects of learning.

Nonetheless, a notable problem for model-free accounts of learning was the phenomenon of rapid recovery of responding following extinction learning (***Redish et al., 2007***). That is, model-free accounts of learning based on trial-by-trial updating predict that the return of responding following extinction should essentially occur at the same rate as initial learning. This prediction is not supported by a wealth of data demonstrating that extinguished responding can rapidly return via reinstatement (i.e., an un-signaled presentation of the outcome), context renewal (i.e., returning to the initial acquisition learning context), or spontaneous recovery (i.e., return of responding following the passage of time) (***Bouton, 2004***). This rapid return of extinguished responding is

an important phenomenon with implications for treatment of clinical disorders such as anxiety disorders and substance use (***Vervliet et al., 2013***; ***Redish, 2004***). Accordingly, solving this problem could potentially lead to substantially improved rates of treatment success.

One of the first applications of a model-based account of learning that could address rapid renewal of responding following extinction was by Redish and colleagues in 2007 (***Redish et al., 2007***). They argued that an agent performs two concurrent tasks during a learning experiment: 1) updating associative strength between a cue and an outcome, and 2) recognizing the current state of the experiment and storing separate associative strengths for the acquisition and extinction contexts. Rapid recovery of responding can be explained by the agent inferring that the current state of the experiment has changed to the acquisition phase and therefore responding to the cue with the associative strength stored for the acquisition phase. Though this initial formulation had limitations (***Gershman et al., 2010***), it stimulated subsequent research developing latent-state models of reinforcement learning (***Gershman and Hartley, 2015***; ***Gershman and Niv, 2012***; ***Gershman et al., 2015b***). For example, if an agent is assumed to infer latent states to explain observed relationships between stimuli, actions, and outcomes, and if inferring separate latent states for acquisition and extinction phases of an experiment explains rapid recovery of responding, then it follows that blurring the experimental distinction between acquisition and extinction phases would result in less recovery of responding following extinction. That is, if the contingency between cue and outcome (e.g., 50% contingency) during the acquisition phase slowly transitions to extinction, rather than an abrupt change, then an agent is less likely to infer a separate extinction context and more likely to modify the initial acquisition memory. Because the acquisition associative strength is lower, less subsequent recovery of responding would be expected in a recall test. A carefully designed experiment using an animal model demonstrated exactly this observation, providing strong evidence for a latent-state model of learning (***Gershman et al., 2013***).

The application and utility of latent-state learning models is not confined to explaining recovery of responding following extinction. Tolman's early theory of cognitive maps (***Tolman, 1948***) posits that an agent forms abstract representations of a task's state space. There has been a recent resurgence of interest in this concept of cognitive maps in the cognitive neuroscience field informed by latent-state computational models (***Gershman and Niv, 2012***; ***Stalnaker et al., 2015***; ***Wikenheiser and Schoenbaum, 2016***). This work, conducted both using animal and human experimental models, suggests that essential functions of the orbitofrontal cortex (OFC) and hippocampus are the encoding and representation of latent task space (i.e., forming cognitive maps of a task's state space) (***Wikenheiser and Schoenbaum, 2016***). This theory potentially powerfully describes a superordinate process that explains why the OFC and hippocampus are implicated in such a diverse array of cognitive and emotional functions. This theory also has implications that tie back into extinction learning and clinical disorders and suggest a novel understanding of the role of the OFC in mediating poor extinction learning.

As can be seen, latent-state theories have significant implications for our understanding of normative and disordered mechanisms of learning. Despite the significant progress in this area, a current weakness is that existing computational models of latent-state learning suffer from several limitations. These include imposing large demands on human computation and memory (e.g., keeping track of past outcomes), failing to describe more traditional experiments of associative learning, and requiring assumptions that restrict which experiments can be described. An additional limitation is that researchers have used different models to explain different learning tasks. These disparate models negate a unified account of latent-state learning. Indeed, one of the appeals of computational modeling applied to the problem of decision-making is that the learning theory is mathematically formalized (***Farrell and Lewandowsky, 2010***). This strength of computational modeling is lost when the mathematical formulation differs with each study and task because the underlying learning theory also necessarily differs across tasks. By contrast, having a single model that can characterize latent-state learning across the many different tasks used to probe latent-state learning would have both theoretical and practical significance. Theoretical significance

would come from formalizing a theory of latent-state learning that generalizes across tasks, thereby potentially powerfully describing the essential nature of latent-state learning. Practical significance would come from providing the research community with a generalizable and flexible model that can be used to integrate hypotheses and findings across the many research domains that propose latent-state models of learning. The purpose of the current work is to develop and test this form of latent-state model of learning.

Model

We present a model of latent-learning to describe behavior in a typical experiment consisting of sequence of trials. Each trial exposes a learning agent to a collection of cues followed by rewards (and/or penalties). Through the experiment, the agent learns to associate cues to rewards. In the Appendix, we show mathematically that our model of latent-state learning is an online approach to reward prediction.

We use the following notation. Cue n on trial t is denoted by $c_n(t)$ and takes a value of 1 when the cue is present and 0 otherwise. Cues are collected in a vector $\vec{c}(t)$. Rewards on trial t are denoted by $R(t)$. After trial t, the strength of the association between cue n and rewards is denoted by $V_n(t)$. An apostrophe denotes transposition.

Building upon the Rescorla-Wagner (RW) model

The RW model proposes associative strength is updated according to:

$$V_n(t) = \begin{cases} V_n(t-1) + \alpha_n(t)\left(R(t) - \sum_{n\,:\,c_n(t)=1} V_n(t-1)\right) & \textit{if } c_n(t) = 1 \\ V_n(t-1) & \textit{if } c_n(t) = 0, \end{cases}$$

with initial associative strength $V_n(0) = 0$. Term $\alpha_n(t)$ is referred to as *associability* and is constant in the RW model. Collecting associative strengths in a vector $\vec{V}(t)$, we can express this update concisely as

$$\vec{V}(t) = \vec{V}(t-1) + A(t)\,\vec{c}(t)\left(R(t) - \vec{c}(t)'\vec{V}(t-1)\right).$$

The term $A(t)$ is now a matrix capturing associability and in the RW model, is diagonal with terms $\alpha_n(t)$ along its diagonal. Changes in associative strength depends on the learning agent's current expectation or best guess $\vec{c}(t)'\vec{V}(t-1)$ for rewards given observed cues. If their expectation exceeds the actual reward, then associative strength declines for each cue present. If their expectation is below the actual reward, then associative strength increases for each cue present. Note that by updating associative strength in vector form, cues no longer need to be binary. That is, the same update formula applies for continuous cues.

Allowing for latent-state learning

We build upon the RW model by allowing a learning agent to propose L competing (i.e. mutually-exclusive) *hypotheses* about associations between cues and rewards (Fig 1). Each hypothesis is referred to as a *latent state* and is associated with its own RW model for predicting rewards from cues. The agent learns about associations for each latent state while learning about their belief in which latent state best explains current observations. We capture belief in latent state l on trial t as a positive variable $p_l(t)$ such that beliefs sum to one: $p_1(t) + \ldots + p_L(t) = 1$.

We let $\vec{V}_l(t)$ represent the vector of associative strengths and $A_l(t)$ represent the associability matrix for latent state l. Cue vectors are also allowed to vary between latent states with $\vec{c}_l(t)$ encoding cues relevant to latent-state l. For example, if latent state 1 hypothesizes that colors are solely important to a task, and latent-state 2 hypothesizes that patterns are solely important

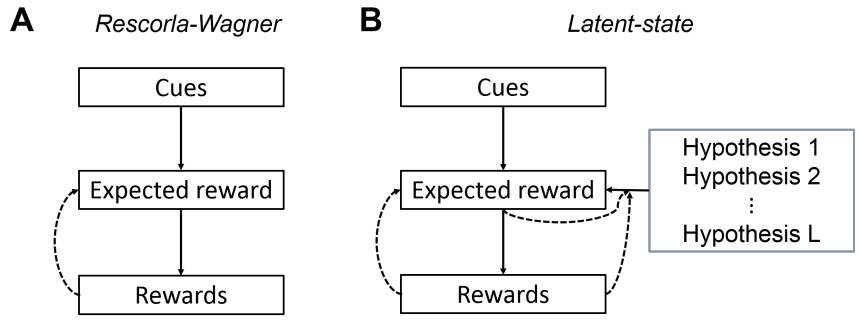


Figure 1. (A) In a typical Rescorla-Wagner model of learning, a learning agent estimates their expectations for rewards based on features and updates these expectations based on the difference between rewards and their expectations. **(B)** In a latent-state model of learning, a learning agent proposes hypotheses about the task, which when combined with observed cues allows them to estimate expected rewards. Based on how well their expectations predict rewards, the learning agent updates their expectations and their belief in each hypothesis.

to a task, then vector $\vec{c}_1(t)$ would encode only color cues, whereas vector $\vec{c}_2(t)$ would encode only patterns. The learning agent updates associative strength as before except for the subscript l:

$$\vec{V}_l(t) = \vec{V}_l(t-1) + A_l(t)\vec{c}_l(t)E_l(t), \tag{1}$$

where we used the shortened notation $E_l(t)$ to represent the prediction error for latent state l on trial t:

$$E_l(t) = R(t) - \vec{c}_l(t)'\vec{V}_l(t-1). \tag{2}$$

An associability process

Associability is constant in the RW model. By contrast, experiments such as those by Hall and Pearce (***Hall and Pearce, 1979***; ***Pearce and Hall, 1980***) suggest associative strength is path-dependent: associative strength depends on its history, changing over time (***Le Pelley, 2004***). We describe associability as depending on current beliefs and a matrix $B_l(t)$ which we refer to as an *effort* matrix:

$$A_l(t) = p_l(t)B_l(t)^{-1}. \tag{3}$$

The effort matrix $B_l(t)$ is updated according to

$$B_l(t) = (1-\alpha_0)B(t-1) + p_l(t)\vec{c}_l(t)\vec{c}_l(t)'. \tag{4}$$

with $B_l(0) = I$ and a parameter $\alpha_0 \in [0,1]$ controlling how much an individual weights older observations versus newer observations.

Increasing associability with beliefs is similar to the Mackintosh model (***Mackintosh, 1975***) in that associability increases when rewards are accurately predicted. Meanwhile, the effort matrix tracks how often a cue appears (diagonal entries) and a pair of cues appear (off-diagonal entries). When $\alpha_0 = 0$, the effort matrix increases over time, its inverse decreases, and associability declines, signaling an agent's expectations are converging. Decreasing associability mirrors stochastic optimization algorithms which often use step sizes proportional to $1/t$ to ensure updated variables converge to a local minimum. Associative strength, however, can change swiftly when an agent shifts to a new latent state in the presence of poor predictions. This feature is similar to Pearce-Hall model (***Pearce and Hall, 1980***), which supposes associability increases in the presence of uncertainty.

Updating latent-state beliefs

Beliefs link latent-states. Many latent-state learning models assume a learning agent reasons in an optimal Bayesian way (***Gershman and Niv, 2012***; ***Chan et al., 2016***; ***Wilson and Niv, 2012***). A learning agent assumes rewards are generated on trial t as follows: a latent random variable

$X(t) \in \{1, \dots, L\}$ is drawn from some distribution and rewards are drawn depending on the current latent state $X(t)$ and associated cues $\vec{c}_{X(t)}(t)$. Under these assumptions, one can infer a posterior distribution for these latent variables through Bayes law:

$$\mathbb{P}\left[X(1{:}t) = k(1{:}t) \,\middle|\, R(1{:}t)\right] \propto \mathbb{P}\left[X(1{:}t) = k(1{:}t)\right] \prod_{\tau=1}^{t} \mathbb{P}\left[R(\tau) \,\middle|\, X(\tau) = l(\tau)\right].$$

where we use the short-hand notation $X(1{:}t) = (X(1), \dots, X(t))$. Belief $p_l(t)$ in latent state l could then simply be the posterior probability that latent variable $X(t)$ is l:

$$\rho_l(t) = \mathbb{P}\left[X(t) = l \middle| R(1{:}t)\right] = \sum_{k(1),\dots,k(t-1)=1}^{L} \mathbb{P}\left[X(1{:}t-1) = k(1{:}t-1), X(t) = l \,\middle|\, R(1{:}t)\right].$$

From this expression, we see that optimal Bayesian inference requires enumerating over all possible sequences of latent states, the number of which grows exponentially with t. Such computation is generally considered optimistic for human computation (***Angela and Dayan, 2005***). Efforts are made reduce computation such as with particle method (***Gershman and Niv, 2012***) or assuming one transition between states (***Wilson and Niv, 2012***).

