SUPPLEMENTARY INFORMATION

Data Driven Equation Discovery Reveals Non-linear Reinforcement Learning in Humans

Kyle J. LaFollette[1]

Janni Yuval[2]

Roey Schurr[3]

David Melnikoff[4]

Amit Goldenberg[5,3,6]

[1]Case Western Reserve University, Department of Psychological Sciences, Cleveland, OH, USA

[2]Google Research, Mountain View, CA, USA

[3]Harvard University, Department of Psychology, Cambridge, MA, USA

[4]Stanford Graduate School of Business, Stanford, CA, USA

[5]Harvard University, Harvard Business School, Boston, MA, USA

[6]Harvard University, Digital, Data and Design Institute, Cambridge, MA, USA

Phase 0 ..... **3**
Equation Recovery from Simulated Data 3
Model Recovery as a Function of Initial Conditions 7
Robustness to Noise Analyses 10
Phase 1 ..... **11**
Demographics 11
Visualization of Empirical Data 12
Proof of Quadratic Q-Weighted Model Point of Under-to-Over Estimation 14
Phase 1 Model Comparison 15
Recoverability of the Quadratic Q-Weighted Model 19
Phase 2 ..... **22**
Phase 2 Model Comparison 22
Phase 2 Model Recovery 25
Phase 2 Parameter Recovery 29
References **34**

Phase 0

Equation Recovery from Simulated Data

Can SINDy use behavioral data from RL tasks to discover the functions from which those data were generated? To answer this question, we simulated the behavior of artificial agents in a standard RL task. The behavior of the artificial agents was governed by learning algorithms that we selected. That is, the true data-generating process was known, and could therefore serve as an objective benchmark for evaluating SINDy. In what follows, we describe the learning task performed by our artificial agents, followed by the learning algorithms our agents used to solve the task.

Learning Task. The learning task was composed of 100 trials. On each trial, the learner observes one of two events: reward or no reward. The true probability of a reward changed trial-to-trial according to a Gaussian random walk (SD=0.1), bounded between 0.1-0.9; the initial value was drawn from a uniform distribution in the range 0.1-0.9.

The goal of the learner is to estimate, on each trial, the true probability of a reward. The learner's estimate on each trial $t \in \{0 \dots 100\}$ is denoted as Q_t, where $Q_{t=0}$ is the learner's initial estimate (always 0.5 in the case of simulated agents). Learners report Q_t at the end of each trial (i.e., after receiving or not receiving a reward).

Learning Algorithms. We created a variety of artificial agents, each of which learned using a different algorithm. We selected a few algorithms that were commonly used in the reinforcement learning literature: the classic Rescorla-Wagner learning algorithm (*1*), a Rescorla-Wagner algorithm with time-dependent exponential decay (*1*), and a Rescorla-Wagner algorithm with asymmetric learning rates for positive and negative feedback (*2–5*). The *Rescorla-Wagner algorithm* assumes that Q_t is updated as follows:

$$Q_{t+1} = Q_t + \alpha(r_t - Q_t)$$

Where r_t indicates the outcome of trial t (1 for reward, 0 for punishment or the omission of reward), and $\alpha \in [0,1]$ is the learning rate at which expectations are updated in response to the reward-prediction error $r_t - Q_t$. The change in Q_t can be written as a linear combination of terms discoverable by SINDy:

$$\dot{Q}_t = \alpha(r_t - Q_t)$$

Where the dot represents the discrete-time derivative operation. The second algorithm, the *Rescorla-Wagner model* with time-dependent exponential decay assumes that agents update Q_t following feedback proportionate to a learning rate that exponentially decays with time:

$$Q_{t+1} = Q_t + e^{-t/\lambda}(r_t - Q_t)$$

Where λ is the rate of decay of the learning rate. This decay parameter effectively reduces the influence of reward-prediction errors over time. The change in Q_t can be written as a linear combination of terms discoverable by SINDy:

$$\dot{Q}_t = e^{-t/\lambda} r_t - e^{-t/\lambda} Q_t$$

The third algorithm is a *Rescorla-Wagner model with asymmetric learning rates* for positive and negative feedback. The algorithm assumes that learning occurs at different rates in response to positive versus negative reward prediction errors:

$$Q_{t+1} = Q_t + \alpha(r_t - Q_t); \qquad \alpha = \{\alpha^+ \; if \; r_t = 1 \;\; \alpha^- \; if \; r_t = 0\}$$

Where α^+ and α^- are the learning-rates following positive and negative feedback, respectively. The change in Q can be written as a linear combination of terms discoverable by SINDy:

$$\dot{Q}_t = \alpha^+ r_t + (-\alpha^-) Q_t + (\alpha^- - \alpha^+) r_t Q_t$$

For each algorithm, we simulated data from the task described above 100 times. The output of each individual simulation was the estimated probability of receiving a reward on each trial, $Q_{1:100}$, as well as the reward received in each trial, $r_{1:100}$. The parameters governing the simulation were identical across agents (we relaxed this assumption below in the noise perturbation analysis), with parameter values sampled for each simulation from uniform distributions (Rescorla-Wagner $\alpha \sim U(0, 1)$; Rescorla-Wagner with time-dependent exponential decay $\lambda \sim U(5, 50)$; Rescorla-Wagner with asymmetric learning rates $\alpha^+ \sim U(0, 1)$ and $\alpha^+ \sim U(0, 1)$). Each simulation contained 100 agents all sharing the sampled parameter values. We provided SINDy with the simulated data to predict the algorithm governing our simulated agents (see Figure S1). A matrix of candidate functions was provided to SINDy for feature selection, on which all models were trained. This procedure was repeated for all the simulations per model with parameters resampled between simulations. See Methods for a more formal description of the SINDy algorithm and notation.

SINDy recovers the ground-truth learning algorithms. For each of the three algorithms, we evaluated whether SINDy was able to recover the ground-truth equation. This was done in two ways: First, we calculated the median coefficient of the determination (R^2) between the simulated *Q* values (generated by agents who used one of the three learning algorithms), and the

Q values generated from the algorithm produced by SINDy. Second, we examined the fraction of simulations in which the algorithm produced by SINDy recovered the ground-truth algorithm up to a small numerical difference in the parameter values (i.e., the same general form with identical terms, but not necessarily the same coefficients).

Note that SINDy explores as many possible algorithms as we provide possible combinations of candidate features, and in many cases an algorithm can be well-represented in a more parsimonious form than it was parameterized. For example, in a case where the simulated agent is using a Rescorla-Wagner algorithm with nearly equal asymmetric learning rates, the model can be approximately written as a standard Rescorla-Wagner algorithm with one learning rate α. In such a case, we expect SINDy to recover the more parsimonious algorithm (see a more comprehensive exploration relating model parameters in SI). Overall, the results of our analysis suggest that the median coefficient of determination (R^2) between the Q values produced by our simulated agents and the Q values generated by SINDy-produced algorithms was almost equal to 1 for all three ground-truth algorithms. The results reported below were observed with noiseless simulations.

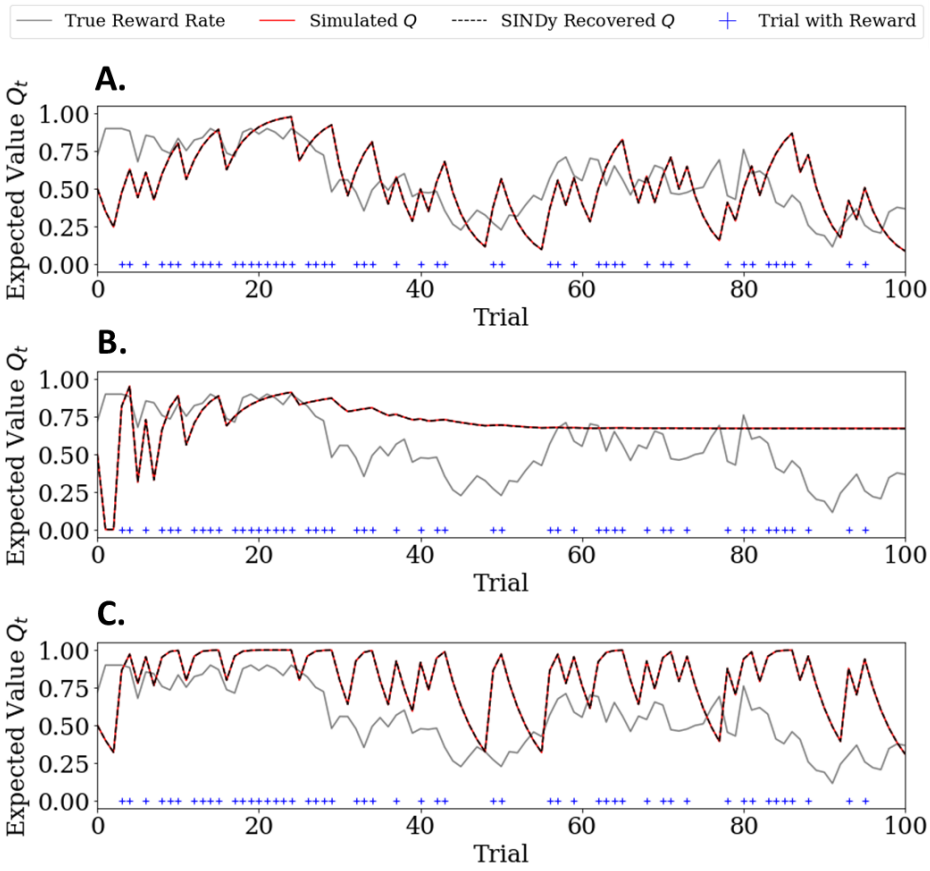

Figure S1. Representative examples of model fits to agents' data based on the three simulated models. The x-axis represents trial number. The y-axis represents the expected value Q after each trial. The grey line represents the true reward rate that the simulated agent is trying to capture. The red line is the ground-truth *Q value* as produced by the agent in the simulation. The black dotted line is the recovered *Q* value using the equation recovered by SINDy. Notably, the black dotted line (recovered Q) sits perfectly on top of the red line (ground truth Q) in all panels, indicating perfect model recovery. (A) Classic Rescorla-Wagner model with learning rate 0.3. (B) Rescorla-Wagner model with time-dependent exponential decay with decay rate t/10. (C) Panel Rescorla-Wagner model with asymmetric learning rates with learning rate to positive feedback is 0.8 and learning rate to negative feedback is 0.2. Blue plus markers are instances of reward.

