A flexible and generalizable model of online latent-state learning

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

1 Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI, USA

2 Department of Psychiatry, University of Wisconsin, Madison, WI, USA

* cochran4@wisc.edu

Abstract

Traditional learning models such as the Rescorla-Wagner model fail to describe important phenomena, notably the rapid return of fear after extinction. To address this shortfall, evidence converged on the idea that learning agents rely on latent-state inferences, i.e. an ability to form beliefs over competing hypotheses. Our goal was to develop a model of latent-state inferences that uses *online* (also known as recursive or adaptive) updates and that can describe behavior in a diverse set of experiments. The resulting model combines the popular Rescorla-Wagner model with an approximate Bayesian filter. In simulation, we demonstrate the model's ability to reproduce learning effects both famously explained and not explained by the Rescorla-Wagner model and to capture recent experiments testing latent-state inferences. We then show the model explains participant behavior better than the Rescorla-Wagner and other models using a novel dataset (N=57 and N=24) of two experiments. Lastly, we derive our model as an online algorithm to maximum likelihood estimation, demonstrating it is 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.

Author summary

Computational researchers are increasingly interested in a structured form of learning known as latent-state inferences. Latent-state inferences is a type of learning that involves categorizing, generalizing, and recalling competing hypotheses about one's environment and is used in situations when the correct hypotheses is

latent or unknown. This type of learning provides a powerful explanation for overgeneralization of a fear memory and for the cognitive role of certain brain regions important to cognitive neuroscience and psychiatry. Accordingly, latent-state inferences is an important area of inquiry. Through simulation, case studies, and theory, we establish a new model of latent-state inferences. Moving forward, we aim to use this framework to measure latent-state inferences in healthy and psychiatric populations.

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 [1]. 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 [2–4]. One significant insight this literature has made is the differentiation of model-free and model-based learning [5, 6]. 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 [7], 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 [8, 9], 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 [10]. 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-signalled 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) [11]. This rapid return of extinguished responding is an important phenomenon with implications for treatment of clinical disorders such as anxiety disorders and substance use [12, 13]. 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 [10]. 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 [14], it stimulated subsequent research developing latent-state models of reinforcement learning [15–18]. 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 [19].

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 [20] 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 [16, 21, 22]. 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) [22]. 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 strengths of computational modeling applied to the problem of decision-making is that the learning theory is mathematically formalized [23]. This strength 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 general 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 present a model of online latent-state learning and to verify the suitability and utility of this model for a diverse set of experiments.

Results

We validated our model using simulation and case studies. We used simulation to verify that our latent-state model can reproduce historically-important learning experiments. We used case studies to verify that our latent-state model can describe learning and decision-making behavior using data from two experiments with human participants. Our model was compared to the Rescorla-Wagner (RW) model and two latent-state learning models: the infinite mixture model of Gershman and Niv in [16] and Gershman (2017) model in [18]. For each model, we computed associative strength of a cue, defined as the expected reward conditional on the cue being presented alone. For models of latent-state learning, we also computed beliefs in a latent state, defined as the estimated posterior probability of a latent state given observations.

Parameters were fixed throughout simulation (Table 1) and chosen for the infinite-mixture model and Gershman (2017) model from their respective papers, except for the maximum number of latent states which we fixed at 15 to reduce memory requirements. S1 Appendix provides additional details and considers three other models. Accompanying code is publicly-available at https://sites.google.com/site/amylouisecochran/.

Blocking and other effects

We first tested whether our model could reproduce learning effects that established the RW model as a powerful model of associative learning (Fig 1). In a blocking experiment, for instance, a learning agent is conditioned to associate a cue (Cue A) with a reward, leading to a high associative strength for Cue A.

Model	Parameter	Description	Value
Our model	α_0	Associative strength parameter	0
Our model	β_0	Standard deviation parameter	0.10
Our model	γ	Latent-state transitions	4.5×10^{-3}
Our model	σ_0	Initial standard deviation	0.5
Rescorla-Wagner	α	Learning rate	0.15
Infinite-mixture	α	Concentration parameter	1
Infinite-mixture	M	Number of particles	100
Gershman (2017)	α	Concentration parameter	0.1
Gershman (2017)	η	Learning rate	0.3
Gershman (2017)	σ_r^2	Reward variance	0.4
Gershman (2017)	σ_x^2	Cue variance	1
All models	τ	Soft-max parameter	5

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

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, our latent-state model, and the Gershman (2017) model predict blocking because of the way associative strength is updated in each model: 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 model, our latent-state model, and the Gershan (2017) 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. The infinite-mixture model does not update associative strength in the same way as the other three models, leading to different predictions for blocking, overexpectation, or conditioned inhibition experiments.

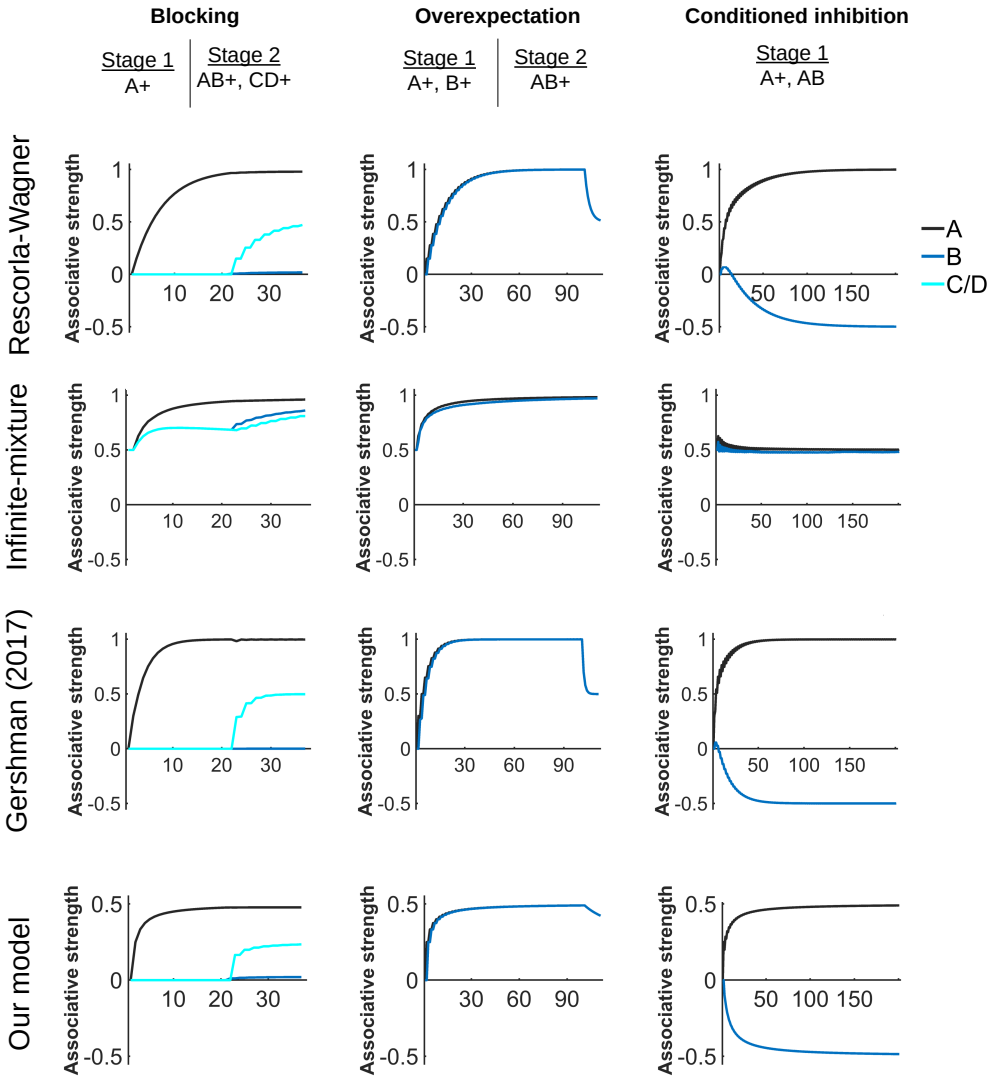

Fig 1. Simulated associative strengths of cues during blocking, overexpectation, and conditioned inhibition experiments, which the Rescorla-Wagner (RW) model famously could explain. Built upon the RW model, our latent-state model and the Gershman (2017) model can also explain these experiments.

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 1).