We capture a more general dynamic learning environment, as in ***Angela and Dayan*** (***2005***), by assuming latent states are a Markov chain in which a latent variable transitions to itself from one trial to the next with probability $1 - \gamma$ and transitions to a new state (all new states being equally-likely) with probability γ. This assumption reflects that learning often involve blocks of consecutive trials, or stages, in which rewards are generated in an identical manner. An optimal Bayesian filtering equation could then be used to update beliefs

$$\rho_l(t) \propto \left(\left(1 - \gamma - \frac{\gamma}{L-1}\right)\rho_k(t-1) + \frac{\gamma}{L-1}\right)\mathbb{P}\left[R(t) \middle| X(t) = l\right].$$

Unfortunately, this equation requires the learning agent knows the probability distribution of rewards for each latent variable. We thus relax the assumption reasons in an optimal Bayesian way. Instead, we initialize beliefs $p_l(0)$ and use an *approximate Bayesian filtering equation* (cf. ***Cappé*** (***2011***))

$$p_l(t) = \frac{1}{l_0}\left(\left(1 - \gamma - \frac{\gamma}{L-1}\right)p_l(t-1) + \frac{\gamma}{L-1}\right)\phi\left(\frac{E_l(t)}{\sigma(t-1)}\right), \tag{5}$$

where ϕ is the probability density function of standard normal random variable; $\sigma(t-1)$ is the agent's estimate of the standard deviation of rewards after trial $t-1$; and

$$l_0 = \sum_{l=1}^{L}\left(\left(1 - \gamma - \frac{\gamma}{L-1}\right)p_l(t-1) + \frac{\gamma}{L-1}\right)\phi\left(\frac{E_l(t)}{\sigma(t-1)}\right) \tag{6}$$

is a normalizing constant to ensure beliefs sum to 1. In other words, we replace the distribution/density function of rewards in the optimal Bayesian filtering equation with a normal density function with mean given by the agent's current estimate $\vec{c}_l(t)'\vec{V}_l(t-1)$ and standard deviation given by the agent's current estimate $\sigma(t-1)$. Even though we use a normal density function, rewards do not need to be normally-distributed or even continuous, as in ***Redish et al.*** (***2007***).

Measuring expected uncertainty

The learning agent may also track the standard deviation $\sigma(t)$. With only one standard deviation, we assume the learning agent estimates the standard deviation *pooled* over each latent state and each cue vector. We use the following update:

$$\sigma^2(t) = \sigma^2(t-1) + \beta_0\left(\overline{E^2}(t) - \sigma^2(t-1)\right) \tag{7}$$

where $\sigma^2(1) = \sigma_0^2$ is some initial estimate of the variance; β_0 is a constant associability parameter for variance $\sigma^2(t)$; and $\overline{E^2}(t)$ is the squared prediction error averaged over latent states:

$$\overline{E^2}(t) = \sum_l p_l(t)E_l(t)^2. \tag{8}$$

While updating variance is uncommon for most learning models, ***Angela and Dayan*** (***2005***) proposed a neural mechanism, via acetylcholine signaling, by which a learning agent keeps track of expected uncertainty in changing environments. As a staple to statistical inference, variance is a natural candidate for quantifying expected uncertainty.

Putting it all together

Our latent-state model of learning is summarized in Algorithm 1 based on Eq. (1)–(8). For each learning task, a researcher specifies the initial number L of latent-states, initial beliefs over latent states, cue vectors, and rewards, which are usually dictated by the task or research goal and fixed for all participants. What remains are four parameters: α_0, β_0, γ, and σ_0. However as illustrated in case studies, we often need only one more free parameter to tune to participant data than the base RW model.

Input: Number of latent states L, initial beliefs $p_l(0)$ and variance $\sigma^2(0)$, cue vectors $\vec{c}_l(t)$;
`/* Initialize states */`
for *each latent state* $l = 1, \ldots, L$ **do**
  Initialize associative strength $\vec{V}_l(0) = 0$;
  Initialize effort matrix $\vec{B}_l(0) = I$;
end
`/* Loop through trials */`
for *each trial* $t = 1, \ldots, T$ **do**
  `/* Update states */`
  for *each latent state* $l = 1, \ldots, L$ **do**
    Update latent-state belief $p_l(t)$ with Eq. (5)–(6);
    Update effort matrix $\vec{B}_l(t)$ with Eq. (4);
    Update associative strength $\vec{V}_l(t)$ with Eq. (1)–(3);
  end
  Update variance $\sigma^2(t)$ according to Eq. (7)–(8);
end

Algorithm 1: Model of latent-state learning with online updates

Variations of the model could be utilized. The effort matrices B_l and variance could be fixed to simplify the model. Alternatively, the model can be extended to allow for multidimensional rewards, decision-making, and unlimited states (see Appendix for details).

Simulation

We simulated behavior to verify that the proposed latent-state model summarized in Algorithm 1 can reproduce historically-important learning experiments. Parameters are fixed (Table 1). For comparison, we used a RW model with a constant learning rate of 0.15. Other model comparisons and simulation details can be found in the Appendix. Accompanying code is publicly-available at https://sites.google.com/site/amylouisecochran/.

For the latent-state model, we centered rewards $R(t)$ around neutral expectations, which in most examples meant $R(t) = 1/2$ if a reward was provided and $R(t) = -1/2$ if a reward was not provided. We prefer center rewards so that summed associative strength of cues is zero and yields neutral expectations at the start of the experiment.

Blocking and other effects

We first tested whether our latent-state learning model could reproduce learning effects that established the RW model as a powerful model of associative learning (Fig 2). In a blocking experiment, for instance, a learning agent is conditioned to associate a cue (Cue A) with a reward,

Parameter	Description	Value
α_0	Associative strength parameter	0
β_0	Standard deviation parameter	0.08
γ	Latent-state transitions	10^{-3}
σ_0	Initial standard deviation	0.5
τ	Soft-max parameter	5

Table 1. Fixed set of parameters used for all simulated experiments.

leading to a high associative strength for Cue A. Afterwards, Cue A is repeatedly presented with another cue (Cue B). The compound is then reinforced with a reward. Even though Cue B is reinforced, Cue B does not acquire the high associative strength as Cue A did. That is, conditioning of Cue A *blocked* conditioning of cue B. The RW model and our latent-state model predict blocking because of the way associative strength is updated in both models: updates depend on prediction error and predictions depend on the sum of associative strengths of present cues. Consequently, associative strength of Cue B changes slightly, because the compound of an excitatory Cue A and neutral Cue B yields small prediction errors.

Our latent-state model reproduces overexpectation and conditioned inhibition, two other effects explained by the RW model. An overexpectation experiment consists of reinforcing two cues separately with a reward and then later as a compound. When presented as a compound, associative strengths decrease. A conditioned inhibition experiment consists of intermixing the reinforcement of one cue (Cue A) with the omission of a reward when the cue is presented with another cue (Cue B). Cue A increases in associative strength while Cue B decreases in associative strength. As with blocking, the RW and latent-state model capture overexpectation and conditioned inhibition because of the way associative strength is updated. For overexpectation, expectations are twice actual rewards when the cues are presented together, yielding negative prediction errors and a decrease in associative strength for each cue. For conditioned inhibition, the negative associative strength of Cue B negates the positive associative strength of Cue A so that the sum of their associative strengths predicts the omission of the reward.

Learning effects not explained by the RW model

We also tested whether our latent-state learning model could reproduce historically-important learning effects not predicted by the RW model due to its assumption of constant associability (Fig 2).

Associability depends on history of cue presentation

Our latent-state model encodes the history of cue presentation in effort matrices $\boldsymbol{B}_l(t)$, endowing associabilty in our model with unique properties. One example is backwards blocking (Fig 3**A**). A compound of two cues (Cues A and B) is reinforced with a reward followed by only one of the cues (Cue A) being reinforced. In the second part, Cue A gains associative strength while Cue B loses associative strength *even though it is not presented*. The RW model cannot explain backwards blocking, because a cue must be present to change its associative strength (Fig 3). By contrast, the associative strength of Cue B correctly decreases in the second part for the latent-state model. Our latent-state model first learns the compound predicts a reward. Without additional information, it splits associative strength equally between the two cues. Later, our latent-state model learns Cue A predicts rewards. Reconciling both parts, our latent-state model increases the associative strength of Cue A while decreasing the associative strength of Cue B. In other words, our latent-state model learns about the *difference* in associative strengths between Cues A and B (Fig 3**C**). Mathematically, applying the inverse of $\boldsymbol{B}_l(t)$ to the cue vector $c_l(t)$ rotates the cue vector from being in the direction of Cue A to be in the direction of Cue A minus Cue B.

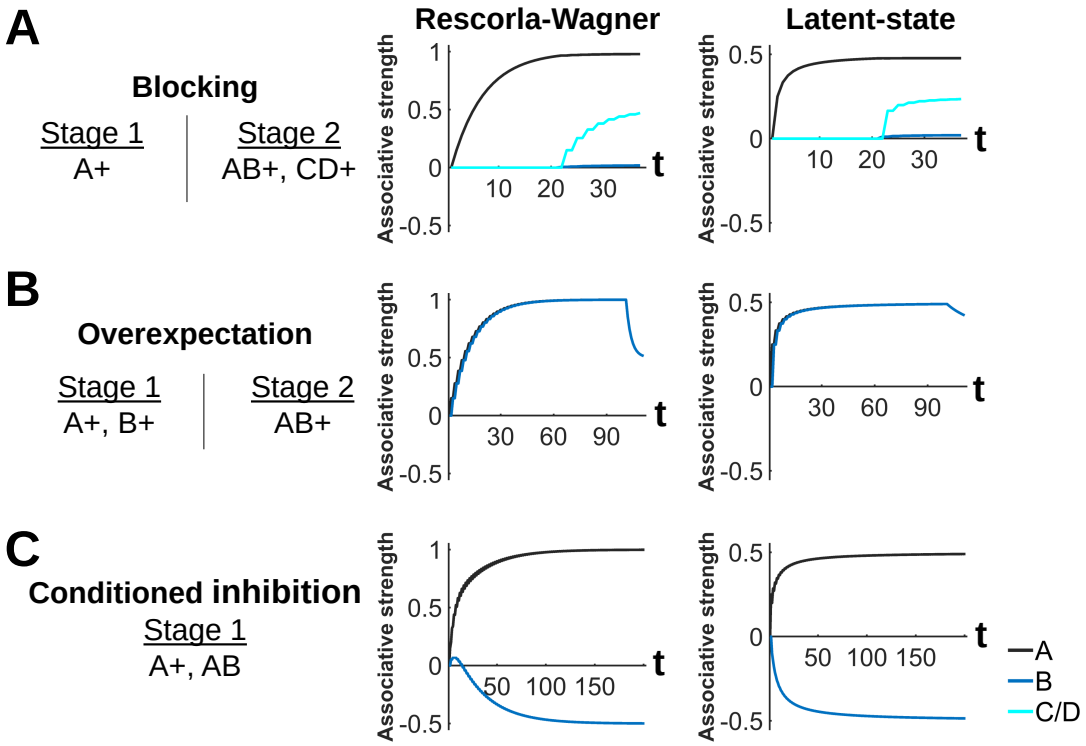


Figure 2. Simulated associative strengths of cues during blocking, overexpectation, and conditioned inhibition experiments, which the Rescorla-Wagner (RW) model famously could explain. Built upon the RW model, our model of latent-state learning can also explain these experiments.

History of cue presentation is also important for reproducing Experiment 1A in ***Rescorla*** (***2000***). This experiment consisted of reinforcing three cues with a reward (Cues A, C, and X) intermixed with not reinforcing two compounds (Compound BX and DX). Cues A, C, and X increased their associative strength, whereas Cues B and D decreased their associative strength. Compound AB was then reinforced. The RW model uses constant associability, which means that reinforcement of compound AB would result in identical changes to associative strength of Cue A as Cue B. By contrast, Rescorla found Cue B experienced a greater change in associative strength than Cue A.