For the Rescorla-Wagner algorithm, the median R^2 between simulated Q-values and the Q-values generated by the SINDy-recovered algorithm was R^2=0.99; 95%HDI=[0.91, 0.99] (HDI stands for highest density interval). In 64% of simulations SINDy recovered exactly the Rescorla-Wagner algorithm. As expected, given SINDy's inherent regularization, for regions of the parameter space where some of the parameters have extreme values, we found that SINDy recovered a simplified version of the true algorithm. For example, the remaining 36% of simulations recovered simplified versions of Rescorla-Wagner, such as an algorithm driven

purely by previous Q when learning rate approached 0, and a model driven purely by reward when learning rate approached 1.

For the Rescorla-Wagner algorithm with time-dependent exponential decay, the median R^2 between simulated Q-values and the Q-values generated by the SINDy-recovered algorithm was R^2=0.99; 95%HDI=[0.94, 0.99]. In 99% of simulations recovered exactly the decaying model, the remaining 1% of which were complicated by simulated decay rates that differed dramatically from the candidate decay rates we provided SINDy (see Methods for matrix of candidate functions).

For the Rescorla-Wagner algorithm with asymmetric learning rates, the median R^2 between simulated Q-values and the Q-values generated by the SINDy-recovered algorithm was R^2=0.982; 95%HDI=[0.908, 0.999]. In 34.8% of simulations recovered exactly the Rescorla-Wagner algorithm with asymmetric learning rates. The remaining 65.2% of simulations recovered simplified versions of the asymmetric algorithm. These included the classic Rescorla-Wagner algorithm when both positive and negative learning rates were near equal and not approaching zero or one, an algorithm driven purely by previous Q when both learning rates were approaching 0, and an algorithm driven purely by reward when both learning rates approached 1.

Model Recovery as a Function of Initial Conditions

As discussed in the main text, model recovery was dependent on the parameter values used to simulate the data. Certain parameter values **reduce** more complicated models (e.g., the RW model with asymmetric learning rates) to simpler forms (e.g., the classic RW model). For example, the data simulated from an "asymmetric" learning model with two practically equivalent learning rates (e.g., 0.4 vs 0.41) will likely mirror data produced from a classic RW model. Because the data in such cases are practically indistinguishable, and because SINDy encourages parsimony, SINDy would most likely discover the more parsimonious model (in this case, the classic RW model).

In Phase 0, when simulating data using the Rescorla-Wagner model, we sampled learning rates from $U(0, 1)$. Some of these learning rates, particularly those close to 0 or 1, reduce the RW model to a simpler form. For example, a RW model with a learning rate of 0 does not learn from

feedback, and as such reduces to only include a term for previous Q. Likewise, a RW model with a learning rate of 1 is myopically driven by feedback, but completely ignores prior expectations, reducing to a model that only includes a term for previous reward. In either case, the Rescorla-Wagner model and these more parsimonious models approach asymptotic equivalence, and SINDy would be correct to recover the more parsimonious model.

In accordance with these hypothesized reductions, 11.9% of models recovered with SINDy included only a term for the previous *Q* and no reward term, and 24.2% recovered a model with only a term for reward and no term for previous *Q* (see Figure S2A). The former was produced when learning-rate approached 0, whereas the later formed when learning-rate approached 1. When simulating the Rescorla-Wagner model with time-dependent exponential decay, 1.5% of the models recovered with SINDy included nuisance terms in excess of the reward and previous *Q* terms (see Figure S2B). These nuisance terms were recovered when decay rate approached 5. This may be explained by the limited exponential decay features included in the candidate feature matrix provided to SINDy: $e^{-t/10}$, $e^{-t/20}$, and $e^{-t/30}$. Only those features provided in the candidate feature matrix are discoverable with SINDy, and so data generated from models with more distinct decay rates may not be recoverable.

Finally, when simulating the Rescorla-Wagner model with asymmetric learning-rates, a wider range of more parsimonious models were recovered. 50.3% of models recovered with SINDy were the standard Rescorla-Wagner model, which occurred when both learning rates approached equivalence and tended away from both 0 and 1 (see Figure S2C). Similar to our Rescorla-Wagner simulations, previous *Q*-only models were recovered when learning-rates approached 0 (4.3%) and reward only models were recovered when learning-rates approached 1 (10.6%). The asymmetric model was recovered when learning rates tended away from each other (34.8%). In the former three cases, the asymmetric learning model and more parsimonious models approach asymptotic equivalence, and we expected SINDy to recover the more parsimonious models.

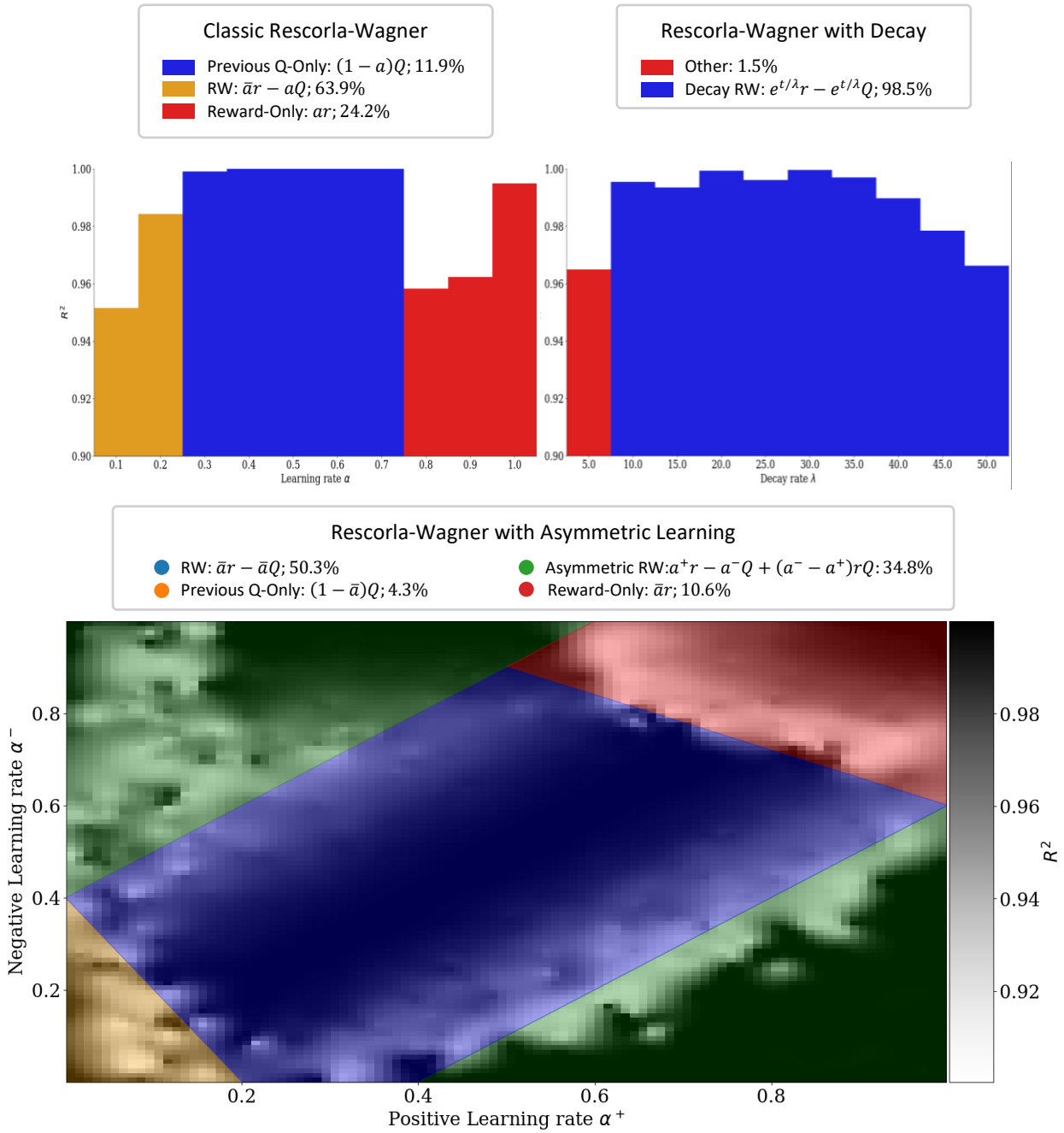

Figure S2. Proportions of models recovered and their fit (R^2) to the simulated data using different parameter values. (a) Models recovered from simulation using classic Rescorla-Wagner. The x axis represents the learning rate and the y axis the R^2 of the model. The blue color indicates the values for each a Rescorla Wagner was captured. When the learning rate was high, it seems that the model captured by SINDy was only focused on reward (red). When the learning rate was low, the model ignored reward (yellow). (b) Models recovered from simulation using Rescorla-Wagner with exponential time decay. It seems that SINDy was not able to capture the

model when decay rates were especially low (red) (c) Models recovered from Rescorla-Wagner model with asymmetric learning rates for positive and negative feedback. The boundaries on which SINDy recovered different reduced forms of the asymmetric model were a function of the model's positive and negative learning rates (x and left y axes). Fit (right y axis) tended to be worse on these boundaries, as SINDy would struggle between models on either side.