Associability depends on history of cue presentation

Our model encodes the history of cue presentation in effort matrices $B_l(t)$, endowing associabilty in our model with unique properties. One example is backwards blocking (Fig 2). 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. The infinite-mixture and the Gershman (2017) model also did not capture backwards blocking. By contrast, the associative strength of Cue B correctly decreased in the second part for our 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. Mathematically, applying the inverse of $B_l(t)$ 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.

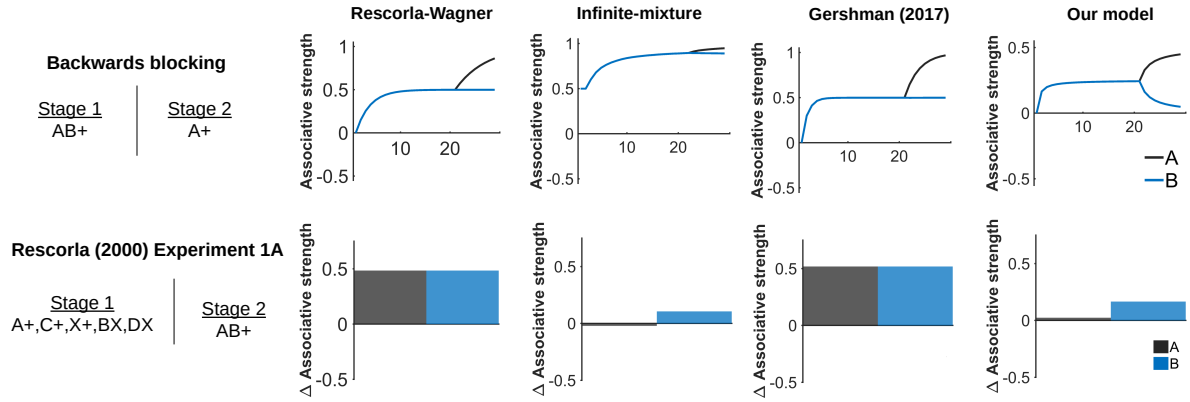

Fig 2. Associability depends on history of cue presentation. Associative strength was simulated for a backwards blocking experiment in which associative strength of a cue decreases even though not presented. Total change in associative strength was simulated during Stage 2 of Experiment 1A in [24] in which associative strength increases more for Cue B than Cue A.

History of cue presentation is also important for reproducing Experiment 1A in [24]. 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 only 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, our latent-state model learns that either Cue B inhibits the reward (food) directly or indirectly through Cue X in the first stage [24]. This latter inhibition is captured by an interaction term between Cue X and Cue B. 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 $B_l(t)$ rotates the cue vector $c_l(t)$ to allow associability to behave differently than in an RW model. Only the infinite-mixture model captured greater associability of Cue B during stage 2, whereas the Gershman (2017) model does not.

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 [8] in which associability remains high in unpredictable circumstances.

To illustrate, we considered Experiment 1B in [24], which involves the same conditioning schedule as Experiment 1A except compound AB is *not* reinforced in Stage 2 (Fig 3). 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 learning agents 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. By contrast, the other models for the given set of parameters do not detect the change in experimental conditions, causing cues A and B to change a similar amount during Stage 2.

A shift in latent state beliefs can also explain explain Experiment 1 in [25]. 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 [9].

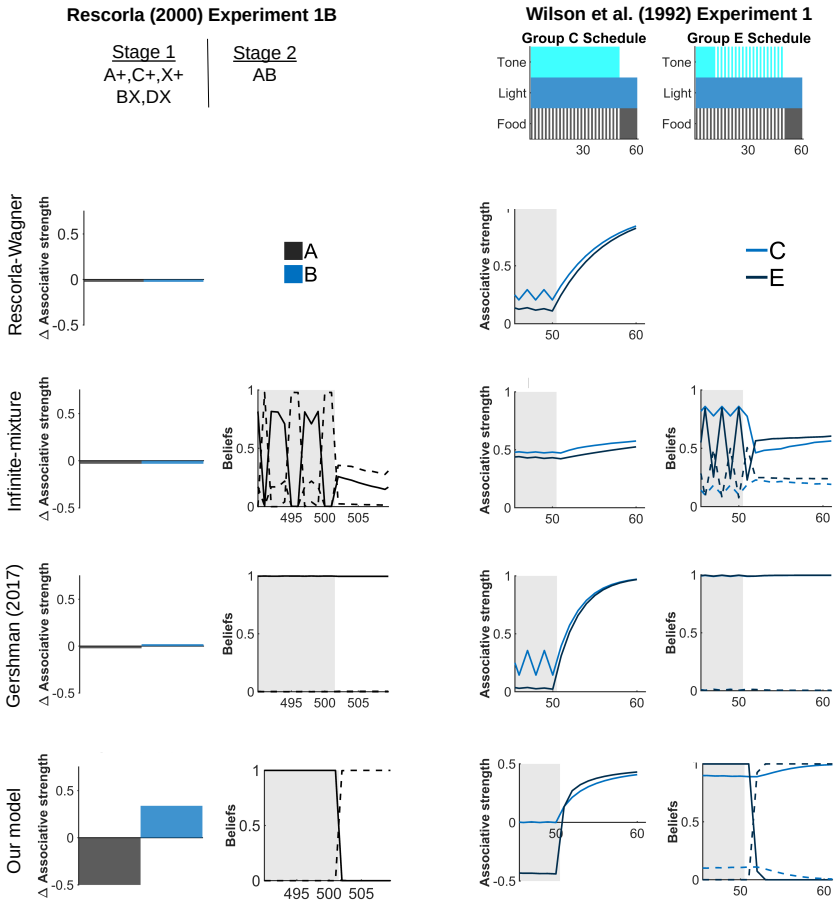

Fig 3. Associability depends on latent-state beliefs. Total change in associative strength was simulated during Stage 2 of Experiment 1B from Rescorla (2000) [24] in which associability was greater for Cue B during Stage 2 than Cue A. Associative strength was simulated for two groups in Experiment 1 from Wilson [25], which pointed to greater associability for the light in Group E than Group C in the final trials of the experiment. Beliefs in the first latent state (solid lines) and the second latent state (dashed lines) are shown for the three models with latent states.

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 3**B**). This shift would be observed as higher associability of the light in Group E than Group C, agreeing with experimental results. By contrast, the other models for the given set of parameters do not detect the change in experimental conditions, causing learning agents to respond more greatly to the light in experimental condition C compared to E.

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.

Reinstatement with renewal

Latent-state learning was offered as an explanation of reinstatement after extinction and the role of context in renewal [10, 14, 15]. We simulated the reinstatement with renewal task in Redish et al [10]. Choosing arm 1 always yields a reward in the first (acquisition) and last (reinstatement) stages of the experiment, except when a reward was obtained on the trial immediately prior. No rewards are obtained during the middle phase (extinction) or when any other arm is pulled. Two experimental conditions are considered: one in which the same context is provided through each phase (no renewal) and another in which a different context is provided during the extinction phase (renewal). Following [7] and [18], context was encoded as a separate cue in all models except our latent-state model. For our latent-state model, we propose that context signals to the agent to shift their beliefs to either a new state when context B is first presented or to an old state when context A is again presented after context B.

In both experimental conditions, our latent-state learner slowly detects the shift when arm-reward contingencies first shift and switches their beliefs to a new latent state, effectively consolidating the memory of the first task phase. 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. Introducing a different context during extinction (renewal) allowed the learning agent to better detect the underlying structure of the task, switching to a new latent state earlier, resulting to higher associative strength in the final phase. The infinite-mixture model predicts that the learning agent switches beliefs in a new latent state only when a different context is introduced (renewal) otherwise using one latent state to predict for all phases of the task. For the given parameters, the Gershman (2017) model uses only one latent state for all three phases of the task for both the no renewal and renewal conditions.