We could reproduce this effect by including interaction terms to our latent-state model. Interactions terms consisted of adding an entry in the cue vector that is 1 when two cues are presented together and 0 otherwise. With interactions, the latent-state model learns that either Cue B inhibits the reward (food) directly or indirectly through Cue X in the first stage (***Rescorla, 2000***). This latter inhibition is captured by an interaction term between Cue X and Cue B (Fig 3**D**). Without any additional information, this inhibition is split evenly between Cue B and the interaction between Cue X and Cue B. The learning agent then learns that Cue B is not inhibitory when combined with Cue A. Reconciling both stages, the learning agent increases the associative strength of Cue B while decreasing the associative strength of the interaction between Cues B and X. Hence, Cue B has greater associability than Cue A during Stage 2. Mathematically, the effort matrix $\boldsymbol{B}_l(t)$ rotates the cue vector $c_l(t)$ to allow associability to behave differently than in an RW model.

Associability also depends on beliefs in latent-states.

Latent-state learning allows an agent to attribute sudden shifts in associations to new latent states, even when the shift is unsignalled. Associability starts relatively high when learning about a new state, thereby explaining the well-known Pearce-Hall effect (***Pearce and Hall, 1980***) in which associability remains high in unpredictable circumstances.

To illustrate, we consider Experiment 1B in ***Rescorla*** (***2000***), which involves the same condition-

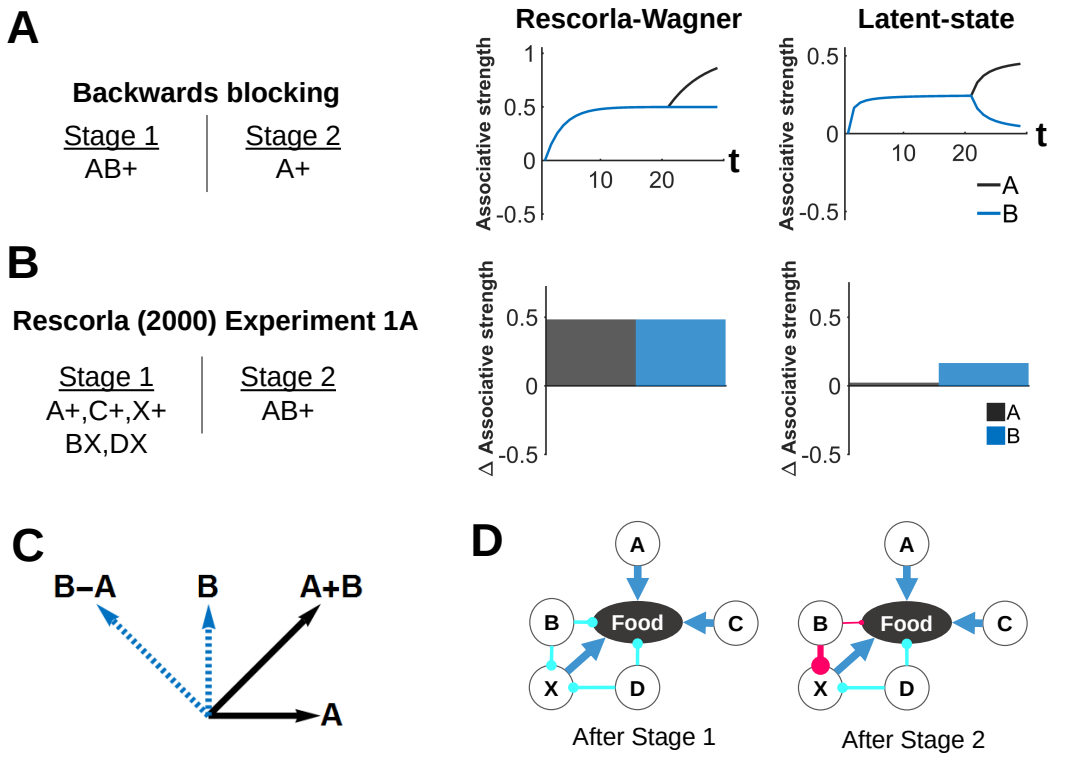


Figure 3. Associability depends on history of cue presentation. **(A)** Associative strength was simulated for a backwards blocking experiment in which associative strength of a cue decreases even though not presented. **(B)** Total change in associative strength was simulated during Stage 2 of Experiment 1A in ***Rescorla*** (***2000***). Associability is higher for Cue B during Stage 2 than Cue A. **(C)** Depiction of cue presentation as vectors during backwards blocking. Associability in our latent-state model depends on how presentations of cues co-vary in an experiment, allowing for associative strengths of both Cue A and Cue B to adjust even though Cue B is not present. **(D)** Illustration of associative strengths in our latent-state model during Rescorla's Experiment 1A . Greater associability for Cue B is attributed to learning that Cue B is not generally inhibitory, but specifically inhibitory when presented with Cue X. Excitatory cues are illustrated with triangle arrowheads; inhibitory cues are illustrated with circles.

ing schedule as Experiment 1A except compound AB is *not* reinforced in Stage 2 (Fig 4). As with Experiment 1B, the RW model would predict that Cue A would undergo the same change in associative strength as Cue B. Rescorla, however, found that Cue A decreased more in associative strength than Cue B. Again including interaction terms, our latent-state model explains that differences in associability arises because of a shift in latent states. The omission of a reward is so surprising to the learning agent—Cue A was always reinforced on prior trials—that they correctly infer a change in conditions and switch their beliefs to a new latent state in which they do not expect Cues A and B to be reinforced. Since rewards were originally expected to follow Cue A but not cue B, the associative strength of Cue A experiences the greater change.

A shift in latent state beliefs can also explain explain Experiment 1 in ***Wilson et al.*** (***1992***). This experiment involved two groups, Group C and Group E. For Group C, the experiment consisted of alternating between reinforcing and not reinforcing two cues (a tone and a light) with a reward (food), followed by reinforcing just the light. Group E had similar experiment conditions, except the tone was omitted in the middle of the experiment on non-reinforced trials. The omission was expected to decrease the associative strength of the light for Group E relative to Group C, which would be observed when the light was paired with food. Surprisingly, Group E responded more favorably to the light than Group C. The leading explanation was that associability of the light was higher for Group E than Group C (***Le Pelley, 2004***).

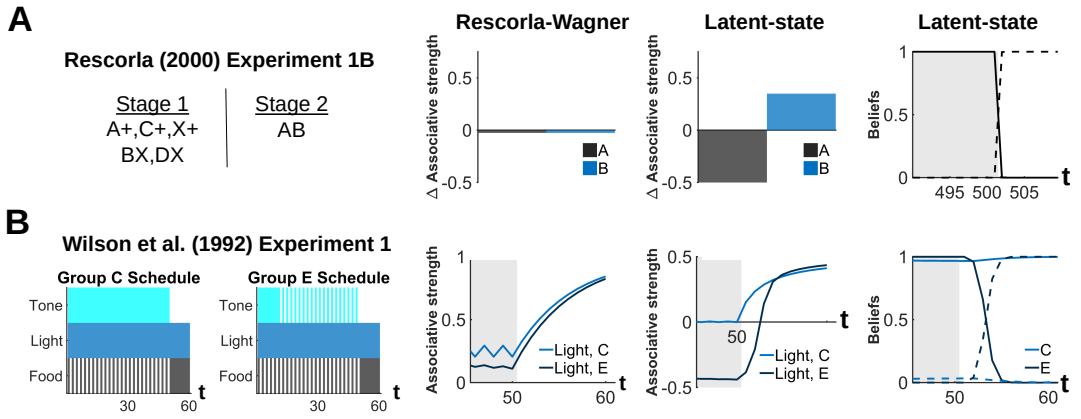


Figure 4. Associability depends on latent-state beliefs. **(A)** Total change in associative strength was simulated during Stage 2 of Experiment 1B from Rescorla (2000) (***Rescorla, 2000***). Associability was more greater for Cue B during Stage 2 than Cue A. **(B)** Associative strength was simulated for two groups in Experiment 1 from Wilson (***Wilson et al., 1992***), which pointed to greater associability for the light in Group E than Group C in the final trials of the experiment. In both experiments, our latent-state model attributes differences in associability to correctly inferring a change in cue-reward contingencies and thereby switching beliefs in one new latent state (solid line) to another latent state (dashed line).

In contrast to the RW model, our latent-state model offers an explanation: that while indeed the associative strength of the light does decrease for Group E relative to Group C, the pairing of the light solely with food is so surprising to Group E that they correctly infer the change in experimental conditions thereby shifting to a new latent state to build new expectations for the light (Fig 4**B**). This shift would be observed as higher associability of the light in Group E than Group C, agreeing with experimental results.

Learning with latent-states

To complete our simulation study, we tested if our latent-state model could describe more recent experiments designed to examine latent-state learning.

Recall

Latent-state learning was offered as an explanation of fear recall in the classic two-day fear conditioning and recall experiment (***Redish et al., 2007***; ***Gershman and Hartley, 2015***). To examine how latent-state learning influences recall, we simulated a two-arm bandit task in which the probability of receiving a reward for each arm switched twice through the experiment (Fig 5). When arm-reward contingencies first shift, our latent-state learner correctly detects the shift and switches their beliefs to a new latent state. By detecting this shift, the agent can both construct new associative strengths for each arm and preserve the old associative strengths that were accurate for the first block of trials. These associative strengths are later *recalled* when arm-reward contingencies revert back and the agent switches their beliefs back to the original latent state.

Weighing hypotheses

While a learning agent may naturally infer (and recall) latent-states to handle unsignalled changes, some experiments explicitly instruct individuals to weigh competing hypotheses. In the 3-arm bandit game in ***Niv et al.*** (***2015***), for instance, each arm had three feature dimensions (pattern, color, shape), with a feature taking one of three possible values along each dimension (dots, bricks, squiggles; red, blue, green; square, circle, triangle) and arms never sharing the same feature. Participants were instructed that only one feature dimension was relevant for predicting the arm with the highest rewards and that a certain “target" feature will generate higher rewards on average. Our latent-state model can represent the three competing hypotheses: patterns are relevant, colors

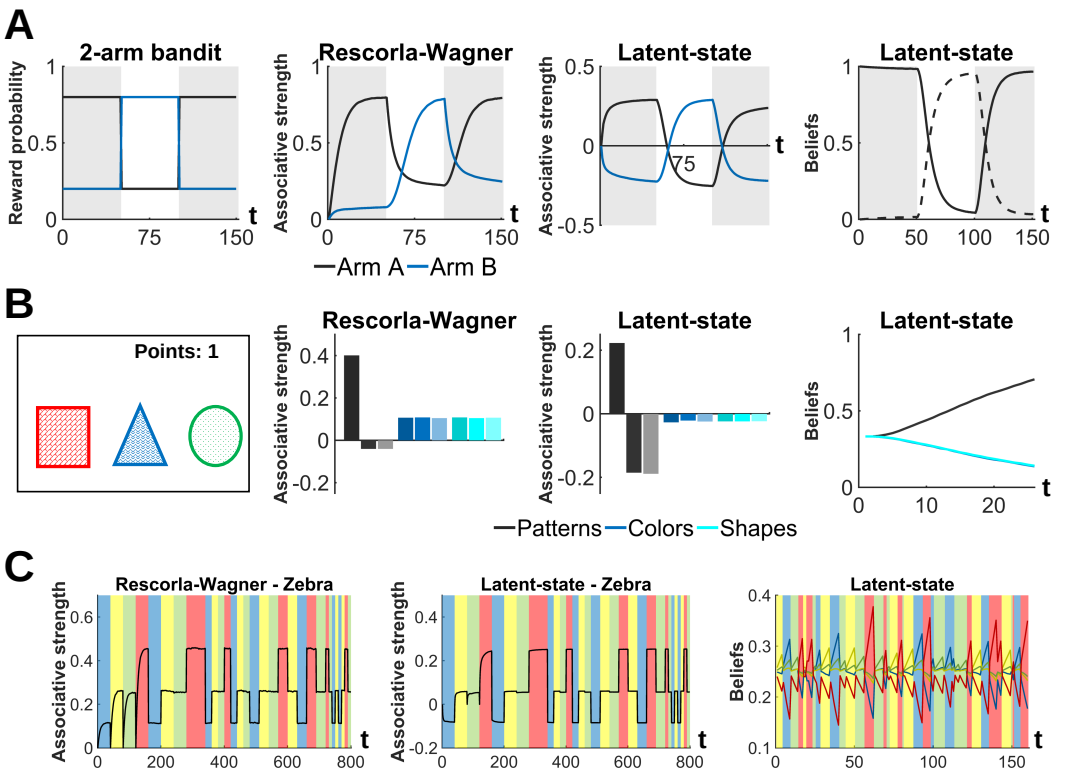


Figure 5. Three experiments testing latent-state learning. **(A)** Associative strength and beliefs were simulated in an experiment in which arm-reward contingencies switch once and then switch back in an experiment. Latent-state learning allows an agent to correctly infer a new latent state after the first switch and recall the original latent state after the second switch. **(B)** Associative strength at experiment end and beliefs were simulated in an experiment that explicitly asked to participant to correctly identify one of three competing hypotheses: patterns predict rewards, colors predict rewards, or shapes predict rewards. A latent-state learner infers the correct hypothesis and associative strengths that accurately reflect predictability of features. **(C)** Associative strength (only one animal is shown) and beliefs were simulated in an experiment that requires a participant to maintain beliefs over competing hypotheses. A latent-state learner can maintain beliefs over multiple hypotheses.

are relevant, or shapes are relevant (Figure 5). For comparison, we used an RW model to model associative strength for the 9 possible features.