Robustness to Noise Analyses

Since we expected that human-generated behavioral data would be much noisier than our simulated data, we examined SINDy's robustness to noise both when introducing noise into the agents' predictions as well as when changing the number of trials that SINDy can use to discover an algorithm. We simulated data again for each of the above algorithms but made two modifications. First, we perturbed Q_t with noise sampled from $N(0, 0.05)$. Second, we modified the number of simulated trials to vary between simulations from list [5, 10, 15, …, 100]. For the standard Rescorla-Wagner algorithm, we fixed learning rate to 0.3, for the decaying algorithm we fixed decay rate to t/10, and for the asymmetric algorithm we fixed learning rate for positive feedback to 0.8 and learning rate for negative feedback to 0.2. A total of 600 simulations were run across the three algorithms and 20 possible trial quantities, ten for each algorithm-trial pair. Recoverability under noise improved with the number of simulated trials (r=0.975, p<0.001), with both the proportion of algorithms recovered with correct form and the recovered algorithm's fit to the simulated data saturating at 50 trials per agent (Figure S3)

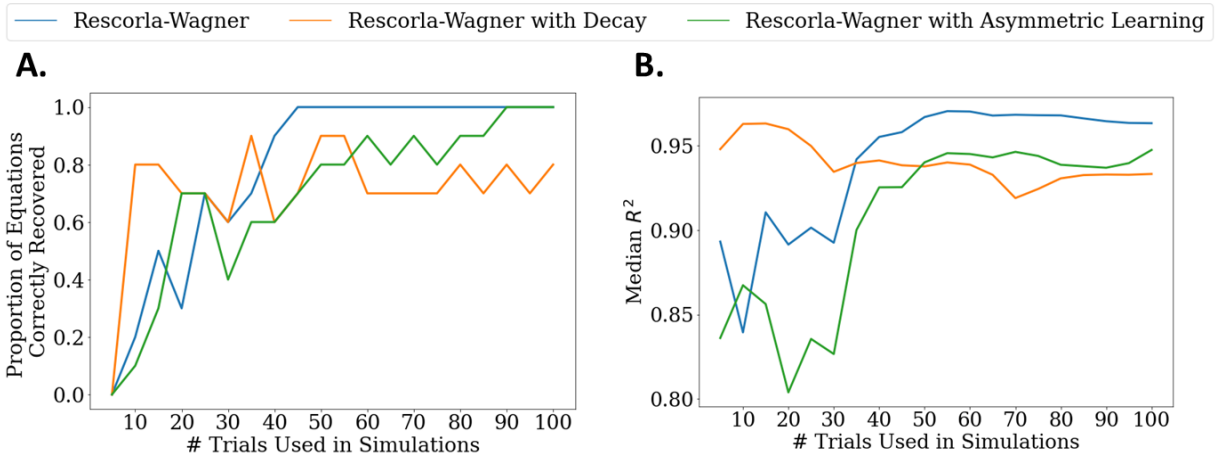

Figure S3. SINDy's robustness to noise with variation in number of trials**.** The x-axis represents the number of trials experienced by each agent in a simulation. The y-axis reflects a measure of robustness to noise, either the proportion of equations recovered in which the general form matched that of the governing equation (Panel A), or the median R^2 between simulated *Q* and SINDy recovered *Q* (Panel B).

Phase 1

Demographics

Table S1. Descriptive table of demographic variables for empirical Studies 1 and 2.

	Individual-level variables	***N***	**Percent**	**Mean**	**SD**
Study 1	Age	455		36.247	12.938
	Gender				
	Male	216	47.5		
	Female	216	47.5		
	Other	23	5		
	Race/ethnicity				
	Hispanic	6	1.3		
	White	356	78.2		
	Black	21	4.6		
	Asian	35	7.7		
	Other	41	9		
Study 2	Age	177		37.875	12.055
	Gender				
	Male	87	49.2		
	Female	85	48		
	Other	5	2.8		
	Race/ethnicity				

	Individual-level variables	***N***	**Percent**	**Mean**	**SD**
	Hispanic	16	9		
	White	135	76.3		
	Black	21	11.9		
	Asian	8	4.5		
	Other	3	1.7		

Visualization of Empirical Data

As an additional explorative measure of the Quadratic *Q*-Weighted model's explanatory validity, we plotted the median empirical trial-by-trial changes in expectation *Q* as a function of previous *Q* and previous reward. We observed that changes in *Q* were indeed a function of previous *Q* and reward, with the greatest changes being post-reward at low *Q*, and post-no reward at high *Q* (Figure S4). Furthermore, the change in *Q* appeared curvilinear across the *Q* spectrum. In both studies, receipt of reward had practically no effect on *Q* when *Q* was already greater than 0.5, and a lack of reward had practically no effect when *Q* was less than 0.5.

Plotting histograms of reported *Q*s between studies revealed *Q* was more under distributed in Study 2 where volatility was greater and initial reward rate was uniform (Figure S5). Although both distributions were inflated at 0 and 1 (i.e., participants tended to evaluate reward with absolute certainty), *Q*s were more normally distributed in Study 1, possibly due to the comparatively lower volatility and initial reward rate being fixed at 0.5.

Finally, we plotted reported *Q*s from trials against to their true generative probability for those trials (Figure S6). The S-shaped distribution of the reported *Q*s reflect the predictions of the *Quadratic Q-Weighted* model – underestimation where probability is high due to exponentially more negative changes in *Q* at high evaluations, and overestimation where probability is low due to near constant positive changes in *Q* at low evaluations.

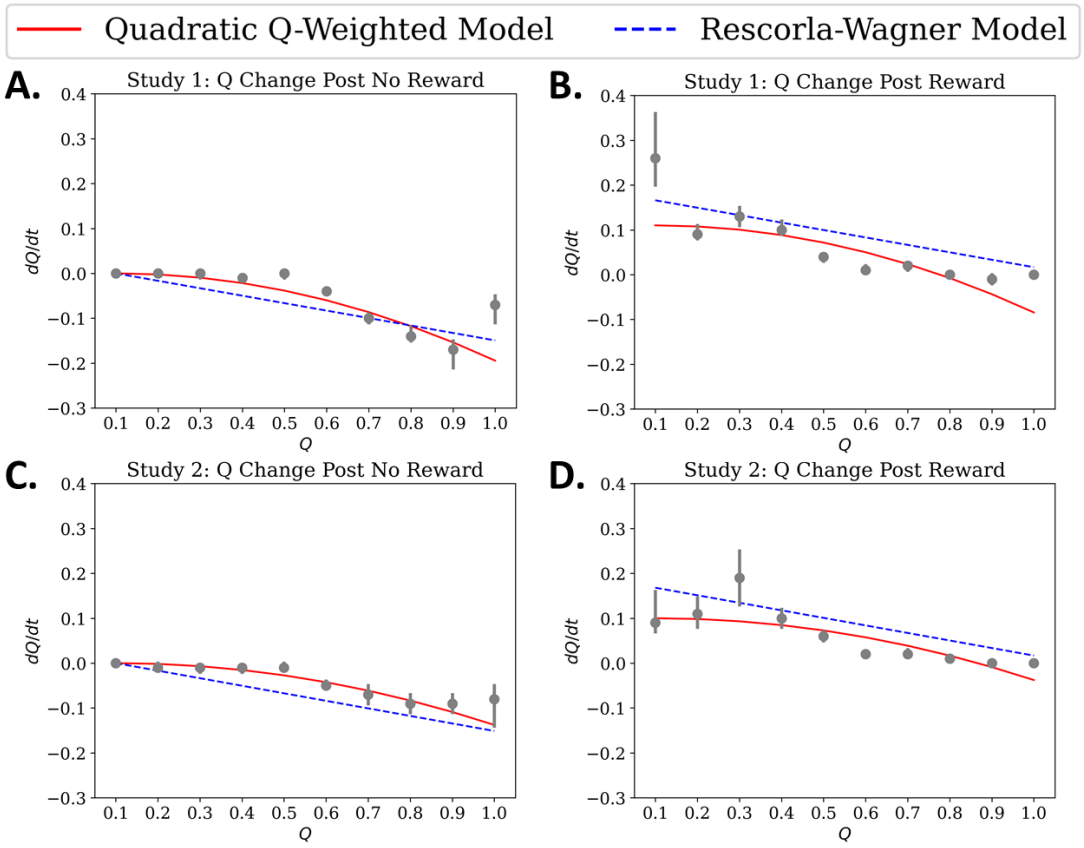

Figure S4. An overview of behavior of the Quadratic Q-Weighted model we discovered using empirical data with SINDy. The x-axes reflect reported Q value and the y-axes are the median change in value. Grey dots show binned Q into 10 discrete categories, each with a bin size of 0.1. Categories were labeled with the upper bound of each bin. Error bars are 95% confidence intervals. (A) Study 1 empirical change in Q following no reward. (B) Study 1 empirical change in Q following reward. (C) Study 2 empirical change in Q following no reward. (D) Study 2 empirical change in Q following reward. Predicted changes in Q according to the best fit Quadratic Q-Weighted model (solid red) and the best fit Rescorla-Wagner model (dashed blue) are overlaid.

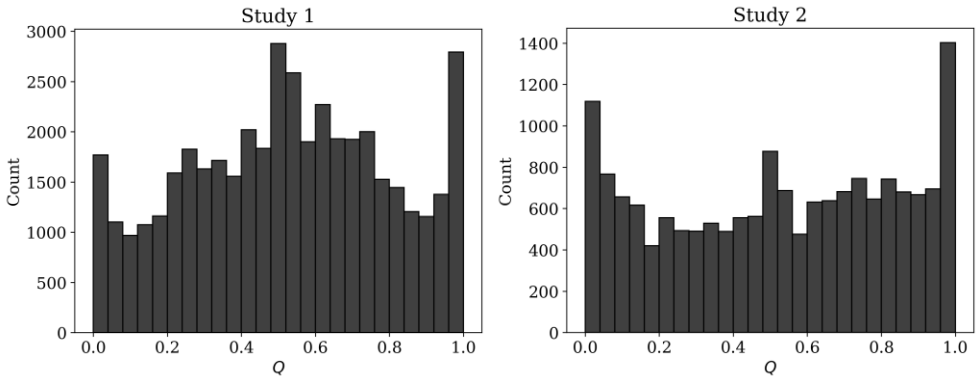

Figure S5. Histograms of Qs reported by all participants from Studies 1 and 2. Histograms reveal inflated evaluations of Q=0 and Q=1, and greater underdistribution in Study 2 than Study 1.