Weighing explicit 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 [26],

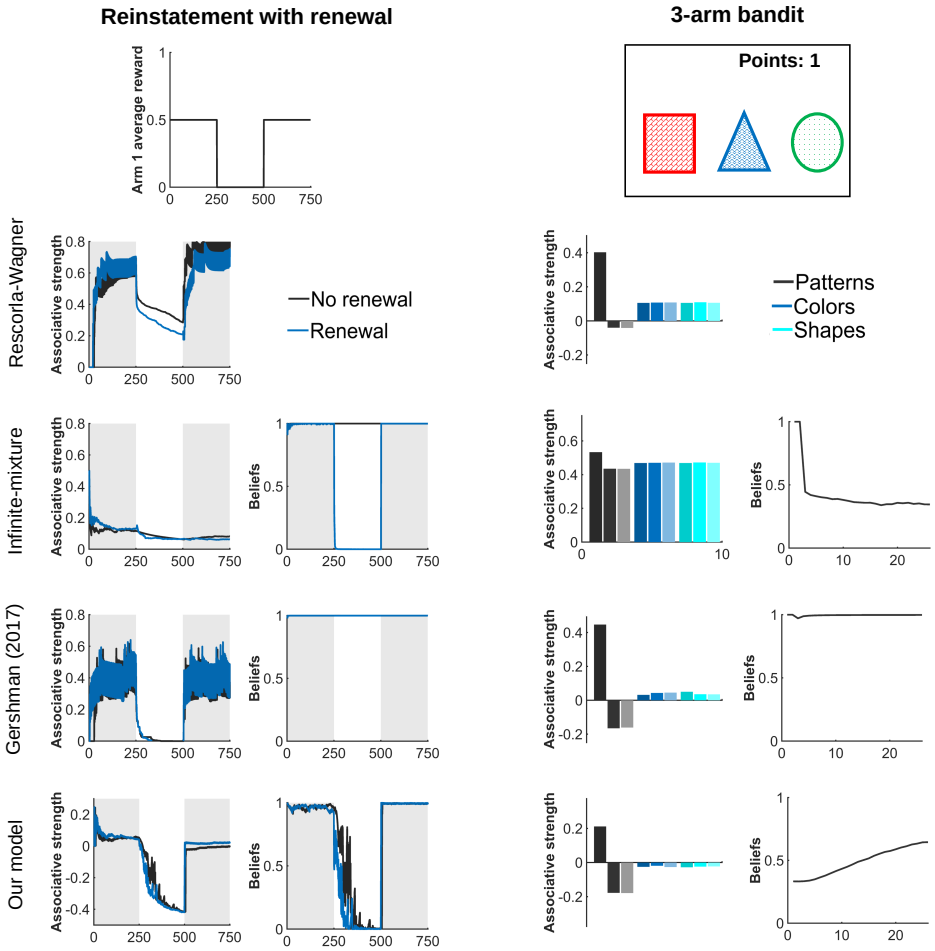

Fig 4. Associative strength and beliefs were simulated in an experiment which involves reinstatement with renewal [10] and from an 3-arm bandit experiment that explicitly asked to participant to correctly identify one of three competing hypotheses: patterns predict rewards, colors predict rewards, or shapes predict rewards. For the latter experiment, only the final associative strengths are shown. For both experiments, only beliefs in the first latent state are shown.

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 represented the three competing hypotheses: patterns are relevant, colors are relevant, or shapes are relevant (Fig 4). For comparison, other models used all 9 possible features as separate cues.

Our latent-state model predicts a high associative strength for the target feature and a low associative strength for other features in the relevant dimension (patterns). These predictions accurately reflect the

predictive value of features in the relevant dimension. Meanwhile, features in non-relevant dimensions acquire no associative strength, correctly signalling that these features are not predictive. The RW model, however, predicts features in non-relevant dimensions also acquire associative strength even though they do not predict rewards.

Our model also predicts that a learning agent increases their belief in the hypothesis that patterns are predicting outcomes as evidence accumulates (Fig 4). This prediction is possible, because each latent state explicitly encodes a hypothesis (e.g., "patterns are relevant") in our latent state model. Other latent-state models, infinite-mixture and Gershman (2017) model, do not predict that a learning agent increases their beliefs in any one state, because each latent state does not explicitly encode a hypothesis about features. In other words, latent states do represent explicit hypotheses for the task. Subsequently, an agent would need to integrate information over latent states to recover a belief in whether patterns, colors, or shapes are relevant.

Maintaining beliefs over multiple hypotheses

Our latent-state model can capture learning in situations not captured by other models. One example is when an agent must explicitly maintain beliefs over competing hypotheses, which was tested in the two-day learning task of [27]. 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 to expect certain animals in certain sectors. During the photograph part of the task, participants were asked to infer the sector based on an observed sequence of animal photographs.

The RW model can capture the tour part by building a separate RW model for each animal of the five animals, whereby which animal is observed is encoded as a five-dimensional vector of zeros and ones. (Fig 5A). The RW model, however, cannot describe how a participant uses this knowledge to infer the latent sector during the photograph part. The infinite-mixture model can capture the tour part by combining animal cues and sector cues, but would require modifications to capture the photograph part. First, the infinite-mixture model would need additional computations to infer the sector from observed animals, since latent states do not correspond to a sector and so beliefs in latent states are not beliefs in a sector (Fig 5B). Second, the sector is used as a cue in the tour part but is latent in the photograph part. The model is not designed to handle latent cues. The Gershman (2017) model was not used to model either part, since it is defined for only a one-dimensional reward. The model would need to be extended to multidimensional rewards. Even if extended, the model would run into the same issues as the infinite-mixture model for the

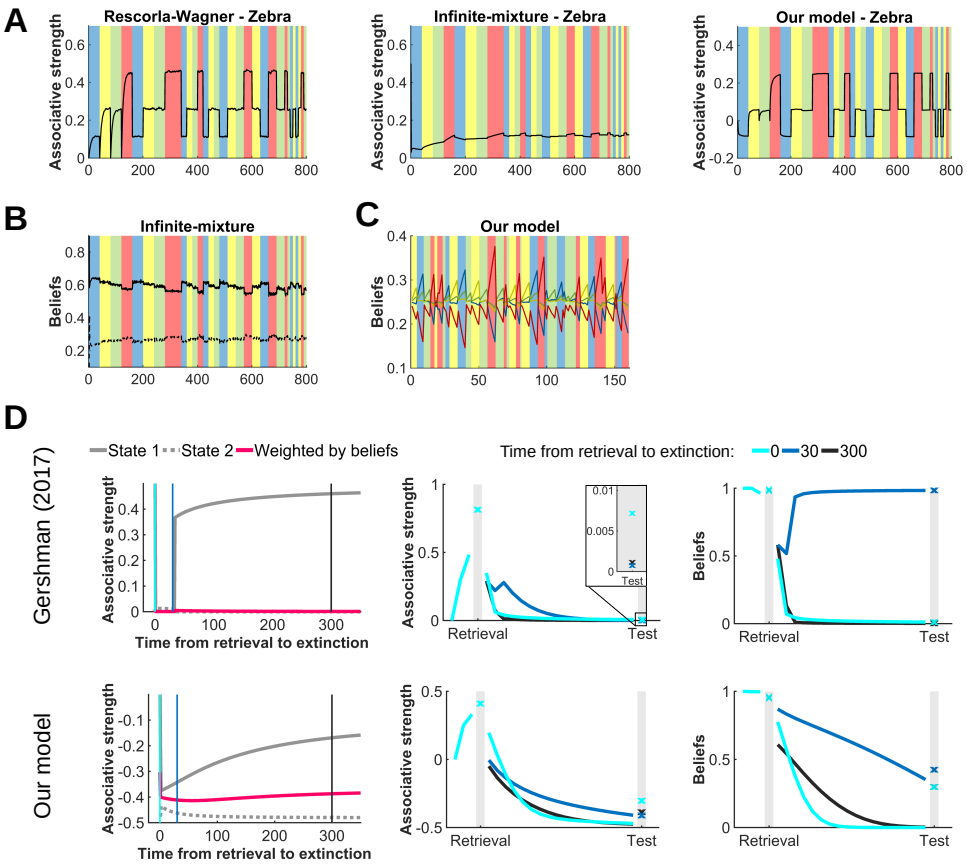

Fig 5. (A) Associative strengths (only one animal shown) from the animal safari task in [27]. Rewards were multidimensional, limiting which models could be simulated. Latent state beliefs for **(B)** the infinite-mixture model during the tour part of the task and **(C)** our model during the photograph part. **(D)** For the memory modification simulation, associative strength during testing as a function of time from retrieval to extinction and associative strength and beliefs through the task for three different times from retrieval to extinction.

photograph part (i.e., modifications to infer sectors and handle latent cues).