The target feature correctly acquires a high associative strength in the RW model, but features in non-relevant dimensions also acquire associative strength even though they do not predict rewards. To compensate for associative strengths of non-relevant features, features in the relevant dimension other than the target feature do not acquire associative strength even though they predict that rewards are delivered at a lower probability. In the latent-state model, the target feature also correctly acquires a high associative strength, but features in non-relevant dimensions acquire no associative strength. Features in the relevant dimension other than the target feature acquire low associative strength correctly signaling these features predict that rewards are delivered at the lower probability. If features acquire associative strength in order to predict rewards, then the latent-state model is better able to uncover how rewards are actually generated by the experiment compared to a RW model.

Maintaining beliefs over multiple hypotheses

Latent-state learning can model learning in situations not captured by a RW model. One example is when an agent must explicitly maintain beliefs over competing hypotheses, which was tested in the two-day learning task of (***Chan et al., 2016***). The task consists of two parts: a *tour* and *photograph*

part. A tour involved a participant visiting four sectors, each associated with its own context (e.g., color, music, image) and with its own probability distribution of observing each of five animals in the sector. While in a sector, animals were presented according to probabilities associated with the sector. The participant learned associations between animals and sectors. During the photograph portion of the task, participants were asked to infer the sector based on an observed sequence of animal photographs.

The RW model can capture participant behavior during the tours portion of the task, but cannot capture participant behavior (relative beliefs in a sector) during the photograph portion of the task (Figure 5). Our latent-state model can capture both. During the tour portion of the task, both the RW model and latent-state learning correctly learn associative strengths (i.e., probabilities in this task) of animals in each sector. During the photographs portion, our latent-state model captures how beliefs in each sector changes as animals are observed. Thus with one model, we capture learning in both portions of this complicated experiment.

Case study

In this section, we use our latent-state model to describe learning and decision-making behavior using data from two experiments with human participants. Our goals were to examine how well our latent-state model explains behavior compared to the RW model. We centered rewards around zero for the RW model to specifically investigate whether latent-state inferences contributed to better model fits rather than centering. Both experiments involved bandit tasks whereby the participant chose between mutually-exclusive options or “arms" based on information provided. Parameters were fit to maximize the likelihood of arm choice assuming trials were independent (see Appendix for details).

Reward-threat conflict

This experiment was designed to examine a participant’s ability to maintain a flexible strategy for obtaining rewards in the presence of threat. Participants (N=55) performed a 3-arm bandit task of 150 trials. Arm choices led to a reward ranging from -10 to 10 and a threat consisting of a distressing or non-distressing image. The participant was instructed to obtain as many points as possible. Probabilities of reward and threat switched in the experiment with the arm that led to the most points also leading to the most threatening images for half of the experiment (Figure 6**A**). The best strategy required choosing arms with a high probability of a distressing image, resulting in potential conflict for a participant.

To model participant behavior, reward and threat outcomes were shifted and scaled to range between -1/2 to 1/2 and then combined into a one-dimensional outcome by taking a weighted average of rewards and threats. For the RW model, three cues were used to encode which arm was chosen. Three parameters were fit to participant behavior: learning rate, relative weighting of threat outcomes versus reward outcomes, and a softmax parameter. By contrast, our latent-state model described participant behavior as weighing three competing hypotheses about which of the three arms should be chosen. The same three cues were used for each hypothesis, except that associative strengths were initialized to favor the arm associated with each hypothesis. Four parameters were fit to participant behavior: learning rate, relative weighting of threat outcomes versus reward outcomes, a softmax parameter, and a latent-state transitions parameter γ. Hence, there is one additional parameter compared to the RW model.

The latent-state model fit participant behavior better than the RW model, with an average improvement of 6.1 (95% CI: 3.1–9.1) in the log-likelihood (Figure 6**B**). Better model fits occurred primarily in the latter part of the experiment, which coincide with switches in reward/threat contingencies. Participants might thus be using latent-state inferences to maintain flexible strategies across the experiment. Each trial, the latent-state model was better able to predict the probability of participant choices by about 0.022 (95% CI: 0.011–0.034) compared to the RW model. This probability estimate was computed as the geometric mean likelihood per trial (i.e. the likelihood raised to

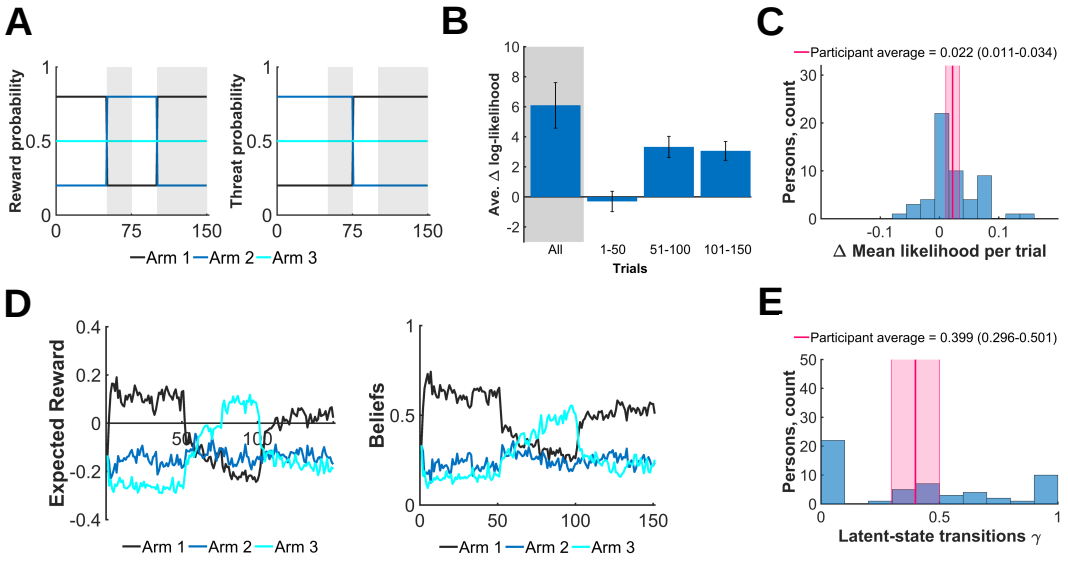


Figure 6. Participant behavior in a reward-threat conflict task. **(A)** Probability of reward and threat for each arm throughout the task. Half the experiment, maximizing reward conflicts with avoiding threat (gray regions). **(B)** Average improvement of latent-state model over a RW model over different sections of the task. **(C)** Improvement by participant of latent-state model over a RW model, as measured by the geometric mean likelihood per trial. **(D)** Expected rewards and beliefs in different arms as predicted by the latent-state model. **(D)** Parameter estimate of latent-state transitions parameter associated with a latent-state model.

the power 1 over the number of trials) averaged across participants. Beliefs across latent-states show participants switch from favoring arm 1 to favoring arm 3 and then back to arm 1 (Figure 6**C**). Compared to the RW model, the latent-state model has one additional parameter (γ) controlling the degree to a participant allows for transitions between latent states (Figure 6**D**). This parameter ranged greatly across participants.

Social learning

This experiment was designed to examine a participant's ability to learn and make decisions while navigating social cues. Participants (N=24) performed a 2-arm bandit task consisting of 72 trials. Arm choices led to a reward ranging from -10 to 10. Prior to each arm choice, participants were provided suggestions on which arm to choose from a caregiver and from a confederate. They were told the confederate is another individual of similar age/gender. In reality, the confederate was simulated and provided randomly generated suggestions. The participant was instructed to obtain as many points as possible. Probabilities of reward and confederate arm choices switched during the experiment, with the confederate providing better suggestions at the start and end of the experiment (Figure 6**A**).

To model participant behavior, we used six cues for the RW model: 2 for arm choice, 2 for agreement or disagreement of arm choice with caregiver, and 2 for agreement or disagreement of arm choice with confederate. Initial associative strengths for two cues (agreeing with caregiver and confederate) were fit to participant behavior in addition to the learning rate and softmax parameter. Varying initial associative strengths captured relative preference at the start of the experiment for making decisions based on internal expectations, caregiver suggestions, or confederate suggestions.

By contrast, our latent-state model described participant behavior as weighing three competing hypotheses: whether to make decisions based on internal suggestions, caregiver suggestion, or confederate suggestion. The same six cues from the RW model were incorporated, but divided among the three hypotheses. The first hypothesis has 2 cues for arm choice; the second hypothesis has 2 for agreement/disagreement with caregiver; and the third hypothesis has 2 cues for agreement/disagreement with confederate. Similar to the RW model, we allow initial associative strength

for following the caregiver and confederate to be fit to behavior in addition to the learning rate, soft-max parameter, and latent-state transitions parameter γ. As before, there is one additional parameter compared to the RW model.

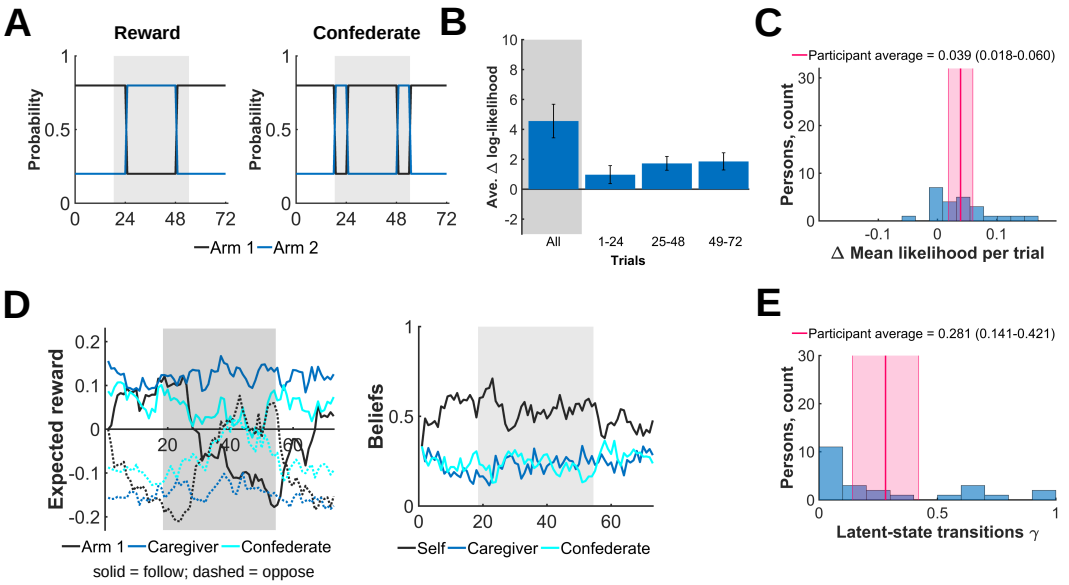


Figure 7. Comparison between RW and latent-state models for describing behavior in social learning task. **(A)** Probability of rewards and confederate suggestion for each arm throughout the task. The confederate often provided bad suggestions (gray regions). **(B)** Average improvement of latent-state model over a RW model over different sections of the task. **(C)** Improvement by participant of latent-state model over a RW model, as measured by the geometric mean likelihood per trial. **(D)** Expected rewards and beliefs in different arms as predicted by the latent-state model. **(D)** Parameter estimate of latent-state transitions parameter associated with a latent-state model.

The latent-state model fit participant behavior better than the RW model, with an average improvement of 4.6 (95% CI: 2.2–6.9) in the log-likelihood (Figure 7**B**). Improvement over the RW model occurred in each third of the experiment. The latent-state model was better able to predict the probability of participant choices by about 0.039 (95% CI: 0.018–0.060) each trial compared to the RW model (Figure 7**C**). Participants might be using latent-state inferences to weigh whether they make decisions based on internal, caregiver, or confederate suggestions.