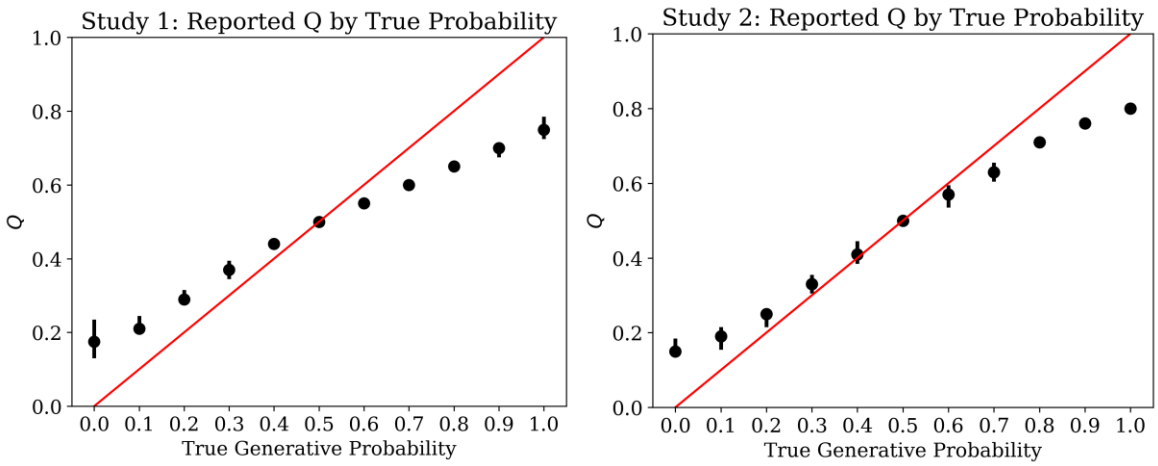

Figure S6. Plotting median reported Q against the true generative probability of a working phone. Dots show binned true generative probabilities into 10 discrete categories, each with a bin size of 0.1. Categories were labeled with the upper bound of each bin. Error bars are 95% confidence intervals.

Proof of Quadratic Q-Weighted Model Point of Under-to-Over Estimation

To approximately determine the reward value at which the transition from underestimation to overestimation occurs, we initiate by taking the time mean of the Quadratic Q-Weighted model's reward equation:

$$\overline{\dot{Q}_t} = a\bar{r}_t - b\overline{Q_t^2} \quad (1)$$

where $\bar{A}$ is the time average of A. Time mean of time derivative is zero ($\overline{\dot{Q}_t} = 0$; assuming negligible drift in the initial and end Q_t values). In order for our model to be unbiased, we stipulate that $\bar{r}_t = \overline{Q_t}$, leading to:

$$0 = a\overline{Q_t} - b\overline{Q_t^2} \quad (2)$$

We then break down Q_t into two components, where Q' represents the deviation from $\overline{Q_t}$:

$$0 = a\overline{Q_t} - b\overline{Q_t} * \overline{Q_t} - b\overline{Q'^2_t} \quad (3)$$

Assuming that $\overline{Q_t} * \overline{Q_t}$ is significantly greater than $\overline{Q'^2_t}$ (valid for non-small or non-large values of r), we simplify to:

$$0 \approx a\overline{Q_t} - b\overline{Q_t} * \overline{Q_t} \quad (4)$$

This simplification leads to the reward value at which our model is unbiased:

$$\bar{r}_t = \overline{Q_t} \approx a/b \quad (5)$$

Phase 1 Model Comparison

Four models were fit to the empirical data from Phase 1 to determine whether the quadratic Q-weighted model discovered by SINDy was a relatively better explanation of the cognitive processes responsible for the data. These additional competing models were the Rescorla-Wagner model, the Rescorla-Wagner model with time-dependent exponential decay, the Rescorla-Wagner model with asymmetric learning-rates, and a binary Kalman filtering model. These models differed in a variety of ways from the models used in Phase 0 simulation analyses. First, we acknowledged an inflated tendency in our Phase 1 raw data for extreme *Q* values to shift towards the center due to the response scale being bounded between 0 and 100 percent. We correct this artificial tendency by treating *true* *Q*-value (distinct from reported *Q*-values) as the hidden state in a hidden Markov model. Assuming that the true *Q*-value is the logit of the subjective probability of success:

$$Q_{subjective} = \frac{1}{1 + e^{-Q_{true}}}$$

Allowing the true *Q* to be unbounded and non-biased at $Q_{true} = 0$. The true *Q* is assumed to evolve as a function of feedback and subjective *Q* (so they're on the same 0-to-1 scale), the form of which depends on the specific updating rule.

Two additional free parameters were added to all models beyond those described in the main text: initial expectation Q_0 on the unbounded true *Q* scale, and expectation-to-response noise $\sigma_{Q\to r}$. Initial expectation Q_0 was the starting value of *Q* prior to learning and was the Q_t used to calculate Q_{t+1} in the delta update at trial t=1. Expectation-to-response noise $\sigma_{Q\to r}$ was the variability in responses sampled from a *Q*-centered Gaussian distribution, $N(Q, \sigma_{Q\to r})$. After each update to *Q* on each trial, participants were assumed to generate a response from this *Q*-centered Gaussian distribution.

In addition to considering the models included in Phase 0 simulations, we also included a binary Kalman filtering model as a candidate for comparison to test the limits of the Quadratic Q-weighted model's predictive accuracy. Kalman filtering models capture multiple latent processes that the SINDy algorithm as we implemented it is unable to model. Therefore, any model recovered by SINDy that performs on par with a Kalman filtering model is particularly impressive. The binary Kalman filtering model we used was developed by Piray and Daw (*6*), and assumes that learning rates are a function of one's learned uncertainty *w* and perceived volatility v. Learning rates take the form:

$$\alpha = \sqrt{w_{t-1} + v_{t-1}}$$

The updating rule for *Q* is then identical to that of the Rescorla-Wagner model, with a dynamically changing learning rate. Learned uncertainty and perceived volatility also evolve with the following updating rules:

$$w_{t+1} = (1 - \mathrm{K})(w_t + v_t)$$

$$w_{t,t+1} = (1 - \mathrm{K})w_t$$

$$v_{t+1} = v_t + \lambda[(Q_{t+1} - Q_t)^2 + w_t + w_{t+1} - 2w_{t,t+1} - v_t]$$

Where λ is the learning rate for volatility, $w_{t,t+1}$ is the autocovariance in uncertainty between trials, and K is the Kalman gain or learning rate for uncertainty. The Kalman gain is also a function of previous learned uncertainty and perceived volatility:

$$\mathrm{K} = \frac{w_t + v_t}{w_t + v_t + \omega}$$

Where ω is the measurement noise. This parameter is included entirely for inference and has no role in a generative, explanatory model (*6*).

Models were fit and penalized maximum likelihood estimates were produced for all model parameters with a quasi-Newton optimization algorithm, L-BFGS, which is the default optimizer available in the Stan probabilistic programming language (*7*). We used the following weakly-informative prior distributions:

All models:

$Q_0 \sim N(0, 0.1)$

$\sigma_{Q \to r} \sim \Gamma(1, 0.2)$

Rescorla-Wagner model (3*N total free parameters):

$\alpha \sim TN(0.2, 0.1, 0, 1)$

Rescorla-Wagner model with time-dependent exponential decay (3*N total free parameters):

$\lambda \sim TN(15, 10, 0, \infty)$

Rescorla-Wagner model with asymmetric learning rates (4*N total free parameters):

$\alpha^+ \sim TN(0.2, 0.1, 0, 1)$

$\alpha^- \sim TN(0.2, 0.1, 0, 1)$

Binary Kalman filtering model (5*N total free parameters):

$v_0 \sim TN(1, 0.1, 0, 10)$

$\lambda \sim TN(0.2, 0.1, 0, 1)$

$\omega \sim \Gamma(1, 1)$

Quadratic Q-weighted model (4*N total free parameters):

$a \sim TN(0.2, 0.1, 0, 1)$

$b - a \sim \Gamma(1, 0.2)$

The $b - a$ parameter from the Quadratic Q-weighted model was used to ensure that b, the coefficient scaling Q^2 in the model, was both deterministic and greater than or equal to a: $b = (b - a) + a$. Last, the v_0 parameter included in the binary Kalman filtering model was the starting value of *v*, similar to how Q_0 is used to initialize *Q* for all models.

We compared the relative fits of the five models using the Bayesian Information Criterion (BIC), which penalizes more complex models for the number of free parameters they

include. BICs were calculated separately for each participant and model. We compared each participant's BIC for the Quadratic Q-weighted model with their BIC for the other candidate models. Figure S7 illustrates the mean differences in BIC between the Quadratic Q-weighted model and the other candidates. Importantly, a more negative BIC value reflects better fit, and so a more negative difference suggests a superior fit for the Quadratic Q-weighted model over the alternative. Our results suggest that the Quadratic Q-weighted model provides a superior fit on average when using standard fitting procedures currently used in computational social science, supporting our findings with SINDy.