Our latent-state model can capture participant behavior in both parts of the task. During the tour part, the learning agent correctly learn associative strengths (i.e., probabilities in this task) of animals in each sector. During the photograph part, our latent-state model captures how beliefs in each sector changes as animals are observed (Fig 5C). Thus with one model, we capture learning in both parts of this complicated experiment.

Memory modification

Our last simulation experiment explores the potential role of latent-state learning in memory modification. Monfils et al [28] in a rodent model and later Schiller et al [29] in humans showed that a single *retrieval* trial between acquisition and extinction can reduce the fear response to the cue in latter tests. The time between

retrieval and extinction was further shown to modify this effect. Only certain windows of times reduced the fear response. They proposed that the retrieval trial helped to reconsolidate the acquisition memory to be less fearful within this *reconsolidation* window. The Gershman (2017) model [18] includes a computational explanation of this phenomenon. With a similar Bayesian framework to the Gershman (2017) model, our model can be modified to include similar memory modifications (S1 Appendix).

We simulated the Monfils-Schiller experiment in [18], varying the time between retrieval and extinction (Fig 5D). Our model reproduces a qualitatively similar result as the Gershman (2017) model. There is reconsolidation window which can help reduce the fear memory. For example in both models, associative strength of the cue during testing (pink lines in Fig 5D) is smaller for a time of 30 from retrieval to extinction relative to time of zero or 300. Both models predict that beliefs in the first latent state (corresponding primarily with acquisition) remains relative high in the reconsolidation window, allowing the agent to update the acquisition memory to be less fearful. Outside the reconsolidation window, the agent updates the second latent state (corresponding primarily with extinction), leaving the acquisition memory relatively in tact. While qualitatively similar between models, the effect of retrieval is more pronounced in our model in terms of differences in magnitude of cue associative strength during testing.

Case studies

Here, we examine how well our latent-state model explains participant behavior from two novel learning tasks compared to the RW model and the Gershman (2017) model. Each task requires participants to chose between mutually-exclusive options or "arms" based on information provided. Our latent-state model is used to describe competing hypotheses (e.g., "arm 1 is better arm 2" vs. "arm 2 is better than arm 1"). Outcomes are continuous in both case studies, so the infinite-mixture model, which assumes binary outcomes, could not be compared. Parameters were fit to maximize the likelihood of arm choice assuming trials were independent. The RW model uses 2 fewer parameters than the other models in each task.

Reward-threat conflict

This experiment examined 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 points and a threat consisting of a distressing or non-distressing image. The participant was instructed to obtain as many points as possible. Probabilities of points and threat switched in the experiment in order to manipulate congruency versus conflict between approach (earn points) and avoidance (avoid distressing

images) goals (Fig 6A). For example, in the first 50 trials (congruent phase), the arm most likely to lead to points was least likely to lead to a distressing image. For all models, three cues were used to encode which arm was chosen. Our latent-state model, however, described participant behavior as weighing three competing hypotheses about which of the three arms should be chosen. The same three cues were used, but associative strengths were initialized to favor the arm associated with each hypothesis.

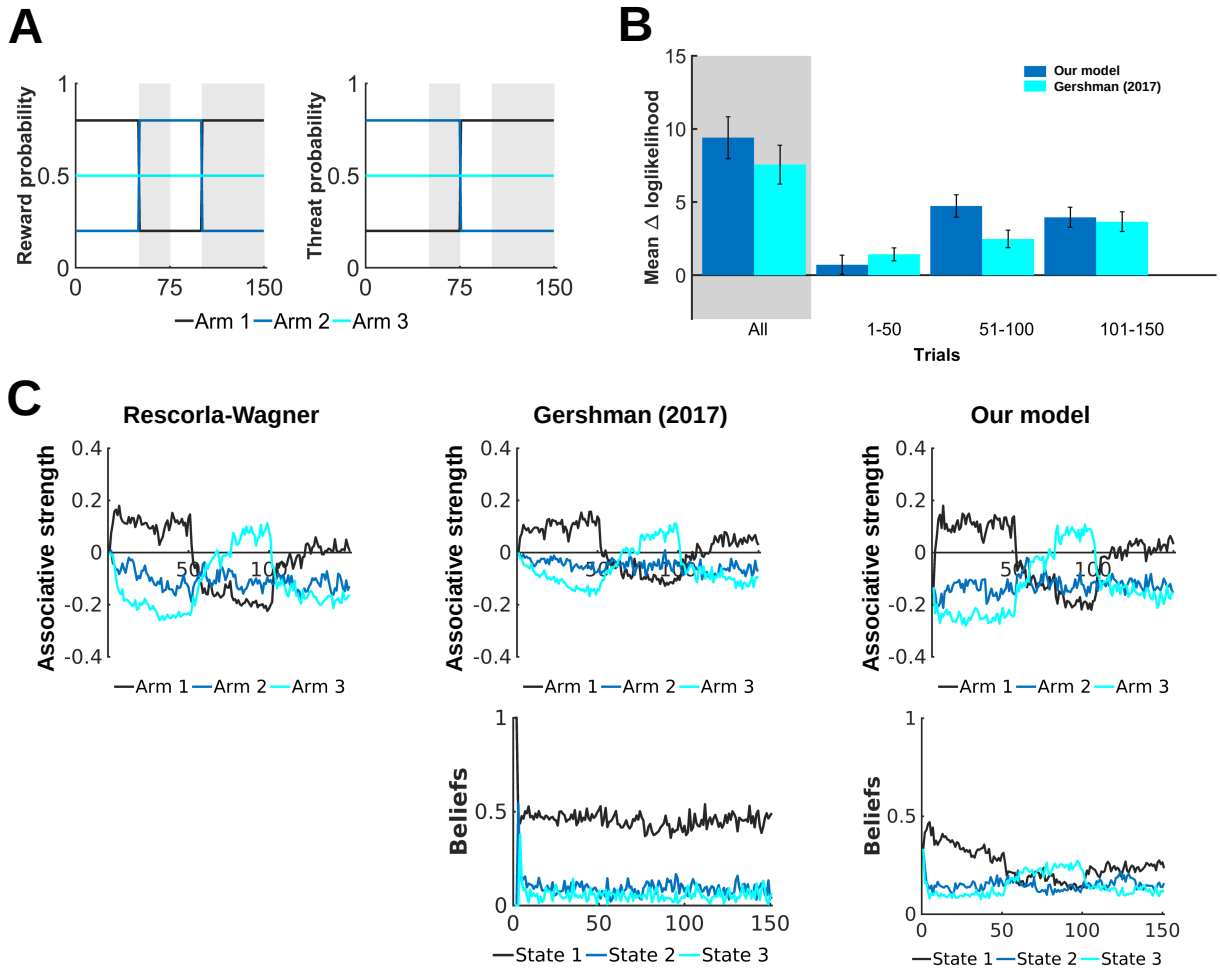

Fig 6. Participant behavior in a reward-threat conflict task. **(A)** Probability of reward and threat for each arm throughout the task. Half the experiment, maximizing reward conflicts with avoiding threat (gray regions). **(B)** Mean increase in log-likelihood of two models compared to a RW model over different sections of the task. **(D)** Associative strength for each arm and beliefs in the first three states as predicted by the models.

Our latent-state model fit participant behavior better than the RW model, with an average (95% CI) decrease in Aikake information criterion (AIC) of 14.8 (9.1, 20.6) and in Bayesian information criterion (BIC) of 8.8 (3.1, 14.5) (Fig 6B). 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. The Gershman (2017) model performed better than the RW model but not better than our latent-state model, with an average decrease over the RW model in AIC of 11.1 (5.8,16.5) and in BIC of 5.1 (0.2, 10.4).

To better isolate the contribution of latent-state learning specifically, we also fit our latent-state model with the effort matrices fixed (accomplished by replacing effort matrices B_l with $\alpha_0 I$). This model performed the best, with an average decrease over the RW model in AIC of 19.1 (13.3, 24.9) and in BIC of 13.1 (7.3, 18.9).

Beliefs from our latent-state model suggest that the participant recognizes the underlying structure of the task, switching their belief in arm 1 (represented as state 1), to a belief in arm 2 (represented as state 2), and then back to arm 1 (Fig 6C). Beliefs of the Gershman (2017) model, however, remain relatively fixed throughout the experiment.

Social learning

This task examined 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 points. 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 (Fig 6A). We used six cues for the RW model and Gershman (2017): 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. Our latent-state model, however, 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.