Expected reward for choosing arms reflects the changes in reward-arm contingencies. Expected reward for following the caregiver stays high for most of the task, whereas expected reward for following the confederate fluctuates throughout the task (Figure 6**D**). The first latent state (self) is associated with higher beliefs averaged over individuals compared to the next two latent states (caregiver and confederate suggestions). The additional parameter in the latent-state model (γ) also varied greatly across participants (Figure 6**E**)

Discussion

We presented a computational model of latent-state learning, using the Rescorla-Wagner (RW) model as its foundation. In a series of simulation experiments, we tested the model's ability to reproduce learning experiments both famously explained and not explained by the RW model and to capture behavior in recent experiments designed to test latent-state inferences. We then used novel datasets (N=55 and N=24) of two learning and decision-making tasks to determine to what extent our latent-state model better explains participant behavior than the RW model. Lastly, we formally derived our latent-state model under the premise of computational rationality, i.e. a learning agent wants to predict outcomes efficiently.

Our goal was to identify a model of latent-state learning that imposes realistic demands on human computation and combines key features of other learning models into a single Bayesian framework. To our knowledge, the resulting model is the first to combine online updates, unlimited latent-states and transitions between latent-states, and a distribution over latent-state beliefs (***Angela and Dayan, 2005***; ***Rescorla, 2000***; ***Redish et al., 2007***; ***Niv et al., 2015***; ***Wilson and Niv, 2012***; ***Chan et al., 2016***). Including these features helps to ensure that the proposed model could examine learning in numerous and diverse experiments, from traditional associative learning experiments to more recent experiments on latent-state inferences. We show that most fall-out naturally when trying to develop an online approach to reward prediction which uses latent-state inferences.

Online updates require only current observations rather than the entire history of observations to update parameters and quantities such as associative strength. Consequently, online approaches are relatively efficient in terms of computation and memory, which could be a more realistic reflection of how humans learn (***Gershman et al., 2015a***). For comparison, an optimal Bayesian approach to latent-state learning would require enumerating over all possible sequences of latent-states, a number that grows exponentially with each trial, imposing a computational and memory burden that is believed to be overly optimistic for human computation (***Angela and Dayan, 2005***). Non-online models of learning overcome their burden with approximate methods such as particle methods (***Gershman and Niv, 2012***; ***Gershman et al., 2010***) and assumptions such as independent trials (***Gershman and Niv, 2012***), one latent state (***Niv et al., 2015***), or one latent-state transition (***Wilson and Niv, 2012***). Assumptions, however, limit the ability of any model to describe learning in varied experiments.

Critically, our latent-state model can capture recall and other learning phenomena that require a more general number of latent states and transitions because of our use of an approximate Bayesian filter (***Cappé, 2011***). An approximate Bayesian filter allows one to maintain and update beliefs over latent-states online. Other online approaches have been proposed for latent-state learning, but often do not include beliefs over latent-states (cf. ***Redish et al.*** (***2007***)). The experiment in ***Chan et al.*** (***2016***), however, required participants to maintain beliefs over multiple latent-states and demonstrated a link between log-beliefs and activity in the orbitofrontal cortex.

Interestingly, our model still updates associative strength in a similar manner to a RW model: change in associative strength is prediction error scaled by associability. In fact, the RW model is a specific case of our latent-state model when there is only one latent state and associability is fixed. For this reason, our latent-state model can reproduce similar learning effects as the RW model such as blocking, overexpectation, and conditioned inhibition. Many latent-state models cannot capture all of these classic learning features (***Gershman and Niv, 2012***). Further, our latent-state model better explained participant behavior (average increase in log-likelihood: 6.1 [95% CI: 3.1–9.1] in the reward-threat task and 4.6 [95% CI: 2.2–6.9] in the social learning task).

Unlike the RW model, our latent-state model allows associability to depend on the history of observations, which is thought to be important for describing learning effects not explained by the RW model, such as the Pearce-Hall effect (***Le Pelley, 2004***; ***Pearce and Hall, 1980***). Associability in our latent-state model involves two components: effort matrices and beliefs in latent-states. Because associability depends on effort matrices, changes in associative strength are rotated into a direction which has not been previously learned, helping to explain backwards blocking. These matrices approximate Fisher information matrices, which are often incorporated into iterative approaches to maximum likelihood estimation (***Cappé and Moulines, 2009***; ***Cappé, 2011***). Meanwhile, because associability also depends on beliefs, associative strength can be updated quickly when a learning agent infers a sudden shift in experimental conditions. This feature allowed the latent-state model to capture experimental results from Experiment 1 in (***Wilson and Niv, 2012***) and Experiment 1B (***Rescorla, 2000***).

The presented model also makes unique testable predictions about humans conduct latent-state learning. For example, associability for a latent-state is predicted to be directly dependent on the

beliefs in that latent-state. Stronger beliefs lead to faster learning, whereas weaker beliefs lead to slower learning. An experiment could be designed to test, for example, if associability decreases by 50% when beliefs are split between two latent-states relative to when beliefs were concentrated on one latent-state. Further, associability is predicted to depend on whether cues had previously been presented together. New combination of cues can lead to quick learning even when individual cues making up the combination have already been presented. Last, latent-state beliefs are predicted to be updated in a non-optimal manner, suggesting one try to induce sub-optimal behavior in an experiment.

Several limitations of the proposed model should be considered. We anticipate there are learning effects that our model fails to describe or there are other models that could explain participant behavior. In addition, we did not examine to what extent beliefs in latent-states and other variables from our model are encoded in various brain regions. We also do not allow beliefs to update randomly, which might lead to better explanations of participant data. Finally, we did not study model performance in terms of predicting rewards as a consequence of using an approximate Bayesian filter. It may useful to know how far from optimal our approximate method is for maximum likelihood estimation in various settings.

In sum, this work establishes the validity of an online model for latent-state inferences that generalizes across experiments. Establishing such a framework is a necessary step towards quantifying normative and pathological ranges of latent-state inferences (and mediating neurocircuitry) across varied contexts. Overall, this work moves us closer to a precise and mechanistic understanding of how humans infer latent-states.

References

Angela JY, Dayan P. Uncertainty, neuromodulation, and attention. Neuron. 2005; 46(4):681–692.

Bouton ME. Context and behavioral processes in extinction. Learning & memory. 2004; 11(5):485–494.

Cappé O. Online EM algorithm for hidden Markov models. Journal of Computational and Graphical Statistics. 2011; 20(3):728–749.

Cappé O, Moulines E. On-line expectation–maximization algorithm for latent data models. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2009; 71(3):593–613.

Chan SC, Niv Y, Norman KA. A probability distribution over latent causes, in the orbitofrontal cortex. Journal of Neuroscience. 2016; 36(30):7817–7828.

Daw ND, O'doherty JP, Dayan P, Seymour B, Dolan RJ. Cortical substrates for exploratory decisions in humans. Nature. 2006; 441(7095):876.

Dayan P, Berridge KC. Model-based and model-free Pavlovian reward learning: revaluation, revision, and revelation. Cognitive, Affective, & Behavioral Neuroscience. 2014; 14(2):473–492.

Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society Series B (methodological). 1977; p. 1–38.

Doll BB, Simon DA, Daw ND. The ubiquity of model-based reinforcement learning. Current opinion in neurobiology. 2012; 22(6):1075–1081.

Farrell S, Lewandowsky S. Computational models as aids to better reasoning in psychology. Current Directions in Psychological Science. 2010; 19(5):329–335.

Gershman SJ, Blei DM, Niv Y. Context, learning, and extinction. Psychological review. 2010; 117(1):197.

Gershman SJ, Hartley CA. Individual differences in learning predict the return of fear. Learning & behavior. 2015; 43(3):243–250.

Gershman SJ, Horvitz EJ, Tenenbaum JB. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. Science. 2015; 349(6245):273–278.

Gershman SJ, Niv Y. Exploring a latent cause theory of classical conditioning. Learning & behavior. 2012; 40(3):255–268.

Gershman SJ, Norman KA, Niv Y. Discovering latent causes in reinforcement learning. Current Opinion in Behavioral Sciences. 2015; 5:43–50.

Gershman SJ, Jones CE, Norman KA, Monfils MH, Niv Y. Gradual extinction prevents the return of fear: implications for the discovery of state. Frontiers in behavioral neuroscience. 2013; 7:164.

Gläscher J, Daw N, Dayan P, O'Doherty JP. States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. Neuron. 2010; 66(4):585–595.

Granjon P. The CuSum algorithm-a small review. . 2013; .

Hall G, Pearce JM. Latent inhibition of a CS during CS–US pairings. Journal of Experimental Psychology: Animal Behavior Processes. 1979; 5(1):31.

Huys QJ, Guitart-Masip M, Dolan RJ, Dayan P. Decision-theoretic psychiatry. Clinical Psychological Science. 2015; 3(3):400–421.

Le Pelley ME. The role of associative history in models of associative learning: A selective review and a hybrid model. The Quarterly Journal of Experimental Psychology Section B. 2004; 57(3b):193–243.

Li J, Schiller D, Schoenbaum G, Phelps EA, Daw ND. Differential roles of human striatum and amygdala in associative learning. Nature neuroscience. 2011; 14(10):1250.

Mackintosh NJ. A theory of attention: variations in the associability of stimuli with reinforcement. Psychological review. 1975; 82(4):276.

Niv Y, Daniel R, Geana A, Gershman SJ, Leong YC, Radulescu A, Wilson RC. Reinforcement learning in multidimensional environments relies on attention mechanisms. Journal of Neuroscience. 2015; 35(21):8145–8157.

Page ES. Continuous inspection schemes. Biometrika. 1954; 41(1/2):100–115.

Pearce JM, Hall G. A model for Pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. Psychological review. 1980; 87(6):532.

Redish AD. Addiction as a computational process gone awry. Science. 2004; 306(5703):1944–1947.

Redish AD, Jensen S, Johnson A, Kurth-Nelson Z. Reconciling reinforcement learning models with behavioral extinction and renewal: implications for addiction, relapse, and problem gambling. Psychological review. 2007; 114(3):784.

Rescorla RA. Associative changes in excitors and inhibitors differ when they are conditioned in compound. Journal of Experimental Psychology: Animal Behavior Processes. 2000; 26(4):428.

Rescorla RA, Wagner AR, et al. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. Classical conditioning II: Current research and theory. 1972; 2:64–99.

Robbins H, Monro S. A stochastic approximation method. In: *Herbert Robbins Selected Papers* Springer; 1985.p. 102–109.

Stalnaker TA, Cooch NK, Schoenbaum G. What the orbitofrontal cortex does not do. Nature neuroscience. 2015; 18(5):620.

Sutton RS, Barto AG, et al. Reinforcement learning: An introduction. MIT press; 1998.

Tolman EC. Cognitive maps in rats and men. Psychological review. 1948; 55(4):189.

Vervliet B, Craske MG, Hermans D. Fear extinction and relapse: state of the art. Annual review of clinical psychology. 2013; 9:215–248.

Wikenheiser AM, Schoenbaum G. Over the river, through the woods: cognitive maps in the hippocampus and orbitofrontal cortex. Nature Reviews Neuroscience. 2016; 17(8):513.

Wilson PN, Boumphrey P, Pearce JM. Restoration of the orienting response to a light by a change in its predictive accuracy. The Quarterly Journal of Experimental Psychology Section B. 1992; 44(1b):17–36.

Wilson RC, Niv Y. Inferring relevance in a changing world. Frontiers in human neuroscience. 2012; 5:189.

Mathematical justification

We justify our choice of model under the premise of *computational rationality* (***Gershman et al., 2015a***), i.e. a learning agent wants to predict rewards relatively efficiently. An optimistic strategy for the learning agent would be to use maximize likelihood estimation (MLE) to aid prediction. This would entail starting with a probabilistic model of rewards defined up to an unknown parameter θ and finding θ to maximize the log-likelihood of rewards (scaled by $1/t$):

$$\ell(\theta) = \frac{1}{t} \log f\left[R(1{:}t) \mid \theta\right]. \tag{9}$$

Treating rewards as a continuous variable, we use f to denote a probability density function. The estimate of θ together with the reward model predict future rewards. With an eye towards efficiency, we focus on *online* approaches, such as the Rescorla-Wagner model, and refer the reader to the work in ***Cappé and Moulines*** (***2009***) and ***Cappé*** (***2011***) on updating latent-variable models online.