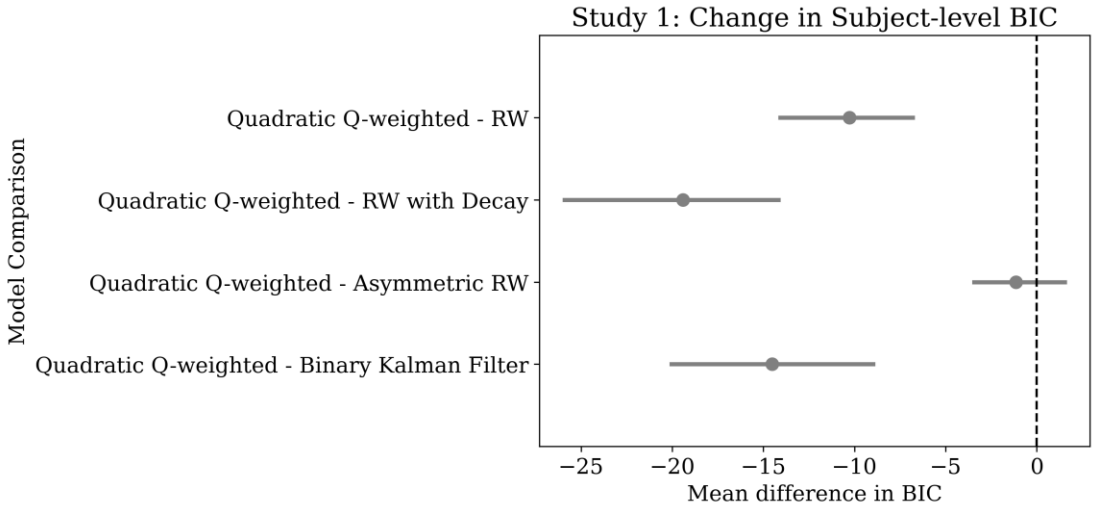

Figure S7. Comparisons in subject-level fit between the Quadratic Q-weighted model and other candidate models. The x-axis reflects the mean subject-level difference in BIC. Error bars are 95% confidence intervals. A more negative difference in BIC reflects superior fit of the Quadratic Q-weighted model over the alternative.

To ensure that these results were not explained in part by overfitting, we further conducted a cross-validation procedure with BICs. Specifically, we fit each model to the last 50 trials of each participant and then evaluated the likelihood of their held-out first using the fitted parameter values. BICs were substantially greater but relatively similar in comparison to our previous analysis (see Figure S8). Although the asymmetric RW model outperformed the QQW model when compared against each other, the asymmetric RW was the winning model amongst the five evaluated for only 5.93% of participants. The QQW model was the winning model for 28.13% of the participants, the highest percentage amongst the candidates.

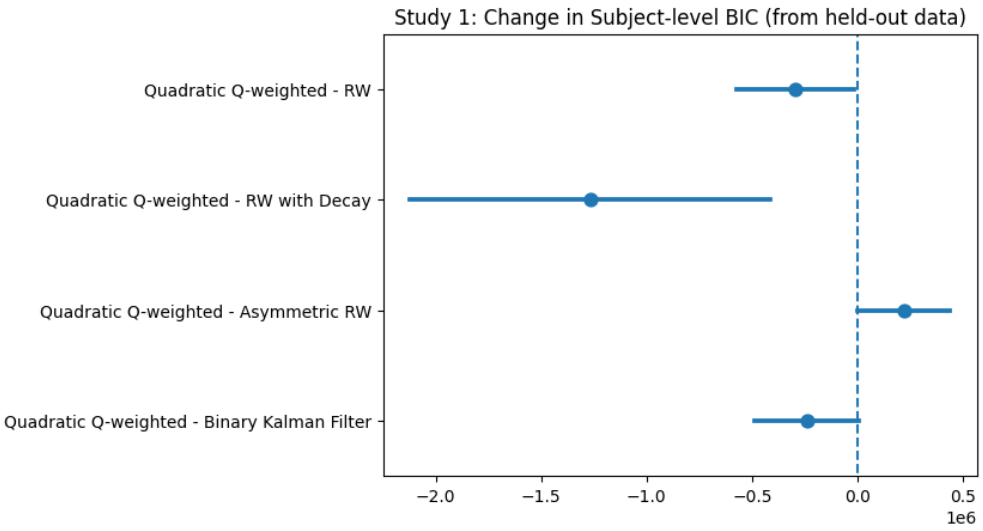

Figure S8. Comparisons in subject-level fit to the first 50 trials of each participant between the Quadratic Q-weighted model and other candidate models, after training on the last 50 trials. The x-axis reflects the mean subject-level difference in BIC. Error bars are SE. A more negative difference in BIC reflects superior fit of the Quadratic Q-weighted model over the alternative.

Recoverability of the Quadratic Q-Weighted Model

To ensure the validity of our approach, we extended our model recovery simulations to include the Quadratic Q-Weighted (QQW) model. These extended simulations demonstrate that the QQW model is reliably recovered when it generates the data, but it is not recovered when data are generated by other functional forms. This indicates that the QQW model is not merely an attractor in model space. Importantly, the Q^2 term was included as a candidate function in all simulation and recovery analyses described in **SINDy Recovers the Ground-Truth Learning Algorithms**. The inclusion of the Q^2 term does not alter the recovered forms of the Rescorla-Wagner (RW) model, RW with time-decay, or RW with asymmetric learning rates. Furthermore, the Q^2 term is not discovered when analyzing synthetic data generated by any of these three models, even across a wide spectrum of learning and decay rates, as detailed in **Model Recovery as a Function of Initial Conditions**.

This highlights a key advantage of the QQW model: it does not suffer from the misidentification issues observed among the RW models. For example, under certain conditions, the RW model and the RW model with asymmetric learning rates are not distinguishable (Figure S2, Panel C). In contrast, the QQW model remains identifiable, as evidenced by Figure S9. Specifically, when data are generated from the QQW model, the best-fitting RW model cannot account for the observed data.

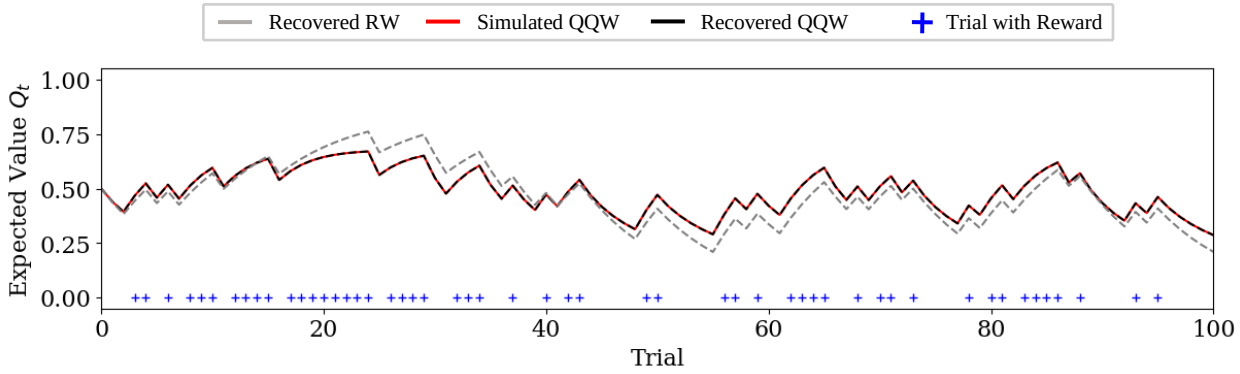

Figure S9. Representative example of the QQW model's fit to QQW agents' data compared to the best fit RW model. The x-axis represents trial number. The y-axis represents the expected value Q after each trial. The red line is the ground-truth Q value as produced by the agent in the simulation. The black dotted line is the recovered Q value using the equation recovered by SINDy. The grey dotted line represents the predicted Q values from the best fit RW model. Notably, the black dotted line (recovered Q) sits perfectly on top of the red line (ground truth Q) indicating perfect model recovery, whereas the RW model deviates from the truth where Q approaches values further from 0.5.

To further examine the robustness of the QQW model, we conducted additional group-level analyses using only the last 50 trials from each participant, per our previous observation that 50 trials was sufficient for model recoverability with SINDy in **Robustness to Noise Analyses**. This subsampling approach allowed us to test whether the QQW model could still be reliably identified with reduced data. The results show that the functional form of the identified QQW model remains unchanged, though the coefficients and model fit exhibit slight variations. For Study 1, SINDy identified the model as $0.11r - 0.24Q^2$ ($R^2 = 0.204$) when using all 100 trials,

compared to $0.11r - 0.23Q^2$ ($R^2 = 0.190$) with the last 50 trials. Similarly, for Study 2, the model identified using all 100 trials was $0.10r - 0.17Q^2$ ($R^2 = 0.196$), while using the last 50 trials resulted in $0.11r - 0.18Q^2$ ($R^2 = 0.187$). These results demonstrate that the QQW model remains stable and identifiable even with a reduced number of trials, underscoring its robustness.

At the individual level, we conducted complementary analyses using Stan to evaluate model fits for each participant. When limiting the Stan fitting to the last 50 trials, the QQW model was identified as the best-fitting model for 34.72% of participants, outperforming the next best model, RW with time decay, which accounted for 21.75% of participants. Moreover, the QQW model provided a better fit than the classic RW model for 74% of participants. These results are illustrated in Figure S10, and reaffirm that the QQW model's superior fit is consistent across both full and reduced trial datasets.

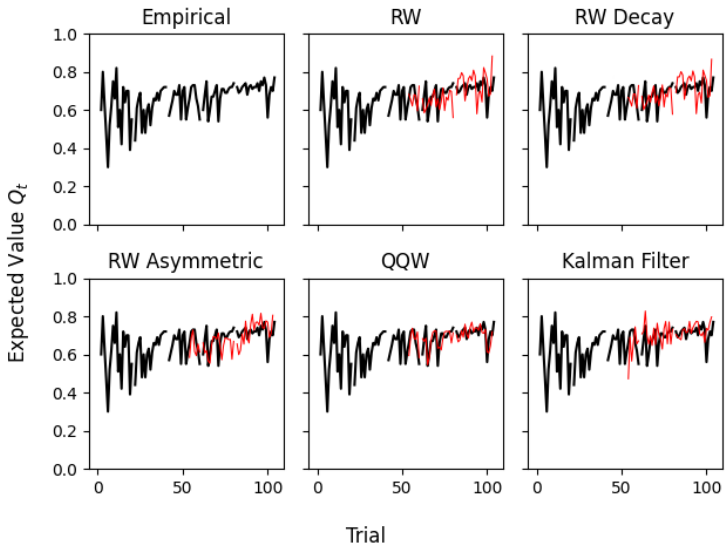

Figure S10. Expected value estimates (Q) over the last 50 trials for a single representative participant, with all models fitted exclusively to these 50 trials. The "Empirical" panel represents the participant's reported values (black lines), while the remaining panels show predictions (red lines) from the models: RW (Rescorla-Wagner), RW with exponential decay, RW with asymmetric learning rates, QQW (Quadratic Q-Weighted), and the Kalman Filter. Missing data correspond to trials where attention checks were administered. Despite being fitted on a limited dataset of only 50 trials, all models capture the general trend of the observed data. Notably, the QQW model provides superior predictions, demonstrating that 50 trials are sufficient for robust model identification and comparison.