Our model fit participant behavior better than the RW model, with an average decrease of 9.7 (5.0, 14.3) in AIC and 5.1 (0.5, 9.8) in BIC (Fig 7B). Improvement over the RW model occurred in each third of the experiment. The Gershman (2017) model did not perform better than our model, with an average decrease over the RW model of 6.2 (3.2, 9.2) in AIC and 1.7 (-1.4, 4.7) in BIC. Hence, participants might be using latent-state inferences to weigh whether they should base their decisions on internal, caregiver, or confederate suggestions.

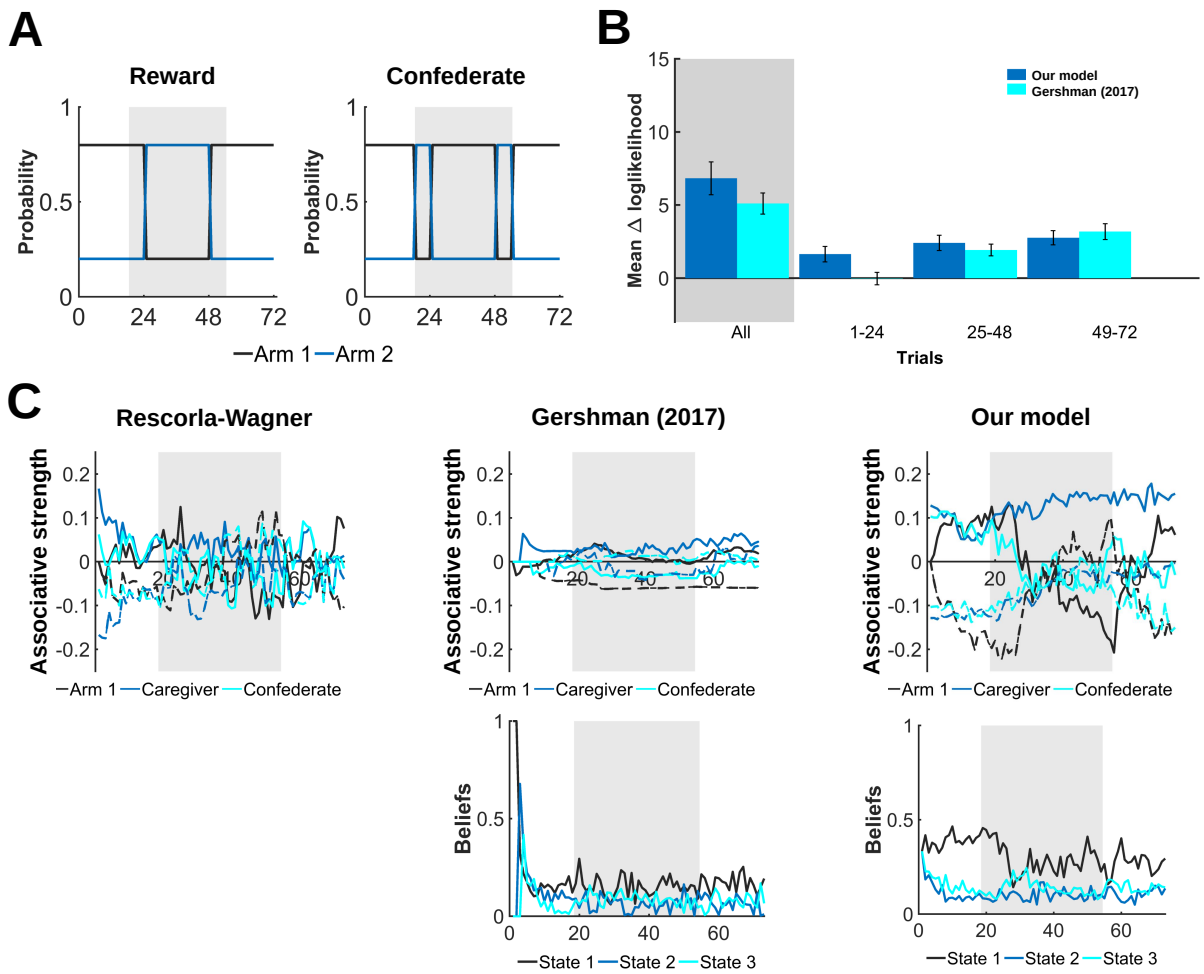

Fig 7. Comparison between models for describing behavior in social learning task. **(A)** Probability of rewards and confederate suggestion for each arm throughout the task. The confederate often provided bad suggestions (gray regions). **(B)** Average improvement of our model and the Gershman (2017) model over a RW model during different sections of the task. **(C)** Associative strengths with following (solid) or opposing (dashed line) arm 1, caregiver suggestion, and confederate suggestions. Beliefs in the first 3 latent states for the latent-state learning models.

We again fit our latent-state model with the effort matrices fixed to better isolate the contribution of latent-state learning specifically. Our model with fixed effort matrix performed the best again, with an average decrease over the RW model of 11.5 (6.4, 16.7) in AIC and 7.0 (1.8, 12.1) in BIC.

For our model, associative strengths for choosing arms reflects the changes in reward-arm contingencies (Fig 6C). Meanwhile, associative strength for following the caregiver stays high for most of the task, whereas associative strength for following the confederate started high but then declined. The first latent state (self) is associated with higher beliefs averaged over individuals compared to the next two latent states (caregiver and confederate suggestions).

Discussion

We presented a computational model of latent-state learning, using the Rescorla-Wagner (RW) model as its foundation, and positing that latent states represent distinct hypotheses in cue-outcome relationships. 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 other models. 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 uses online updates and is sufficiently flexible for describing learning in many experiments. The resulting model allows for unlimited latent-states and transitions between latent-states, memory modification, and a distribution over latent-state beliefs [10, 18, 24, 26, 27, 30, 31]. 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 features 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 [32]. 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 [30]. Other models of latent-state learning with less computational burden use approximate methods such as particle methods or maximum a posteriori estimates [14, 16, 18] and assumptions such as independent trials and binary outcomes [16], one latent state [26], or one latent-state transition [31]. Assumptions, however, limit the ability of any model to describe learning in varied experiments.

Critically, our latent-state model can capture reinstatement with renewal and other learning phenomena that require a more general number of latent states and transitions because of our use of an approximate Bayesian filter [33]. 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. [10]). The experiment in [27], 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. Some latent-state models cannot capture all of these classic learning features [16]. Further, our latent-state model better explained participant behavior than the RW model and the Gershman (2017) model in [18].

Our latent-state model allows associability to depend on the history of observations. This feature is thought to be important for describing learning effects not explained by the RW model, such as the Pearce-Hall effect, and is incorporated into newer models, such as the Hybrid RW model [8, 9, 34]. 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 [33, 35]. 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 our latent-state model to capture experimental results from Experiment 1 in [31] and Experiment 1B [24].

The presented model also makes testable predictions about how 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.

Methods

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 S1 Appendix, we derive our model of latent-state learning as 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:

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

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 . 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 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 [8, 36] suggest associative strength is path-dependent: associative strength depends on its history, changing over time [9]. 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 [37] 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, signalling 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 [8], 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 [16, 27, 31]. A learning agent assumes rewards are generated on trial t as follows: a latent random variable $X(t) \in \{1, \ldots, L\}$ 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 posteriori 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 $X(1{:}t) = (X(1), \ldots, X(t))$. Belief $p_l(t)$ in latent state l could then simply be the posterior probability $\rho_l(t)$ that latent variable $X(t)$ is l:

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

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 [30]. Efforts are made to reduce computation such as with particle method [16] or assuming one transition between states [31].

We capture a more general dynamic learning environment similar to [30], by assuming latent states are a Markov chain in which a latent variable transitions to itself from one trial to the next with probability $1-\gamma(L-1)/L$ and transitions to a new state (all new states being equally-likely) with probability $\gamma(L-1)/L$. 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

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

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. [33])

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

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

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

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.

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, Yu and Dayan [30] proposed a neural mechanism, via acetylcholine signalling, 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. Further, we also center rewards $R(t)$ around neutral expectations so that summed associative strength of cues is zero and yields neutral expectations at the start of the experiment. In most examples, this means $R(t) = 1/2$ if a reward was provided and $R(t) = -1/2$ if a reward was not provided. What remains are four parameters: α_0, β_0, γ, and σ_0. However as illustrated in case studies, we often need only two more free parameters 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, unlimited states, and memory modification (see S1 Appendix for details).