Rescorla-Wagner model as an online approach to maximum likelihood estimation

Before we motivate our latent-state model, we explore how one could perform MLE online to predict rewards without latent-states. In the process, we show that the RW model is an online approach to MLE and illuminate a pathway for performing MLE online with latent-states. The most natural modeling choice for rewards is arguably a standard Gaussian regression model, ubiquitous in statistical inference:

$$R(t) = \vec{c}(t)'\vec{\mu} + \epsilon(t)$$

where $\vec{c}(t)$ is the cue vector on trial t (independent variables); $\vec{\mu}$ are unknown parameters (unknown regression coefficients); and $\epsilon(t)$ is an independent Gaussian random variable (error) with mean zero and unknown variance v. Unknown parameters are collected in $\theta = \begin{bmatrix}\vec{\mu}' & v\end{bmatrix}'$.

Although we could maximize ℓ directly, we want an online approach to maximization and instead consider applying Newton's method for optimization (i.e., gradient descent), which involves repeatedly updating an estimate $\hat{\theta}$ of θ with a new estimate of the form:

$$\hat{\theta} - \lambda(t) H(t)^{-1} \nabla_\theta \ell(\theta)\Big|_{\theta=\hat{\theta}}$$

for some learning rate $\lambda(t)$ and an appropriate matrix $H(t)$. Matrix $H(t)$ can take many forms and is important for determining how quickly estimates converge to a maximum/minimum. An ideal choice is the Hessian resulting in the Newton-Raphson method, but the Hessian is often noninvertible in MLE. A common alternative is the negative of the Fisher information matrix $I(\theta)$ scaled by $1/t$ and evaluated at the current estimate (***Cappé and Moulines, 2009***). Even with Newton's method, rewards still need to be stored up to trial t, but Newton's method can be turned into an online algorithm using the well-known Robbins-Monro method (***Robbins and Monro, 1985***). We simply replace the gradient with an (stochastic) approximation that depends only on the current trial:

$$\nabla_\theta \ell(\theta) \approx \nabla_\theta \log f\left[R(t) \mid \theta\right]$$

Note that each side of the equation is equal in expectation when rewards are independent and identically-distributed.

Upon replacing the gradient with a stochastic approximation, we arrive at the RW model if we use a constant learning rate $\lambda(t) = \lambda_0$ and the matrix

$$H(t) = -\frac{1}{\hat{v}} \begin{bmatrix} I & 0 \\ 0 & \frac{1}{2\hat{v}} \end{bmatrix}.$$

For the Gaussian regression model, matrix $H(t)$ resembles

$$-\frac{1}{t} I(\theta) = \frac{1}{t} \mathbb{E}\left[\nabla_\theta^2 \log f\left[R(1{:}t) \,\middle|\, \theta\right] \,\middle|\, \theta\right] = -\frac{1}{v} \begin{bmatrix} \frac{1}{t}\sum_{s=1}^{t} \vec{c}(s)\vec{c}(s)' & 0 \\ 0 & \frac{1}{2v} \end{bmatrix}.$$

evaluated at $\hat{\theta} = \begin{bmatrix} \vec{\hat{\mu}}' & \hat{v} \end{bmatrix}'$ We simply replaced $\frac{1}{t}\sum_t \vec{c}(t)\vec{c}(t)'$ with the identity matrix I, which ensures $H(t)$ is invertible. Putting the online approach together with the fact that

$$\nabla_\theta \log f\left[R(t) \mid \theta\right] = \frac{1}{v}\begin{bmatrix} \vec{c}(t)(R(t) - \vec{c}(t)'\vec{\mu}) \\ -\frac{1}{2} + \frac{1}{2v}(R(t) - \vec{c}(t)'\vec{\mu})^2 \end{bmatrix}$$

for the Gaussian regression model, we arrive at the following online update for the estimate $\hat{\theta}$:

$$\hat{\theta} + \lambda_0 \begin{bmatrix} \vec{c}(t)(R(t) - \vec{c}(t)\vec{\hat{\mu}}) \\ (R(t) - \vec{c}(t)'\vec{\hat{\mu}})^2 - \hat{v} \end{bmatrix}.$$

Critically, estimates $\vec{\hat{\mu}}$ are updated exactly as in an RW model:

$$\vec{\hat{\mu}} + \lambda_0 \vec{c}(t)(R(t) - \vec{c}(t)\vec{\hat{\mu}}).$$

In other words, the RW model is an online approach to maximum likelihood estimation.

Latent-state learning as an online approach to expectation-maximization

Motivated by the previous argument, we now explore how one could perform MLE online to predict rewards with latent-states. A natural choice for modeling rewards is to use a Markov chain of Gaussian regression models. Latent random variable $X(t) \in \{1, \ldots, L\}$ represents the active hypothesis on trial t. Given the active hypothesis $X(t) = l$, rewards are described by a Gaussian linear regression model:

$$R(t) = \vec{c}_l(t)'\vec{\mu}_l + \epsilon(t)$$

where $\vec{\mu}_1, \ldots, \vec{\mu}_L$ are unknown parameters and $\epsilon(t)$ is an independent Gaussian random variable with mean zero and unknown variance v. The extension to latent-state learning thus leads to L different Gaussian regression models for rewards rather than one. As discussed in the Model section, we assume $X(t)$ is a Markov chain that transitions to itself from one trial to the next with probability $1-\gamma$ and transitions to a new state (all new states being equally-likely) with probability γ. Parameter γ and initial probabilities for $X(0)$ are assumed to be known/given, leaving unknown parameters collected in $\theta = \begin{bmatrix} \vec{\mu}_1' & \cdots & \vec{\mu}_L' & v \end{bmatrix}'$.

In a latent-variable setting, a popular algorithm to perform MLE is expectation-maximization (EM) (***Dempster et al., 1977***). An iterative algorithm, EM improves upon a current estimate $\hat{\theta}$ at each iteration in two steps. Fixing the current trial number t, an *expectation* step uses a current estimate $\hat{\theta}$ to calculate the expectation of the complete log-likelihood of rewards and latent states given rewards and scaled by $1/t$:

$$\begin{aligned} Q(\theta|\hat{\theta}) =& \frac{1}{t}\mathbb{E}\left[\log f\left[R(1{:}t), X(1{:}t) \,\middle|\, \theta\right] \middle| R(1{:}t), \hat{\theta}\right] \\ =& \frac{1}{t}\sum_{s=1}^{t}\mathbb{E}\left[\log f\left[R(s) \,\middle|\, X(s), \theta\right] + \log \mathbb{P}\left[X(s) \,\middle|\, X(s-1)\right] \middle| R(1{:}t), \hat{\theta}\right] \end{aligned}$$

A *maximization* step searches for a maximum of $Q(\theta|\hat{\theta})$ over θ. The maximum is then used as the estimate in the next iteration. The crux of the EM method is that $Q(\theta|\hat{\theta})$ is easier to maximize than the log-likelihood $\ell(\theta)$ of the observed data and that improvement in $Q(\theta|\hat{\theta})$ guarantees improvement in $\ell(\theta)$.

Mirroring our argument for the Rescorla-Wagner model, we can replace the maximization step with a Newton update:

$$\hat{\theta} - \lambda(t)H(t)^{-1}\nabla_\theta Q(\theta|\hat{\theta})\Big|_{\theta=\hat{\theta}}$$

for appropriate learning rate $\lambda(t)$ and matrix $H(t)$. Even in a latent-variable setting, one choice for $H(t)$ is still the negative of the Fisher's information matrix scaled be $1/t$ and evaluated at $\hat{\theta}$, but

Newton's method still requires all rewards up to trial t. To recover an online approach, we use a stochastic approximation to $\nabla_\theta Q(\theta|\hat{\theta}(t))$ in the maximization step:

$$\begin{aligned}\nabla_\theta Q(\theta|\hat{\theta}(t)) \approx\ & \nabla_\theta \mathbb{E}\left[\log f\left[R(s)\,\middle|\,X(s),\theta\right] + \log \mathbb{P}\left[X(s)\,\middle|\,X(s-1)\right]\middle|R(1{:}t),\hat{\theta}\right] \\ =\ & \sum_{l=1}^{L} \mathbb{P}\left[X(t)=l|R(1{:}t),\hat{\theta}\right] \nabla_\theta \log f\left[R(s)\,\middle|\,X(s)=l,\theta\right]\end{aligned}$$

We then replace $\mathbb{P}\left[X(t)=l|R(1{:}t),\hat{\theta}\right]$, which is computed using all the rewards, with latent-state beliefs $p_l(t)$ defined at Eq. 5 which is computed online. This replacement yields the stochastic approximation

$$\nabla_\theta Q(\theta|\hat{\theta}(t)) \approx \sum_{l=1}^{L} p_l(t)\nabla_\theta \log f\left[R(s)\,\middle|\,X(s)=l,\theta\right] = \frac{1}{\nu}\begin{bmatrix} p_1(t)\vec{c}_1(t)\left(R(t)-\vec{c}_1(t)'\vec{\mu}_l\right) \\ \vdots \\ p_L(t)\vec{c}_L(t)\left(R(t)-\vec{c}_L(t)'\vec{\mu}_L\right) \\ -\frac{1}{2}+\frac{1}{2\nu}\sum_{l=1}^{L} p_l(t)\left(R(t)-\vec{c}_l(t)'\vec{\mu}_l\right)^2 \end{bmatrix}$$

The last equality follows from our assumption that rewards are a Markov chain of Gaussian regression models.

Upon replacing the gradient in Newton's method with this stochastic approximation, we can arrive at our latent-state learning model if we use $\lambda(t) = 1/t$ and matrix

$$H(t) = -\frac{1}{\hat{\nu}}\begin{bmatrix} \frac{1}{t}B_1(t) & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{t}B_2(t) & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & \frac{1}{t}B_L(t) & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{2\hat{\nu}} \end{bmatrix},$$

where we used the definition of $B_l(t)$ at Eq. 4. For a Markov chain of Gaussian regression models, matrix $H(t)$ resembles the negative of the Fisher information matrix scaled by $1/t$:

$$\begin{aligned}-\frac{1}{t}I(\theta) &= \frac{1}{t}\mathbb{E}\left[\nabla_\theta^2 \log f\left[R(1{:}t), X(1{:}t)\,\middle|\,\theta\right]\middle|\theta\right] \\ &= -\frac{1}{\nu}\begin{bmatrix} \frac{1}{t}\sum_{s=1}^{t}\rho_1(s,\theta)\vec{c}_1(s)\vec{c}_1(s)' & 0 & \cdots & 0 & 0 \\ 0 & \frac{1}{t}\sum_{s=1}^{t}\rho_2(s,\theta)\vec{c}_2(s)\vec{c}_2(s)' & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & \frac{1}{t}\sum_{s=1}^{t}\rho_L(s,\theta)\vec{c}_L(s)\vec{c}_L(s)' & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{2\nu} \end{bmatrix}\end{aligned}$$

where $\rho_l(s,\theta) = \mathbb{P}\left[X(s)=l|\theta\right]$. We simply replaced diagonal blocks with $\frac{1}{t}B_l(t)$ which ensures $H(t)$ is invertible and replaced $\rho_l(s,\theta)$ with latent state beliefs $p_l(s)$. With these choices for $\lambda(t)$ and $H(t)$, we update $\hat{\theta}$ online according to:

$$\begin{aligned} &\hat{\mu}_1 + p_1(t)B_1(t)^{-1}\vec{c}_1(t)\left(R(t)-\vec{c}_1(t)'\vec{\hat{\mu}}_1\right) \\ &\qquad\vdots \\ &\hat{\mu}_L + p_L(t)B_L(t)^{-1}\vec{c}_L(t)\left(R(t)-\vec{c}_L(t)'\vec{\hat{\mu}}_L\right) \\ &\hat{\nu} + \frac{1}{t}\left(\sum_{l=1}^{L} p_l(t)\left(R(t)-\vec{c}_l(t)'\vec{\hat{\mu}}_l\right)^2 - \hat{\nu}\right)\end{aligned}$$

We thus arrive at our latent-state model for updating associative strengths at Eq.(1). We can also recover our latent-state model for updating variance at Eq.(7), if we replace the learning rate $\frac{1}{t}$ with a constant learning rate β_0. This completes our motivation for our latent-state model.