Phase 2

Phase 2 Model Comparison

As described in the main text, datasets for Phase 2 were curated using the Niv Lab OpenData repository. Original analysis scripts were obtained from the identified studies (See Table S2). All analysis scripts were executed as-is in Matlab to calculate original BIC values for the authors' models. We then modified those scripts to replace any instances of Rescorla-Wagner delta updating rules with the Quadratic Q-Weighted model instead. In this section, we will describe the most critical modifications made to each script. Some additional changes were made not reported in this document, primarily to set up new variables and supporting architecture. See our Github for the exact modified scripts. The following changes were made to the analysis scripts accompanying the Kool et al., 2017 paper (*8*), where they coded:

```
dtQ(1) = Qd(state2,subdata.choice2(t)) - Qd(1,subdata.choice1(t));
% backup with actual choice (i.e., sarsa)
    Qd(1,subdata.choice1(t)) = Qd(1,subdata.choice1(t)) + lr*dtQ(1);
% update TD value function

    dtQ(2) = subdata.win(t) - Qd(state2,subdata.choice2(t));
% prediction error (2nd choice)

    Qd(state2,subdata.choice2(t)) = Qd(state2,subdata.choice2(t)) + lr*dtQ(2);
% update TD value function
    Qd(1,subdata.choice1(t)) = Qd(1,subdata.choice1(t)) + lambda*lr*dtQ(2);
```

This code reflects a two-stage model in which participants take two actions. First, they act on expectations of State 1 transitioning to a desired State 2, and then act on expectations of reward at State 2. dtQ(1) is the difference in the value of the State 2 they transitioned to and where they expected to transition to following their first stage choice. dtQ(2) is the difference in the value of the reward they receive and their expectation of reward at State 2. Participants use these prediction errors to update their expectations at both stages, proportional to a learning rate lr. After feedback at State 2 is received, participants also update their State 1 expectations proportional to that feedback scaled by both a learning rate lr and an eligibility trace parameter lambda. Importantly, each of these three updates are Rescorla-Wagner delta updating rules. We replaced these updates with:

```
dtQ(1) = rc*Qd(state2,subdata.choice2(t)) - qc*(Qd(1,subdata.choice1(t))^2);
% backup with actual choice (i.e., sarsa)
```

```
    Qd(1,subdata.choice1(t)) = Qd(1,subdata.choice1(t)) + dtQ(1);
% update TD value function

    dtQ(2) = rc*subdata.win(t) - qc*(Qd(state2,subdata.choice2(t))^2);
% prediction error (2nd choice)

    Qd(state2,subdata.choice2(t)) = Qd(state2,subdata.choice2(t)) + dtQ(2);
% update TD value function
    Qd(1,subdata.choice1(t)) = Qd(1,subdata.choice1(t)) + lambda*dtQ(2);
% eligibility trace
```

This code replaced the three delta updating rules with Quadratic Q-Weighted models, with different coefficients scaling the effects of reward/outcome and of previous expectation.

The following changes were made to the analysis scripts accompanying the Lefebvre et al., 2017 (*9*), Palminteri et al., 2017 (*10*), and Chambon et al., 2020 (*11*) papers, where they coded:

```
PEc =  rew - Q(s(i),a(i));

Q(s(i),a(i)) = Q(s(i),a(i)) + lr1 * PEc * (PEc>0) +  lr2 * PEc * (PEc<0);

C(s(i),a(i)) = C(s(i),a(i)) + tau * (1 - C(s(i),a(i))); % increasing the chosen
option choice trace

C(s(i),3-a(i)) = C(s(i),3-a(i)) + tau * (0 - C(s(i),3-a(i))); % decreasing the
unchosen option choice trace
```

These papers used a two-armed bandit task where participants needed to anticipate the probability of reward from two options and act on their expectations. PEc is the difference in the value of reward received after taking action a(i) at state s(i) and the participant's expectation of reward for action a(i) at state s(i). Participants then use this prediction error to update their expectations proportional to a learning rate. There are asymmetric learning rates for positive and negative prediction errors, with lr1 scaling positive prediction errors, and lr2 scaling negative prediction errors. Participants also maintain a choice trace, or a tendency to perseverate on past choices. When an option is chosen, it is "rewarded" and its "choice prediction error" is conceptualized as the difference between its reward and the expectation of choosing that option. Choice traces are updated with this prediction error proportional to a choice trace accumulation rate tau. Importantly, both of these expectation and trace updates are Rescorla-Wagner delta updating rules. We replaced these updates with:

```
PEc =  rew_coef*rew - qval_coef*(Q(s(i),a(i))^2);

Q(s(i),a(i)) = Q(s(i),a(i)) + PEc * (PEc>0) +  PEc * (PEc<0);

C(s(i),a(i)) = C(s(i),a(i)) + rew_tau - qval_tau*(C(s(i),a(i))^2); % increasing the
chosen option choice trace

C(s(i),3-a(i)) = C(s(i),3-a(i)) - qval_tau*(C(s(i),3-a(i))^2); % decreasing the
unchosen option choice trace
```

This replaced an asymmetric learning model and a RW-like model of choice traces with Quadratic Q-Weighted models.

The following changes were made to the analysis scripts accompanying the Decker et al., 2016 (*12*), Potter et al., 2017 (*13*), and Nussenbaum et al., 2020 (*14*) papers, where they coded:

```
% State prediction error: Difference between the Q value from the first stage choice
and the updated Q of the ultimate second stage choice
    tdQ(1) = Qd(stateshort(i), choice2short(i)) - Qd(1,choice1short(i));

    %RPE - reward - second-stage value estimate
    tdQ(2) = moneyshort(i) - Qd(stateshort(i),choice2short(i));

    % MF update - update first-stage choice values based on state prediction error
and discounted RPE
    Qd(1,choice1short(i)) = Qd(1,choice1short(i)) + alpha * tdQ(1) + lambda * alpha *
tdQ(2);

    % MB update - directly to the Q value of the stage 2 stimulus
    Qd(stateshort(i),choice2short(i)) = Qd(stateshort(i),choice2short(i)) + alpha *
tdQ(2);
```

This code reflects a two-stage model that is identical to that used by Kool and colleagues (*8*). We replaced this with:

```
% State prediction error: Difference between the Q value from the first stage choice
and the updated Q of the ultimate second stage choice
    tdQ(1) = rew_coef*Qd(stateshort(i), choice2short(i)) -
qval_coef*(Qd(1,choice1short(i))^2);

    %RPE - reward - second-stage value estimate
    tdQ(2) = rew_coef*moneyshort(i) -
qval_coef*(Qd(stateshort(i),choice2short(i))^2);

    % MF update - update first-stage choice values based on state prediction error
and discounted RPE
    Qd(1,choice1short(i)) = Qd(1,choice1short(i)) + tdQ(1) + lambda * tdQ(2);
```

```
% MB update - directly to the Q value of the stage 2 stimulus
Qd(stateshort(i),choice2short(i)) = Qd(stateshort(i),choice2short(i)) + tdQ(2);
```

Similar to the modifications made to the Kool et al., 2017 analysis scripts, these changes replaced three delta updating rules with a Quadratic Q-Weighted model. After replacing all delta updating rules across all of the 9 curated studies, we re-ran the analysis scripts to calculate new BICs for our alternate models. Results from these alternate models are reported in the main text.

Table S2. Sources for Phase 2 data and analysis scripts.

Dataset	***Link***
Kool et al., 2017 Experiments 1 & 2	https://osf.io/yg82m/
Lefebvre et al., 2017 Experiments 1 & 2	https://github.com/spalminteri/conf-bias-meta-analysis
Palminteri et al., 2017 Experiment 1	https://github.com/spalminteri/conf-bias-meta-analysis
Chambon et al., 2020 Experiment 4	https://github.com/spalminteri/conf-bias-meta-analysis
Decker et al., 2016	https://osf.io/we89v/
Potter et al., 2017	https://osf.io/we89v/
Nussenbaum et al., 2020	https://osf.io/we89v/

Phase 2 Model Recovery

To assess the recoverability of the Quadratic Q-Weighted (QQW) model beyond our original task, we conducted a model recovery analysis using synthetic data generated from the QQW and Rescorla-Wagner (RW) variants of models of two distinct tasks: The Two-Step task and the Confirmation Bias task. For each model, we generated synthetic datasets using both the QQW and RW formulations and then fit both variants to the data, computing the Bayesian Information Criterion (BIC) for each fit. The difference in BIC (QQW BIC minus RW BIC)

serves as an index of model recoverability, where negative values indicate that the QQW model provides a better explanation of the data than the RW alternative. See recovery-analysis.ipynb in the Github repository for simulation code and model specifications. 100 synthetic datasets were simulated per model, per task, and 1000 trials were simulated within each of those synthetic datasets.

The results of this analysis are summarized in the Figure S11. For synthetic data generated from the QQW variant of the Two-Step task, the QQW model was correctly recovered, as evidenced by a negative BIC difference—indicating superior model fit relative to the RW model. However, in the Confirmation Bias task, while the QQW model still outperformed the RW model on QQW-generated data, the BIC difference was smaller, suggesting weaker—but still present—recoverability. In contrast, for datasets generated from the RW variants of each model, the RW model was consistently favored, with substantial positive BIC differences across both tasks. These results confirm that the RW model is reliably recovered when it is the true data-generating process.