Mathematical justification

We justify our choice of model under the premise of *computational rationality* [32], 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 $1/t$):

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

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 [35] and [33] 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 ν. Unknown parameters are collected in $\theta = \begin{bmatrix} \vec{\mu}' & \nu \end{bmatrix}'$.

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 [35]. 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 [38]. We simply replace the gradient with an (stochastic) approximation that depends only on the current trial:

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

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

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

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

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

evaluated at $\hat{\theta} = \begin{bmatrix} \vec{\hat{\mu}}' & \hat{\nu} \end{bmatrix}'$ 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

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

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

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

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

In a latent-variable setting, a popular algorithm to perform MLE is expectation-maximization (EM) [39]. 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

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

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 $1/t$:

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

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.

Funding

ALC is funded by the National Institute of Mental Health (US; MH112876). JMC is funded by the National Institute of Mental Health (US; MH108753; MH106860) and the Brain and Behavior Research Foundation.

References

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

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

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

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

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

10. 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.

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

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

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

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

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

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

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

18. Gershman SJ, Monfils MH, Norman KA, Niv Y. The computational nature of memory modification. Elife. 2017;6:e23763.

19. 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.

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

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

22. 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.

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

24. 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.

25. 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.

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

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

28. Monfils MH, Cowansage KK, Klann E, LeDoux JE. Extinction-reconsolidation boundaries: key to persistent attenuation of fear memories. science. 2009;324(5929):951–955.

29. Schiller D, Monfils MH, Raio CM, Johnson DC, LeDoux JE, Phelps EA. Preventing the return of fear in humans using reconsolidation update mechanisms. Nature. 2010;463(7277):49.

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

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

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

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

34. 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.

35. 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.

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

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

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

39. 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.

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

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

S1 Appendix

Extensions

There are four 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.

Third, memory modification may alter how associative strengths and beliefs are updated. Gershman et al [18] offered three specific computational mechanisms by which memory can be modified. These mechanisms can be incorporated into our model due to similarities in modeling frameworks. The first mechanism describes how associative strength changes when a protein synthesis inhibitor is administered: the associative strengths $\vec{V}_l(t)$ for each latent state l is replaced with $\vec{V}_l(t)(1 - p_l(t))$ which reflects that changes are greatest for latent states with high beliefs (i.e., beliefs near 1). The second mechanism describes the influence of retrieval on a memory by way of "rumination". This mechanism is captured by repeatedly replacing associative strengths $\vec{V}_l(t-1)$ and variance $\sigma^2(t-1)$ with the present estimate $\vec{V}_l(t)$ and $\sigma^2(t)$ performing the updates on trial t again. The third mechanism describes how time corrodes current beliefs to reflect "forgetting" in between task phases. While the Gershman (2017) model uses a temporal kernel, we simply replace $p_l(t)$ with

$$(1-\gamma)^{ITI-1} p_l(t) + \left(1 - (1-\gamma)^{ITI-1}\right)/L,$$

where ITI is the time between trials within a task phase. This update amounts to repeatedly updating latent state beliefs for $ITI - 1$ iterations according to the Markov chain transition probabilities. Beliefs would become evenly distributed between latent states in the limit as ITI goes to infinity.

Lastly, many learning-state learning models do not assume the number of latent states *a priori* [14, 16]. 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 [30].

We present one possible mechanism based on Page's algorithm [40] for solving a problem known as change point detection (cf. [41]). The idea is to test whether we can reject our current model in favour 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.

Comparison to other learning models

We use similar notation to the main text with a vector $\vec{V}(t)$ to denote associative strengths on trial t, 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 $\vec{c}(t)$. We assume rewards are one-dimensional unless otherwise specified. Models are extended to decision-making tasks as was done with our latent-state model: estimating the expected reward *for* choosing each option, incorporating these rewards into a softmax function, and choosing an option according to the probability recovered from the softmax function.

In the main text, we compared our latent-state model to two other models of latent-state learning: the infinite-mixture model in [16] and the Gershman (2017) model in [18]. In prior work, these models helped to explain learning effects not captured by traditional learning models, such as extinction with renewal and memory consolidation [14, 16, 18]. Our latent-state model shares a similar Bayesian hierarchical framework to these models whereby rewards are assumed to be constructed in a generative way: a latent state is first randomly drawn on each trial and then rewards are drawn randomly with a distribution that depends on which latent state is drawn. This framework allows learning agents to learn about the hidden structure of their environment. Both models assume a prior for latent states arising from the Chinese Restaurant Process or time-sensitive Chinese Restaurant Process. In contrast, our model assumes latent states evolve according to a discrete-time Markov Chain.

Two quantities are compared in the main text between all models of latent-state learning. First, we compared beliefs in latent state l on trial t, which we defined to be the estimated posterior probability in latent state l given the history of cues and rewards:

$$\mathbb{P}\left[X(t) = l \middle| R(1{:}t-1), \vec{c}(1{:}t-1)\right].$$

Each model of latent-state learning uses different approaches to estimate the quantity above: the infinite-mixture model uses a particle method, the Gershman (2017) model uses maximum a posterior (MAP) estimates of latent causes, and our latent-state model uses an approximate Bayesian filter. Second, we compared associative strength of a cue c, which we defined to be the estimate of the expected reward if cue c

were presented alone:

$$\mathbb{E}\left[R(t)\middle|\vec{c}(t)=\vec{e}_c, R(1{:}t-1), \vec{c}(1{:}t-1)\right].$$

where $\vec{e}_c$ denote the cue vector if cue c was presented alone (i.e. a standard basis vector). This quantity can be expressed as a weighted average of expectations over latent states:

$$\sum_l \mathbb{E}\left[R(t)\middle|X(t)=l, \vec{c}(t)=\vec{e}_c, R(1{:}t-1), \vec{c}(1{:}t-1)\right] \mathbb{P}\left[X(t)=l\middle|\vec{c}(t)=\vec{e}_c, R(1{:}t-1), \vec{c}(1{:}t-1)\right],$$

which is similar to Equation (3) in [18] after summing over latent states prior to trial t and considering what rewards would be expected if cue c were presented alone and Equation (9) in [18] after considering what rewards would be expected if cue c were presented alone. The expectation

$$E\left[R(t)\middle|X(t)=l, \vec{c}(t)=\vec{e}_c, R(1{:}t-1), \vec{c}(1{:}t-1)\right].$$

is the cth entry in $\vec{V}_l(t)$ for our latent-state model and in terms of the notation in [18], is the weight w_{lc} for the Gershman (2017) model. Meanwhile, the quantity

$$\mathbb{P}\left[X(t)=l\middle|\vec{c}(t)=\vec{e}_c, R(1{:}t-1), \vec{c}(1{:}t-1)\right]$$

is equal to

$$\mathbb{P}\left[X(t)=l\middle|R(1{:}t-1), \vec{c}(1{:}t-1)\right]$$

for our latent-state model based on modeling assumptions and proportional to

$$\mathbb{P}\left[\vec{c}(t)=\vec{e}_c\middle|X(t)=l, R(1{:}t-1), \vec{c}(1{:}t-1)\right] \mathbb{P}\left[X(t)=l\middle|R(1{:}t-1), \vec{c}(1{:}t)\right]$$

for the infinite-mixture model and Gershman (2017) model by Bayes theorem. The last term in the above expression is approximated by beliefs in latent state l. Beliefs and associative strengths were estimated using associated code with the infinite-mixture model and with the Gershman (2017) model, modified to estimate associative strengths and to allow for decision-making.

The infinite-mixture model could be simulated in the case of multidimensional rewards for the animal safari task from [27], since it combines cues/rewards into a single vector of any size for each trial and does

not distinguish between cues and rewards when updating. In the animal safari task, we compared the expected value for reward d between our latent-state model and infinite-mixture model given cue c only and the history of rewards and cues:

$$\mathbb{E}\left[R_d \middle| \vec{c}(t) = \vec{e}_c, R(1{:}t-1), \vec{c}(1{:}t-1)\right].$$

The infinite-mixture model, however, could not be used in the case study, since rewards have to be binary. By contrast, we did not simulate the Gershman (2017) model in the case of multidimensional rewards, since it was not defined for such a setting. It does allow for continuous rewards, so we did use the model for the case study. These comparisons are reported in the main text.