Extensions

There are three extensions of the model to consider. To extend to multidimensional reward setting, we suppose that the learning agent observes D rewards on each trial: $R_1(t), \ldots, R_D(t)$. In this case, we assume the learning agent updates a separate RW model for each learning state and *each reward dimension*. We replace the cue vector, associative strength, prediction error, associability matrix, and effort matrix with corresponding vectors/matrices $\vec{c}_{l,d}(t), \vec{V}_{l,d}(t), E_{l,d}(t), A_{l,d}(t), B_{l,d}(t)$ that depends on each latent state l, trial t, and dimension d. We update these variables as in the one-dimensional setting with $R(t)$ replaced with the corresponding reward $R_d(t)$. Meanwhile, latent-state beliefs $p_l(t)$ remain specific to a latent state and not the reward dimension. Assuming rewards are independent across dimensions, we replace

$$\phi\left(\frac{E_l(t)}{\sigma(t-1)}\right),$$

in Eq. (5)–(6) with

$$\prod_d \phi\left(\frac{E_l(t)}{\sigma(t)}\right).$$

We continue to pool variances $\sigma^2(t)$ into a single quantity and thus replace Eq. (8) with

$$\overline{E^2}(t) = \frac{1}{D}\sum_{d=1}^{D}\sum_{l=1}^{L} p_l(t)E_{l,d}^2.$$

Many learning tasks require an agent to decide between M mutually-exclusive options on each trial with rewards generated depending on the agent's decision. To apply our model to such a decision-making environment, we define $\vec{c}_{l,d,m}(t)$ to be the feature vector *if* the agent were to choose option m. Then based on the agent's current model on trial t, expected reward along dimension d for choosing option m would be

$$W_{d,m}(t) = \sum_{l=1}^{L} p_l(t)\vec{c}_{l,d,m}(t)' V_{l,d}(t)$$

The agent then chooses option m on trial t with probability

$$\frac{\exp \sum_{d=1}^{D} \tau_d W_{d,m}(t)}{\exp\left(\sum_{d=1}^{D}\sum_{n=1}^{M} \tau_d W_{d,n}(t)\right)}$$

for parameter τ_d determining how much an agent explores versus exploits and weights rewards along each dimension. Note that since the expected reward per option $W_{d,m}(t)$ is available, rules other than the above soft-max rule, e.g. ϵ-greedy rules, could be used to model how an agent makes a decision.

Lastly, many learning-state learning models do not assume the number of latent states *a priori* (***Gershman and Niv, 2012***; ***Gershman et al., 2010***). Instead, latent states are included *ad infinum* if needed to better explain rewards. Incorporating this feature is not as simple as letting $L = \infty$ (or at least let L be very large) for two reasons. First, the learning agent would need to keep track of, and update variables for, a large number of latent states, which is clearly inefficient. Second, because our approach is online, any latent states with identical (or near-identical) histories would be updated in an identical (or near-identical) manner leading to redundant states, which is also inefficient. Non-online approaches remove this redundancy by optimizing over latent states. Rather than specifying a large number of latent states *a priori*, we can extend our model to incorporate an *online* mechanism to trigger an addition of new latent states when needed to explain rewards. This mechanism could reflect the need to capture *unexpected uncertainty*, as discussed in Yu an Dayan (***Angela and Dayan, 2005***).

We present one possible mechanism based on Page's algorithm (***Page, 1954***) for solving a problem known as change point detection (cf. ***Granjon*** (***2013***)). The idea is to test whether we

can reject our current model in favor of an alternative model with new latent states. Specifically, we let $L = L(t)$ be the number of latent states actively considered on trial t and suppose that a learning agent is initially considering L_0 latent states, with cue vectors defined for each latent state. (Note: $L(0)$ is typically a multiple of L_0 to allow for initial copies of L_0 to predict rewards later in the experiment when needed.) At the start of the experiment, we keep track of model performance using a statistic $q(t)$:

$$q(t) = \max\left(q(t-1) + \log\left(\frac{l_1}{l_0}\right) - \delta, 0\right) \tag{10}$$

with a scalar parameter δ and $q(0)$ initialized to zero. Importantly, l_1/l_0 is the likelihood ratio between the current model with $L(t)$ states to a *null* model with L_0 states and zero associative strengths, where l_0 was the normalizing constant for beliefs defined in Eq. (6) and extended for multidimensional rewards and

$$l_1 = \frac{1}{L_0}\prod_{d=1}^{D}\phi\left(\frac{R_d(t) - 0}{\sigma(t-1)}\right) \tag{11}$$

If $q(t)$ exceeds a threshold η, then we reset $q(t)$ to zero and add L_0 states using the same cue vectors associated with the original L_0 states. Adding unlimited states yields Algorithm 2. Extensions to experiments with multidimensional rewards and decision-making can be made to this algorithm, as was done for Algorithm 1.

Input: Initial number L_0 of latent states of interest, initial number L of latent states, initial beliefs $p_l(0)$ and variance $\sigma^2(0)$, cue vectors $\vec{c}_l(t)$;
/* Initialize states */
for *each latent state* $l = 1, \ldots, L$ **do**
  Initialize associative strength $\vec{V}_l(0) = 0$;
  Initialize effort matrix $\vec{B}_l(0) = I$;
end
/* Loop through trials */
for *each trial* $t = 1, \ldots, T$ **do**
  /* Check if new states are needed */
  if *Add new states based on* (10) **then**
    /* Initialize new states */
    for *each new latent state* $l = 1, \ldots, L_0$ **do**
      Initialize associative strength $\vec{V}_{L+l}(t-1) = 0$;
      Initialize effort matrix $\vec{B}_{L+l}(t-1) = I$;
      Initialize latent-state belief $\vec{p}_{L+l}(t-1) = 0$;
      Reset model fit statistic $q(t) = 0$;
    end
    Update number of latent states $L \leftarrow L + L_0$;
  end
  /* Update states */
  for *each latent state* $l = 1, \ldots, L$ **do**
    Update latent-state belief $p_l(t)$ with Eq. (5)–(6);
    Update effort matrix $\vec{B}_l(t)$ with Eq. (4);
    Update associative strength $\vec{V}_l(t)$ with Eq. (1)–(3);
  end
  Update variance $\sigma^2(t)$ according to Eq. (7)–(8);
end

Algorithm 2: Model of latent-state learning with online updates and *unlimited* latent states. Additions to Algorithm 1 are written in red.

Simulation details

For most experiments, we represented cue presentation on trial t as vectors $\vec{c}_1(t) = \ldots = \vec{c}_L(t)$ such that each entry in the vector corresponded to a particular cue (e.g., cue A) and took a value of 1 if the cue was present and 0 otherwise. Unless otherwise specified, we also assumed two latent states $L = 2$ with initial beliefs $p_1(0) = 1$ and $p_2(0) = 0$ to capture one latent state for initial predictions and an additional latent state for later predictions if needed to explain rewards. For all decision-making tasks, we used a soft-max function to define how a learning agent makes decisions randomly based on current expectations, and for tasks involving randomly generating rewards, we simulated the task for 5000 runs and report learning behavior averaged over simulation runs. Except for our last example (animal safari task from ***Chan et al.*** (***2016***)), we encoded binary rewards as -1/2 or 1/2 for our latent-state model, and so, associative strengths are often shifted by 1/2 for our model when compared to other models. Our last example encoded entries in the reward vector as either -1/5 or 4/5 for our latent-state model to reflect five mutually-exclusive reward outcomes rather than two.

Blocking, overexpectation, and conditioned inhibition

For these experiments, we followed conditioning schedules in ***Le Pelley*** (***2004***). Blocking simulations consisted of 20 trials of reinforcing Cue A, followed by 16 trials alternating between reinforcing a compound of Cue A and Cue B and reinforcing a compound of Cue C and Cue D. Overexpectation simulations consisted of 100 trials alternating between reinforcing Cue A and reinforcing Cue B, followed by 10 trials of reinforcing a compound of Cue A and Cue B. Conditioned inhibition simulations consisted of 200 trials alternating between reinforcing Cue A and not reinforcing a compound of Cue A and Cue B.

Backwards blocking

We used a similar conditioning schedule of the blocking experiment: 20 trials of reinforcing Compound AB, followed by 16 trials of reinforcing Cue A.

Experiments 1A and 1B in *Rescorla* (*2000*)

We followed the conditioning schedule in the original experiment (***Rescorla, 2000***). For both experiments, the first stage of training involved presenting the five trial types (A+,C+,X+,BX,DX) in random order for a total of 500 trials (= $[8 \times 12$ days $+ 4 \times 1$ day$] \times 5$ trial types). In Stage 2, experiment 1A followed this conditioning by presenting the reinforced compound AB+ for a total of 8 times, whereas experiment 1B followed this conditioning by presenting the non-reinforced compound AB 8 times. We measured the change in associative strength from the end of Stage 1 to the end of Stage 2.

Experiment 1 in *Wilson et al.* (*1992*).

We followed the conditioning schedule used to simulate this experiment in the latent-state modeling paper of ***Gershman and Niv*** (***2012***). For Group C, this schedule consisted of 50 trials of alternating between reinforcing a light and tone with food and presenting a light and tone without food, followed by 10 trials of reinforcing the light with food.

2-arm bandit with recall

We simulated a 2-arm bandit task in which the probability of receiving a reward for each arm switched twice through the experiment. For trials 1-50 and trials 101-151, Arm A and Arm B received a reward with probability 0.80 and 0.20, respectively. For trials 51-100, these probabilities were switched with Arm A and Arm B receiving a reward with probability 0.20 and 0.80, respectively. Cue vectors encoded arm choice (e.g., $\vec{c}_{l,1,1}(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}'$ if Arm A is chosen and $\vec{c}_{l,1,2}(t) = \begin{bmatrix} 0 & 1 \end{bmatrix}'$ if Arm B is chosen).

3-arm bandit game from *Niv et al.* (*2015*)

In the original experiment, each stage lasted anywhere from 15 to 25 trials. Because rewards were randomly generated, we fixed the number of trials to 25 in the simulation. We also fixed the target feature. We averaged associative strengths and beliefs over simulated runs. Note, the number of latent states could also be chosen to match the 9 possible target feature dimensions as in *Niv et al.* (*2015*), but we opted for one latent state per feature dimension to reflect that information on a feature provides information on other features in the same dimension. Three hypotheses also agrees with a model of the typical N-arm task in that a single latent state corresponds to each feature dimension (i.e., arm). Cue vectors encoded which feature along the relevant feature dimension for the given latent state was present on each trial (e.g., $\vec{c}_{2,1,1}(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}'$ encoded that Arm A was colored green, since latent state 2 represented the belief in colors and green was the first color along the color dimension).

Animal safari task from *Chan et al.* (*2016*)

We followed the schedule as reported in *Chan et al.* (*2016*). On day 1, the participant tours each sector four times consisting of 40 trials in the first two tours and 20 trials in the second two tours. On day 2, the participant tours each sector four times consisting of 30 trials in the first two tours and 10 trials in the second two tours. Prior to each tour, the relevant sector is shown on a map and sector-specific cues (e.g., colors) are associated with each animal picture. Sector order is clockwise around the map on the first two sets of sector tours and then chosen uniformly at random for the last six sets of sector tours. For each tour, animals were drawn at random with probability depending on which sector is currently being toured. Probabilities were taken from *Chan et al.* (*2016*). We fixed presentation order of sectors to average behavior across simulation runs.

On day 2, the tours were followed by the photograph part of the task. The photograph part consisted of six sessions. The first two sessions consisted of 20 trials and the last four sessions consisted of 40 trials. In each session, the participant is shown a sequence of animals from one of the sectors. Animals in each sequence are drawn according to the probabilities associated with one of the sectors, with sequence length varying from 1 to 6 pictures. At the end of sequence, the participant is asked which of two sectors were more likely. Theses two sectors are chosen uniformly from the four possible sectors and do not necessarily include the sector from which the animals were actually drawn. To identify the correct sector, the participant must thus infer the right context (i.e., sector) only from observations of animal frequencies. Sequence length is chosen uniformly at random (but then truncated if length would lead to more trials than are designed per session). Sectors are also chosen uniformly at random, as well as which of the two sectors are presented at the prompt. We fixed presentation order of sectors and sequence lengths to average behavior across each simulation run.