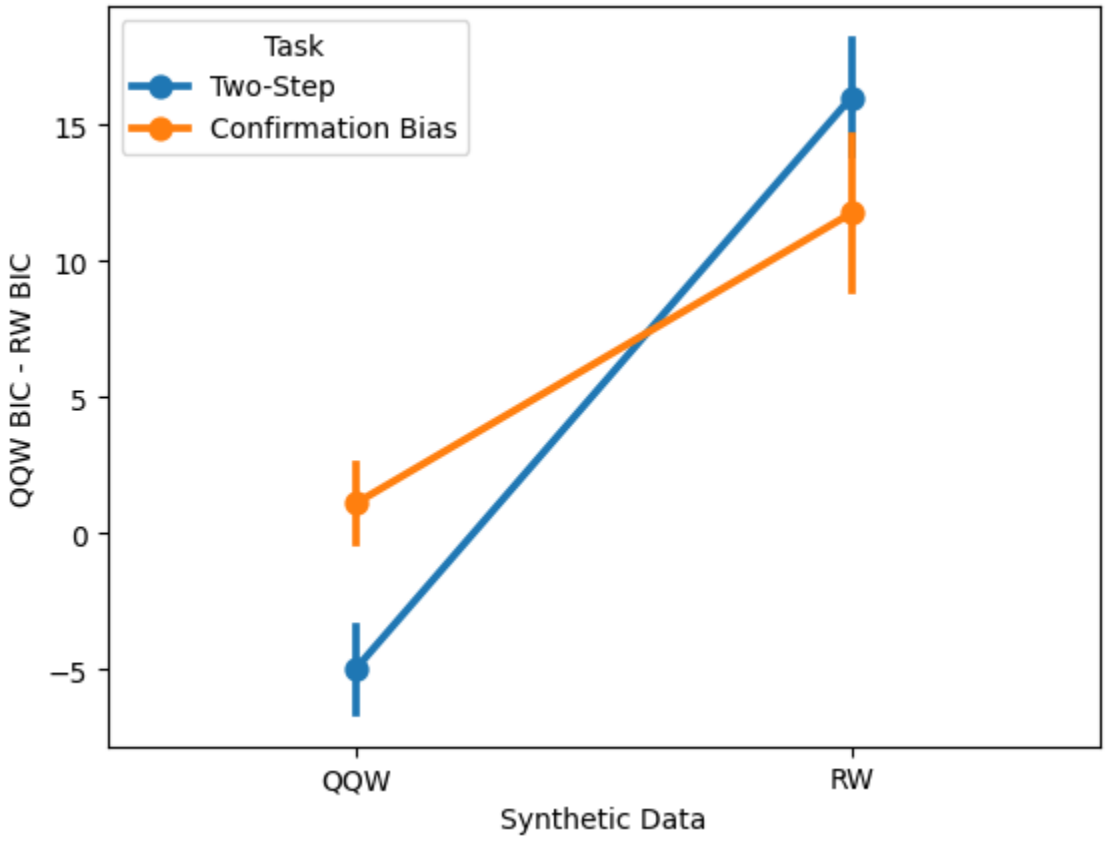

Figure S11. Model recovery results comparing BIC differences between QQW and RW model fits across synthetic datasets. Negative values indicate that QQW better explains data generated from the QQW model, while positive values indicate that RW better explains data generated from the RW model. Results are shown separately for the Two-Step task (blue) and the Confirmation Bias task (orange), with error bars representing ±1 SEM across synthetic participants.

We extended our model recovery analysis by examining whether specific generative parameter values influence the ability to distinguish the QQW model from its Rescorla-Wagner (RW) counterpart. The central question guiding this analysis was whether QQW-generated synthetic data becomes harder to distinguish from RW-generated data when participants behave more randomly or noisily—that is, when parameters such as the inverse temperature (β) are low, weakening the influence of QQW-derived expectations on choice.

To explore this, we computed Pearson correlations between each generative parameter and the difference in BIC (QQW BIC minus RW BIC). Negative correlations suggest that higher parameter values improve QQW recoverability, whereas positive correlations suggest that higher values may reduce it.

For models of the Two-Step task, several parameters meaningfully predicted model distinguishability. Most notably, the reward scaling parameter a—which amplifies the influence of received outcomes in the QQW update—was strongly negatively correlated with BIC differences ($r = –0.41$, $p < .001$), indicating that stronger reward-driven updating enhances the detectability of QQW. The inverse temperature for first-stage choices, $\beta 1$, was also negatively associated with BIC differences ($r = –0.20$, $p = .044$), suggesting that more deterministic choice behavior facilitates QQW recovery. In contrast, the second-stage inverse temperature $\beta 2$ was not predictive. The coefficient b, which scales the influence of the prior Q-value squared term in the QQW equation, also predicted lower QQW BIC ($r = –0.25$, $p = .013$). Finally, the perseveration parameter p, indexing choice stickiness, was not significantly associated with model recovery, suggesting that response repetition alone does not obscure or enhance QQW-specific learning patterns on the Two-Step task. See Table S3.

In the models of the Confirmation Bias task, a somewhat different set of predictors emerged. The difference between the inverse temperature and the confirmatory scaling parameter ($\beta - \varphi$) was negatively correlated with BIC differences ($r = -0.28$, $p = .005$). Importantly, this means that QQW was more recoverable when confirmation bias was weaker relative to deterministic choice—i.e., when φ was smaller than β. A similar pattern was evident in the confirmation bias learning rate τ, which showed a significant negative correlation ($r = -0.22$, $p = .030$). Higher values of φ alone were associated with reduced QQW recoverability ($r = 0.22$, $p = .032$), likely because they promote confirmatory learning dynamics that mask QQW learning. Other parameters, including the QQW reward scaling coefficient a and expectation scaling coefficient b, did not significantly impact model recoverability. See Table S4.

Together, these findings suggest that QQW recoverability depends not only on the data-generating model, but also on the behavioral regime induced by generative parameter settings. When participants behave in ways that amplify the influence of reward-based updating (high a) and make choices more deterministically (high β, low φ), the QQW model becomes more identifiable. In contrast, when behavior is dominated by confirmatory mechanisms (high φ), the distinctions between models blur. These insights reinforce that QQW captures a recognizable computational signature—but that this signature is only recoverable under parameter regimes that expose its unique structure.

Table S3. Correlations Between Generative Parameters and BIC Differences (Two-Step Task)

Parameter	**Pearson *r***	***p*-value**
a	–0.41	< .001
β1	–0.20	.044
β2	–0.08	.445
w	0.22	.028
p	0.08	.432
λ	–0.09	.382
b	–0.25	.013

Table S4. Correlations Between Generative Parameters and BIC Differences (Confirmation Bias Task)

Parameter	Pearson *r*	*p*-value
β – φ	–0.28	.005
a	0.001	.992
τ	–0.22	.030
β	–0.19	.063
φ	0.22	.032
b	–0.03	.733

Phase 2 Parameter Recovery

To ensure the interpretability and robustness of the QQW model, we conducted a parameter recovery analysis by fitting both QQW and RW variants to synthetic datasets generated using the QQW model of the Two-Step task. These plots and correlations quantify the degree to which known generative parameters are recaptured during estimation, a necessary validation step in any modeling framework.

For the models of the Two-Step task, the results demonstrate strong parameter recoverability for key QQW-specific components (see Table S5 and Figure S12). The QQW reward scaling parameter a, central to the model's nonlinear learning rule, was recovered with high fidelity (r = 0.84), confirming that the model reliably captures the influence of recent outcomes on value updating. The prior-expectation scaling parameter b was also recoverable, though to a more modest degree (r = 0.31), suggesting that while its effects are detectable, they are more entangled with noise or other parameters.

First- and second-stage inverse temperatures ($\beta 1$ and $\beta 2$) were also well recovered. Particularly, $\beta 2$ showed near ceiling-level recovery (r = 0.87), supporting the idea that stochasticity in second-stage choices is a well-estimated and behaviorally distinct feature. Interestingly, the RW model also captured $\beta 2$ effectively, indicating that this parameter may not strongly differentiate the two models. Conversely, $\beta 1$ showed better recovery in QQW than in RW (r = 0.61 vs. r = 0.47). The model-based weight w and eligibility trace λ showed moderate recovery (r = 0.46 and 0.44, respectively), with more noise around extreme values. This likely

reflects ceiling and floor effects commonly seen in bounded parameters during recovery, compounded by interaction with reward scaling. Finally, the stickiness parameter p was nearly perfectly recovered (r = 0.998), underscoring its strong and separable influence on choice behavior independent of the learning process.

We assessed parameter recovery in the Confirmation Bias task by simulating data from the QQW variant and re-fitting both QQW and RW models (see Table S6 and Figure S13). The confirmation bias weighting parameter ϕ was recovered with high accuracy (r = 0.93), demonstrating that QQW can reliably capture asymmetries in how confirmatory versus disconfirmatory evidence shapes belief updating. The learning rate τ, which scales the magnitude of confirmation bias updating, also showed strong recoverability (r = 0.80). Inverse temperature β was moderately recovered (r = 0.59) and showed nearly identical recovery patterns in both the QQW and RW models.

The QQW-specific reward scaling coefficient a was only modestly recovered (r = 0.36), and many fitted values clustered at the ceiling, suggesting saturation or overfitting at the high end of the range. Finally, the prior-expectation weighting term b, which enhances the role of previously learned value expectations via a Q^2 term, showed poor recoverability (r = 0.18). This may reflect its relatively subtle influence on choice or potential redundancy with φ in modulating asymmetry.

In summary, the QQW model demonstrates solid recoverability of its core psychological parameters—particularly φ and τ—within confirmation bias tasks. These results validate the model's usefulness for probing asymmetric belief updating mechanisms, even if QQW-specific scaling terms like a and b are more difficult to estimate precisely.