We also considered three models of traditional associative learning: the RW model and two additional models. For these models, multidimensional rewards were simulated by building a separate model for each reward dimension. We considered the Hybrid model in [34], since this model allows associability to vary in time. 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 benefit of the Hybrid model, however, is that associability $A(t)$ is updated 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 [34]. This 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 from Rescorla [24] and Wilson et al. [25]. The model also does not keep track of beliefs over latent-states, limiting their ability to model latent-state inferences in the

photograph part of the experiment in [27]. The Hybrid model did not fit participant behavior well compared to the other models. The Hybrid model even had slightly worse performance to the RW model in the reward-threat conflict task (mean increase of 0.0 [-1.7, 1.6] in AIC and 3.0 [1.3, 4.6] in BIC) and in the social learning task (mean decrease of 0.4 [-1.7, 2.5] in AIC and mean increase of 1.9 [-3.9, 0.2] in BIC).

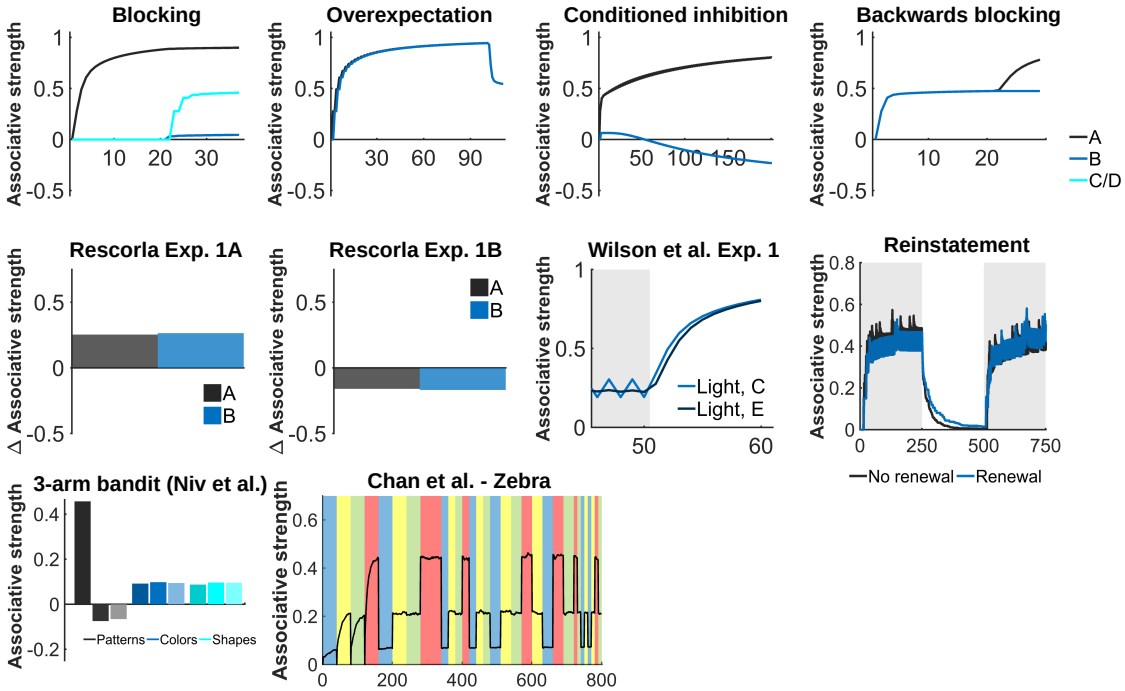

Fig 8. Simulated behavior of the Hybrid model for the same learning experiments in the main text.

We also considered the feature reinforcement learning with decay (fRL+decay) model in [26], since this model provided the best explanation of behavioral and neural data of the 3-arm bandit task in [26], 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 to allow for 'forgetting' of the associative strength. That way, the model captures how one might forget over time about associations between cues and rewards. 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 [26]. Similar 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 the experiment in Wilson et al. [25], and partly captured conditioned inhibition, but could not capture blocking, overexpectation or the experiments from Rescorla [24]. The model also does not keep track of beliefs over latent-states, and so, cannot model latent-state beliefs during the photograph part of the experiment in [27]. In the reward-threat conflict task, the fRL+decay model improved upon the Rescorla-Wagner model (mean increase of 15.5 [10.5, 20.3] in AIC and 12.4 [7.5, 17.4] in BIC). Among all the models considered, only our latent-state model with fixed effort matrix performed slightly better for this task. In the social learning task, the fRL+decay performed slightly better than the RW model (mean decrease of 3.3 [0.1, 6.5] in AIC and 1.0 in BIC [-2.2, 4.3]). The models with latent-state learning performed better than the fRL+decay model for this task.

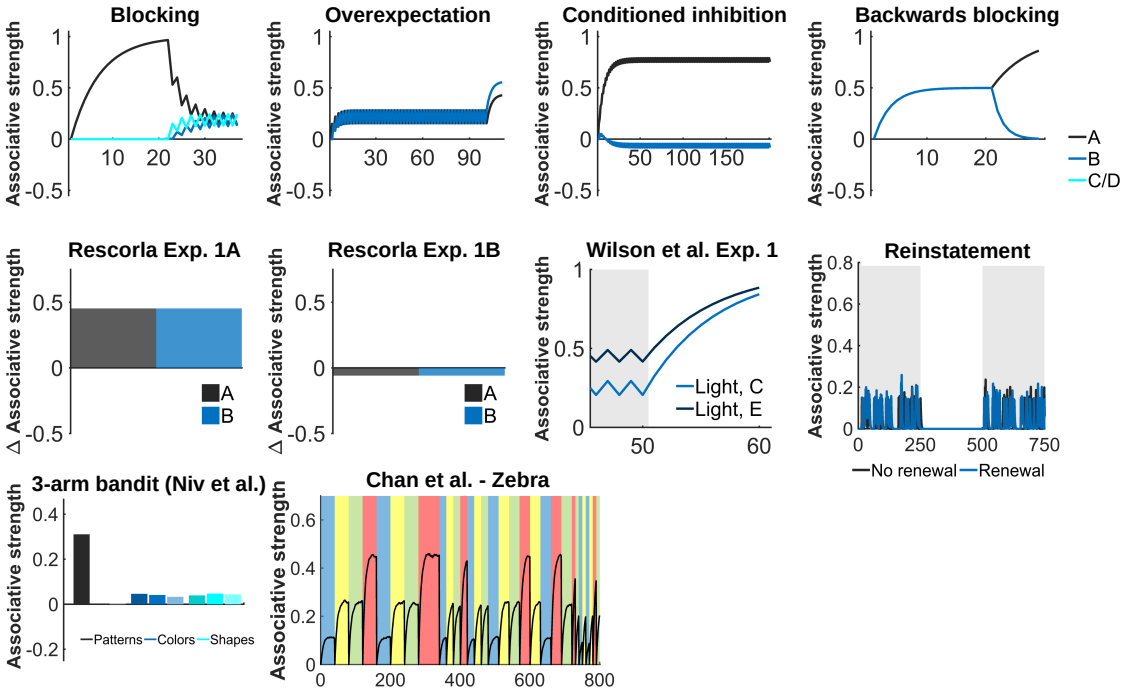

Fig 9. Simulated behavior of the fRL+Decay model for the same learning experiments in the main text.

Lastly, we again considered the Gershman (2017) model but with a higher concentration parameter ($\alpha = 1$). We used a larger concentration parameter, because we found the smaller $\alpha = 0.1$ relied on one latent state for a number of simulation experiments (i.e. beliefs for latent state 1 were near one). For both concentration parameters, the Gershman (2017) model could capture blocking, overexpectation, and conditioned inhibition, but could not capture the experiments from Rescorla [24] or Wilson et al [25]. Both concentration parameters also led to similar predictions for associative strengths for the reinstatement with renewal task in [10] and the 3-arm bandit task in [26]. Unlike the lower concentration parameter, the higher concentration parameter, however, did not lead to a reconsolidation window, i.e. a set of time lags between

retrieval and extinction in which the associative strength of the cue (pink line in Fig 10) could be decreased during testing.