To model the “tour" portion of the task with our latent-state model, we start with two latent states ($L = 2$) as we did with blocking and other experiments: one to build initial predictions and a second latent state if needed. We assumed five outcomes per trial ($D = 5$) to encode which of the five animals is observed. By allowing for five outcomes, we presume a participant correctly uses both the presence *and* absence of an animal to update expectations for each animal in a sector. For each reward dimension and latent-state, the cue vector is defined to be a 4-dimensional vector with each entry encoding whether or not a participant is each of the four sectors. We hence collapsed all contextual information (e.g., colors, sounds) into a single value to represent whether a participant is in the sector. For the “photograph" portion, we start with eight latent states ($L = 8$): four to represent initial predictions for the four unknown sectors, i.e. the four competing “hypotheses" about which sector the participant is currently located, and an additional four copies of these latent states to represent later predictions for the four unknown sectors (if needed to explain rewards). Initial beliefs are divided equally among each of the first four latent states, and rewards are defined as in the “tour" portion. With no cues to represent a sector, the cue vector is the scalar 1 for each reward dimension and latent-state. To account for learning on the “tour" portion, we initialize

associative strengths for each latent state to be the final associative strengths acquired for the associated sector during the tour. With each new sequence, we carry associative strengths forward, but reset initial beliefs to be equally divided among the first four latent states.

Simulation comparison to other learning models

We simulated the same experiments in the Simulation section using two additional models. While we do not expect any model (including our latent-state model) to capture every learning effect of interest, our goal was to illustrate that capturing all the learning effects that our latent-state model can capture is not trivial. It is important to note, however, that we fix parameters in these two models and that there may be parameters that could be adjusted to match various learning effects. In what follows, we use similar notation to the Model section with a vector $\vec{V}(t)$ to denote associative strengths on trial t denoted, a vector $\vec{c}(t)$ to denote which cues are presented on trial t, $R(t)$ to denote rewards on trial t, and $E(t)$ to denote prediction error on trial t:

$$E(t) = R(t) - \vec{c}(t)'\vec{V}(t-1).$$

In addition, we use $C(t)$ to denote the diagonal matrix whose diagonal entries make up the cue vector $\vec{c}(t)$. Lastly, we extend these definitions for the case of multidimensional rewards as was discussed in the Model section.

We first considered the Hybrid model in ***Li et al.*** (***2011***). This model updates associative strength in a similar manner to the RW model:

$$\vec{V}(t) = \vec{V}(t-1) + \kappa_H A(t)\vec{c}(t)E(t)$$

where κ_H is a constant learning rate parameter and $A(t)$ is an matrix capturing associability. The Hybrid model, however, allows associability $A(t)$ to update based on prediction errors. Specifically, associability $A(t)$ is a diagonal matrix that is initialized to be the identity matrix and updated according to

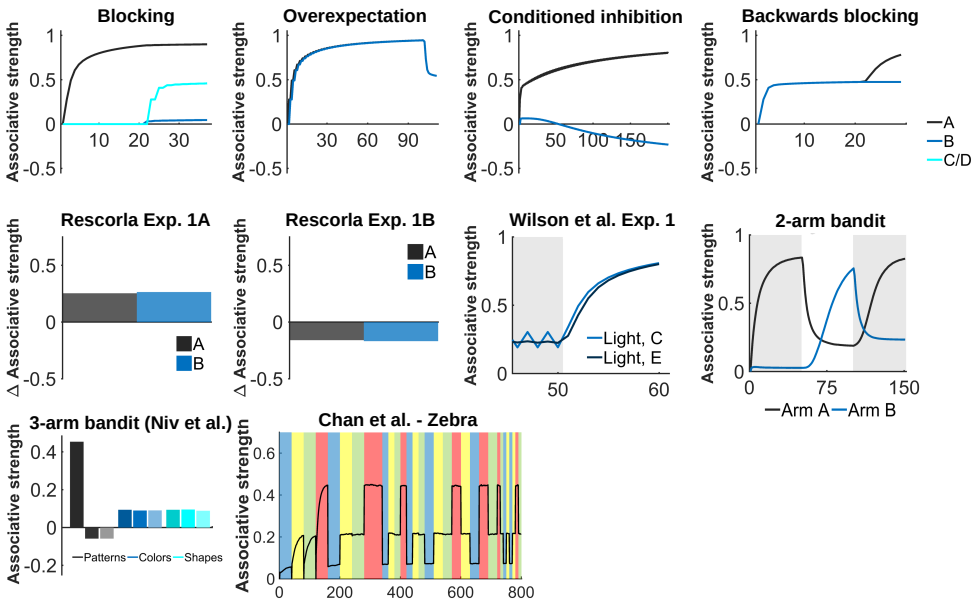
$$A(t) = A(t-1) + \eta_H C(t)\left(A(t-1) - |E_d(t)|C(t)\right)$$

where η_H is a constant parameter. Hence, associability increases when the model poorly predicts rewards, allowing for quicker learning in uncertain environments. In all simulation experiments, we fixed parameters $\eta_H = 0.857$ and $\kappa_H = 0.3$ and initialized each associative strength entry in $\vec{V}_d(0)$ to zero and each associability entry in the diagonal of $A(0)$ to 0.926. Except for κ_H, these parameters are found in the Supplement of ***Li et al.*** (***2011***). The supplement uses a value of 0.166 for κ_H, which we found poorly highlighted the associability process in simulation experiments. We thus increased κ_H. The simulated examples demonstrate the Hybrid model with the given parameters captures some, but not all of the learning effects (Fig 8). For example, the Hybrid captured blocking, overexpectation, and conditioned inhibition, but did not capture backwards blocking or the experiments in ***Rescorla*** (***2000***) and ***Wilson et al.*** (***1992***). The model also does not keep track of beliefs over latent-states, limiting their ability to model latent-state inferences in the photograph portion of the experiment in ***Chan et al.*** (***2016***).

We also considered the feature reinforcement learning with decay (fRL+decay) model in ***Niv et al.*** (***2015***). This model provided the best explanation of behavioral and neural data of the 3-arm bandit task in ***Niv et al.*** (***2015***), one of the experiments simulated in the current paper. The fRL+decay model updates associative strength in a similar manner to the RW model for cues that are present, but otherwise decays associative strength to zero for cues that are absent. These updates can be written concisely as:

$$\vec{V}(t) = \vec{V}(t-1) + \kappa_{fRL}\vec{c}(t)E(t) - \eta_{fRL}(I - C(t))\vec{V}(t-1)$$

for a learning rate parameter κ_{fRL} and a decay rate parameter η_{fRL}. We set parameters $\kappa_{fRL} = 0.15$ and $\eta_{fRL} = 0.45$ to approximately match parameter estimates in Table 1 of ***Niv et al.*** (***2015***). Similar



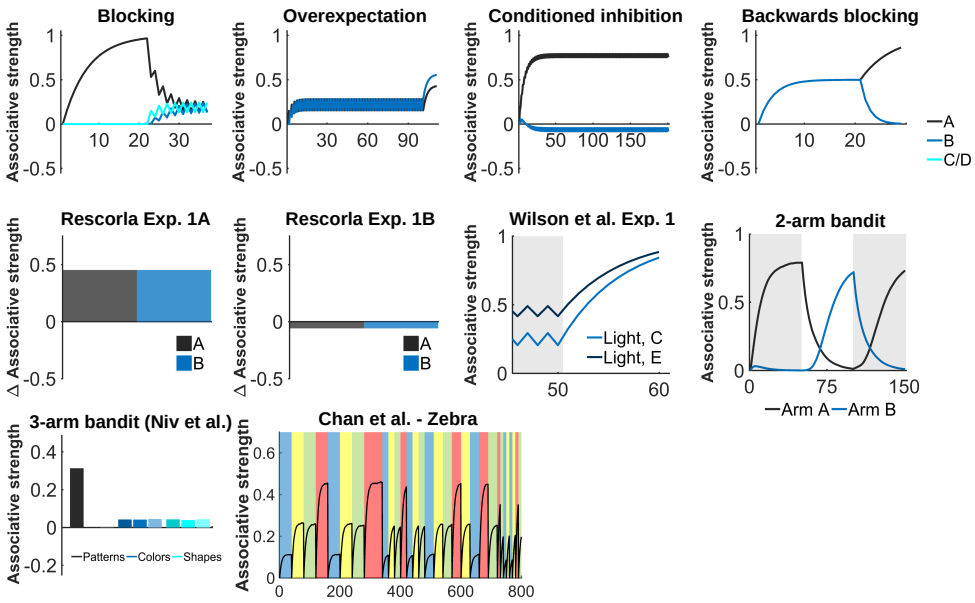
Appendix 0 Figure 8. Simulated behavior of the Hybrid model for the same learning experiments in the main text.

to the Hybrid and RW model, the fRL+decay can capture some, but not all of the simulated learning effects. For example, the fRL+decay model captured backwards blocking and partly captured blocking and conditioned inhibition, but could not capture overexpectation or the experiments in ***Rescorla*** (***2000***) and ***Wilson et al.*** (***1992***). The model also does not keep track of beliefs over latent-states, and so, cannot model latent-state beliefs during the photograph portion of the experiment in ***Chan et al.*** (***2016***).

Case study details

Both case studies involved a learning and decision-making task based on an N-arm bandit. The probability of arm choice was modeled using a soft-max function as discussed in the Model section. Latent-state learning was modeled using Algorithm 1. Two parameters were fixed in both case studies: the variance learning rate ($\beta_0 = 0$) and initial standard deviation ($\sigma_0 = 0.2$). With parameters σ_0 and β_0 fixed, the latent-state transition parameter γ determines the degree to which latent-state inferences were utilized. Large γ describes a participant who frequently switches between latent states to explain rewards, and small γ describes a participant more likely to stay with one latent-state.

For the first task, we scaled reward outcomes to lie between -1/2 and 1/2 and encoded threat outcomes to be -1/2 for a distressing image and 1/2 for a non-distressing image. These transformation ensured reward and threat outcomes were centered and had the same range of values. With outcomes centered, the RW model and latent-state due not differ due to centering. Three cues were associated with each RW model, and each latent state in the latent-state model. One cue encoded whether the first arm was chosen, another cue encoded whether the second arm was chosen, and the last cue encoded whether the third arm was chosen. The latent-state model encoded 3 hypotheses based on which of the 3 arms should be chosen. For each hypothesis, associative strength was initialized to be either 0.9 when the cue encoded the arm that should be chosen based on the given hypothesis or 0.1 otherwise. As done in the simulation experiments, the latent-state model started with 6 latent-states, 3 to capture initial model building of the 3 hypotheses and another 3 replicates of these hypotheses to allow for participants to flexibly build new predictions models if needed. Latent-state beliefs were initialized to be equally divided among the first 3 latent-states.



Appendix 0 Figure 9. Simulated behavior of the fRL+Decay model for the same learning experiments in the main text.

For the second task, we scaled reward outcomes to lie between -1/2 and 1/2. For the RW model, participant behavior was described with 6 cues: 2 to encode whether the participant chose Arm A or Arm B, 2 to encode whether the participant followed or opposed the caregiver's suggestion, and 2 to encode whether the participant followed or opposed the confederate's suggestion. Four parameters (one learning rate, one soft max, and two initial associative strengths) were fit to participant behavior. Two initial associative strengths (one for following the caregiver and one for following the confederate) were fit to participant behavior to reflect individual variation in social preferences. Initial associative strengths for opposing the caregiver and confederate were defined so that the associative strengths for following/opposing the caregiver sum to zero and similarly for following/opposing the confederate.

For the latent-state model in the second task, participant behavior was described by assuming the participant weighs three competing hypotheses: whether to make decisions based on their own expectations, caregiver suggestion, or confederate suggestion. The same six cues from the RW model were used, but divided among the three hypotheses. The first hypothesis is associated with two cues to encode whether the participant chooses Arm A or Arm B; the second hypothesis is associated with two cues to encode whether the participant follows or opposes the caregiver's suggestion; and the third hypothesis is associated with two cues to encode whether the participant follows or opposes the confederate's suggestion. Five parameters (learning rate α_0, softmax parameter, latent-state transitions γ, and two initial associative strengths) were fit to data. There was one additional parameter in the latent-state model compared to the RW model. Similar to the RW model, we allow initial associative strength for following the caregiver and confederate to be fit to behavior. Initial associative strengths for opposing the caregiver confederate defined so that the associative strengths for following/opposing the caregiver sum to zero and similarly for following/opposing the confederate. Lastly, the latent-state model started with 6 latent-states, 3 for initial model building of the 3 initial hypotheses and 3 replicates of these latent states to allow for participants to flexibly build new predictions models if needed. Latent-state beliefs were initialized to be 1/3 for the first three states and zero otherwise.

In each task, expected rewards for the latent-state model were calculated by taking a weighted average of associative strengths over latent-states which use similar cues (e.g., over latent-states which use caregiver suggestions as cues), with weights given by beliefs in the respective latent

states normalized to sum to one. Additionally, beliefs for the latent-state model were summed over latent-states which use similar cues.