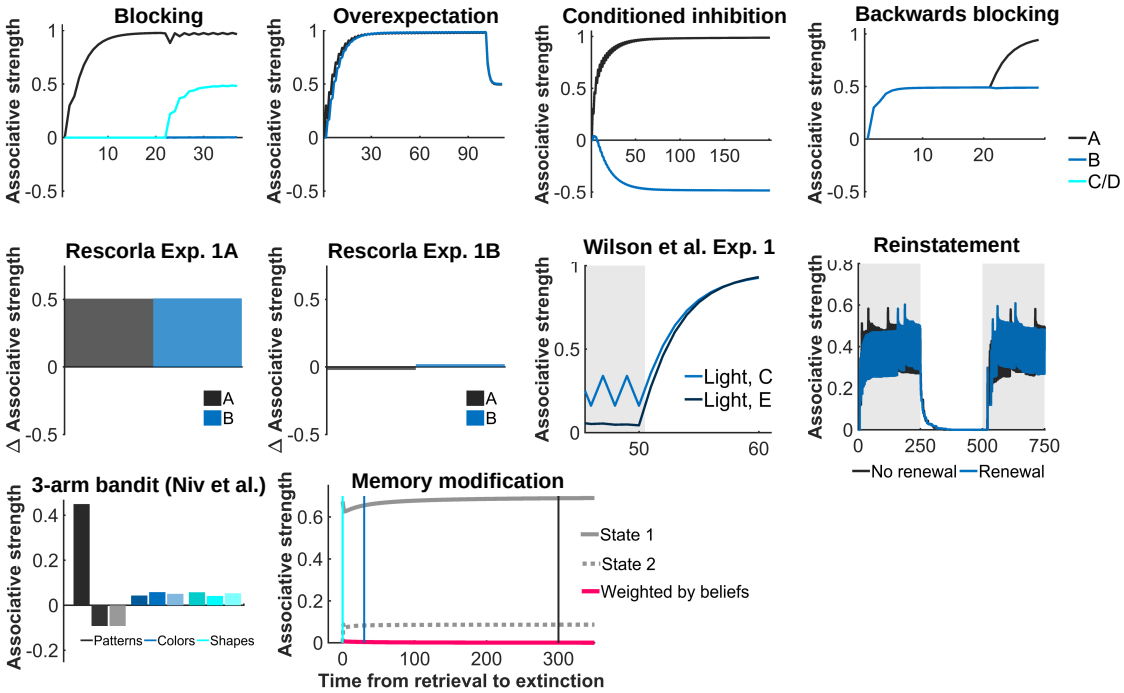

Fig 10. Simulated behavior of Gershman (2017) model with concentration parameter $\alpha = 1$ for the same learning experiments in the main text.

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 $L = 2$ for our model 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 1000 runs and report learning behavior averaged over simulation runs. Except for our last example (animal safari task from [27]), 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 [9]. 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 [24]

We followed the conditioning schedule in the original experiment [24]. 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 ($= [8 \times 12 \text{ days} + 4 \times 1 \text{ day}] \times 5$ 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 [25].

We followed the conditioning schedule used to simulate this experiment in the latent-state modeling paper in [16]. 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.

Extinction with renewal in [10].

We followed the extinction with renewal simulation experiment in [10]. This experiment consists of three phases (acquisition, extinction, and renewal) of 250 trials. Each trial, the agent must choose between 10 mutually exclusive choices or arms. Choosing arm 1 always leads to a reward of 1 during acquisition and renewal phases, except when rewarded on the trial immediately prior. Otherwise, no reward is given. No

rewards are given during the extinction phase. Two experimental conditions are considered in which either the same context is provided throughout each of the three phases (AAA) or the same context is provided during acquisition and renewal phases with a new context provided during the extinction phase (ABA). For all models except our latent-state model, cue vectors encoded arm choice and context. For our latent-state model, cue vectors only encoded arm choice; context was captured by shifting beliefs: re-setting beliefs to be split equally between latent state 1 and 2 when the new context (context B) is first presented at the start of the extinction and when the old context (context A) is presented again at the start of the renewal phase. This was used to capture the ability of a context to signal to the participant that need to be open to starting a new model in the presence of new cue or return to an old model in the presence of an old cue.

3-arm bandit game from [26]

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 [26], 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). For all models except our latent-state model, cue vectors encoded 9 possible target features. For our latent-state model, 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 in [27]

We followed the schedule as reported in [27]. 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 task in [27]. 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.

For all models, the "tour" portion of the task is described by assuming 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. Cue vectors encoded which of the four sectors the participant currently is visiting. We hence collapsed all contextual information (e.g., colors, sounds) into a single value to represent whether a participant is in the sector. For 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.

Only our latent-state model was used to model the "photograph" portion. For this model, 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.

Memory modification

We followed the simulation experiment in [18], which was designed to reflect the experiments in Monfils et al [28] and Schiller et al [29]. The experiment consisted of four phases. First a cue was reinforced on three

trials for an acquisition phase. Second, the cue is presented once but not reinforced for a 'retrieval' phase. Third, the cue is presented for 18 trials but not reinforced for a 'extinction' phase, and fourth, the cue is presented and not reinforced for a final 'test' phase. Further, we assumed that the time (with same units as inter-trial time) between each phase was 0, 20, x, and 200, respectively. Parameter x represents the time between retrieval and extinction and was varied from 0 to 350. The Gershman (2017) model was simulated as was done in [18]. Our model was simulated using the memory modification mechanism of rumination and forgetting (discussed in Extensions above) to update associative strengths and variance for a certain number of iterations and then to update latent-state beliefs, depending on the time between retrieval and extinction. The number of iterations for the rumination step was defined as the minimum of 3 and 1+x (with x defined above). This definition was chosen to agree with the Gershman (2017) model.

Case studies details

Both case studies involved a learning and decision-making task based on an N-arm bandit. Five models (RW, latent-state, Gershman [2017], hybrid, and fRL+decay) were fit to participant behavior. The infinite-mixture model was not fit to participant behavior, since it requires binary rewards and rewards were continuous in each example.

Our latent-state model was modeled using Algorithm 1. As done in the simulation experiments, our latent-state model started with a latent state for each possible hypothesis to capture initial model building and another latent state that replicates each hypothesis to allow for participants to flexibly build new predictions models if needed. Latent-state beliefs were initialized to be equally divided among the initial hypotheses. One parameter was fixed in both case studies: the variance learning rate ($\beta_0 = 0$). With parameter β_0 fixed, the latent-state transition parameter γ and the initial standard deviation σ_0 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. Meanwhile, a small σ_0 describes a participant who is sensitive to prediction error, causing frequent switching in latent states, and large σ_0 describes a participant who is not sensitive to prediction error.

Reward-threat conflict task

For the reward-conflict task, point outcomes were scaled to lie between -1/2 and 1/2 and threat outcomes were encoded to be -1/2 for a distressing image and 1/2 for a non-distressing image. Thus, models did not

differ due to centering. Reward $R(t)$ was then defined to be a weighted average of the point and threat outcome on trial t, with a participant-specific weight between zero and one that was fit to behavioral data.

Three cues were associated with each model or each latent state. 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. Our latent-state model also encoded 3 hypotheses about which arm to choose. This was accomplished by initializing associative strength to be either 0.9 when the cue encoded the arm that should be chosen based on the given hypothesis or 0.1 otherwise.

Three parameters were fit to each participant for the RW model (learning rate, soft-max parameter, and point/reward weight parameter). In addition to these parameters, an associability learning rate was fit for the hybrid model; a decay rate for the fRL+decay parameter; transition rate parameter and initial standard deviation for our latent-state parameter; and a concentration parameter and reward variance for the Gershman (2017) model. Thus, 3-5 parameters were fit depending on the model.

Social learning task

For the second task, we scaled reward outcomes to lie between -1/2 and 1/2. Participant behavior was described with 6 cues for 4 of the 5 models (RW, hybrid, fRL+decay, and Gershman [2017]). Two cues encoded whether the participant chose Arm A or Arm B, 2 encoded whether the participant followed or opposed the caregiver's suggestion, and 2 encoded whether the participant followed or opposed the confederate's suggestion. For our latent-state model, the same six cues from the RW model were used, but divided among three competing hypotheses: whether to make decisions based on their own expectations, caregiver suggestion, or confederate suggestion. 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.

Four parameters were fit to participant behavior for the RW model (a learning rate, a soft max parameter, and 2 initial associative strengths). Two initial associative strengths (one for following the caregiver and one for following the confederate) were included to reflect individual variation in social preferences at the start of the experiment. 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. In addition to these parameters, an associability learning rate was fit for the hybrid model; a decay rate for the fRL+decay parameter; transition rate parameter and initial standard deviation for our latent-state parameter; and a concentration parameter and

reward variance for the Gershman (2017) model. Thus, 4-6 parameters were fit depending on the model.