Table S5. Two Step Task QQW parameter recovery

Parameter	Description	Recovery r	Notes
a	Reward scaling in QQW update	0.84	Strong, consistent recovery
b	Prior Q^2 scaling in QQW update	0.31	Moderate recovery
β1	First-stage inverse temperature	0.61	Moderate recovery, better than fitted RW (r = 0.47)
β2	Second-stage inverse temperature	0.87	Very strong; near parity with fitted RW (r = 0.88)
w	Model-based weight	0.46	Modest recovery, with noise at extremes
λ	Eligibility trace	0.44	Moderate recovery, limited precision at boundaries
ρ	Choice stickiness	1.00	Near-perfect; recovers linearly across range

Table S6. Confirmation Bias Task QQW parameter recovery

Parameter	Description	Recovery r	Notes
a	Reward scaling in QQW update	0.36	Modest recovery; ceiling effects present
β	Inverse temperature	0.59	Moderate recovery; similar in QQW and RW fits
ϕ	Confirmation bias weight	0.93	Very strong recovery; QQW-sensitive
τ	Learning rate for confirmation bias	0.80	Strong, task-relevant learning parameter
b	Q^2 prior weight in QQW update	0.18	Weak recovery; parameter may play subtle role

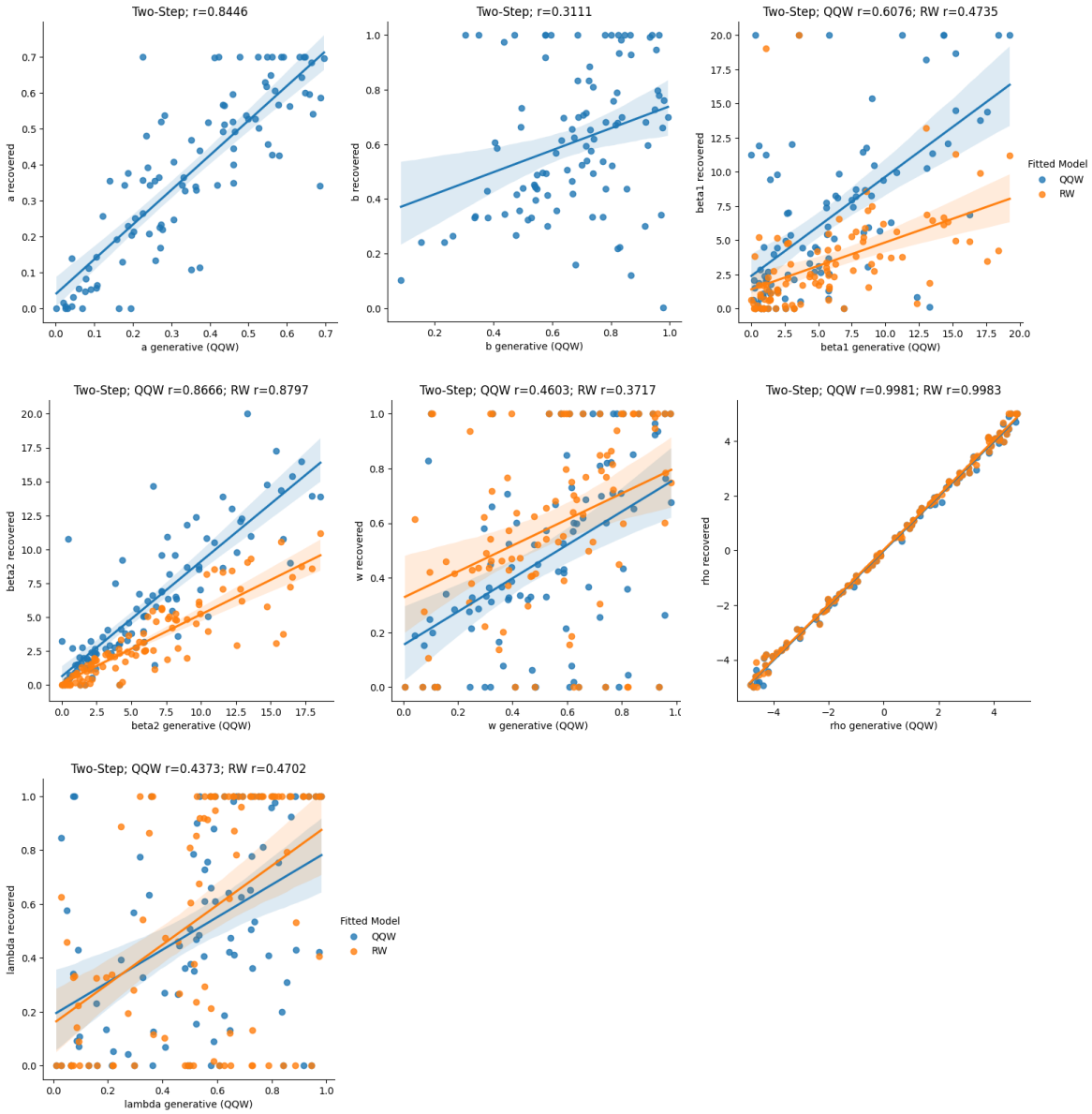

Figure S12. Parameter recovery for the Two-Step task using synthetic data generated from the QQW model. Each panel plots recovered parameter estimates against their generative values. Blue points represent fits from the QQW model; orange points show fits from the RW model (where applicable). Parameters include QQW-specific components such as reward scaling (a) and prior Q^2 weighting (b), as well as shared components like inverse temperatures ($\beta 1$, $\beta 2$), model-based weight (w), eligibility trace (λ), and perseveration (ρ). Most parameters show strong to moderate recovery, with particularly high recoverability, supporting the interpretability of QQW estimates in this task.

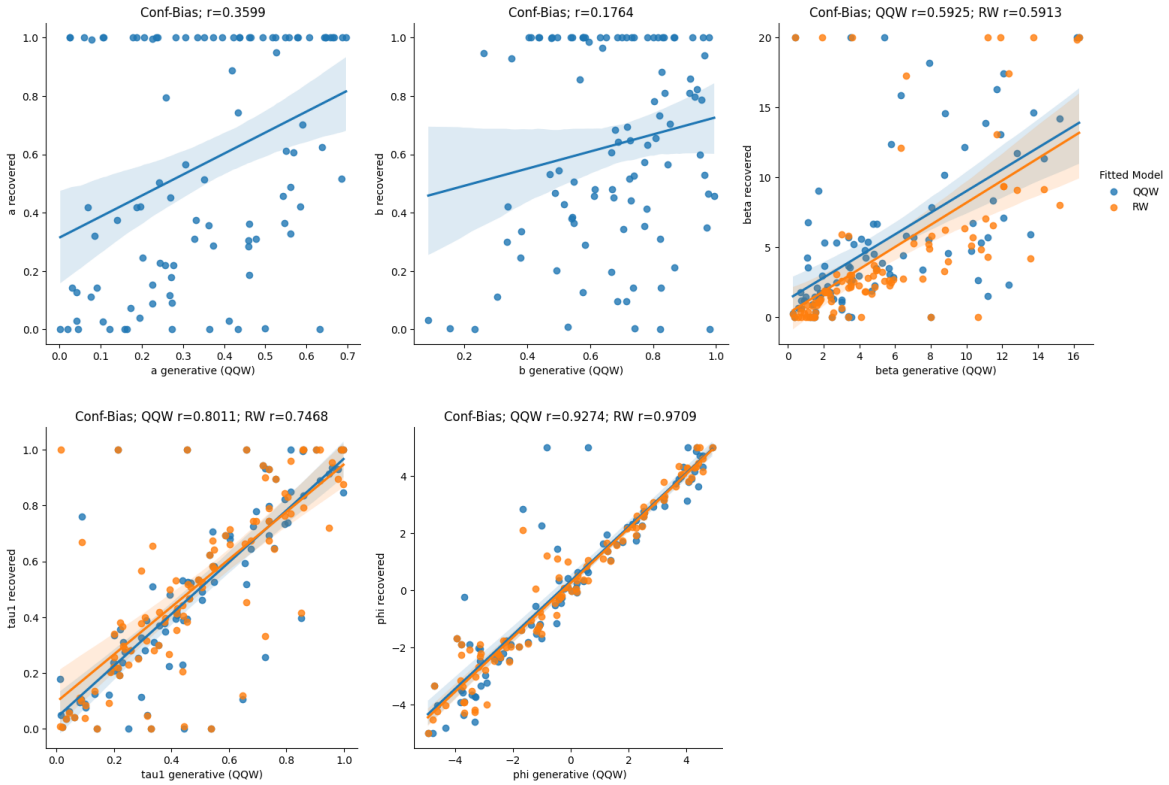

Figure S13. Parameter recovery for the Confirmation Bias task using synthetic data generated from the QQW model. Each panel shows recovered parameter estimates as a function of their true generative values. Blue points represent fits from the QQW model; orange points represent fits from the RW model. Parameters include QQW-specific terms such as reward scaling (a) and prior Q^2 weighting (b), as well as key decision-making and belief-updating parameters: inverse temperature (β), learning rate for confirmation bias updating (τ), and confirmation weighting (ϕ).

References

1. R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction* (MIT press, Boston, MS, 1998).
2. N. D. Daw, S. Kakade, P. Dayan, Opponent interactions between serotonin and dopamine. *Neural Networks* **15**, 603–616 (2002).
3. M. J. Frank, A. A. Moustafa, H. M. Haughey, T. Curran, K. E. Hutchison, Genetic triple dissociation reveals multiple roles for dopamine in reinforcement learning. *Proceedings of the National Academy of Sciences* **104**, 16311–16316 (2007).
4. Y. Niv, J. A. Edlund, P. Dayan, J. P. O'Doherty, Neural prediction errors reveal a risk-sensitive reinforcement-learning process in the human brain. *J. Neurosci.* **32**, 551–562 (2012).
5. M. J. Frank, B. B. Doll, J. Oas-Terpstra, F. Moreno, Prefrontal and striatal dopaminergic genes predict individual differences in exploration and exploitation. *Nat Neurosci* **12**, 1062–1068 (2009).
6. P. Piray, N. D. Daw, A simple model for learning in volatile environments. *PLoS Comput Biol* **16**, e1007963 (2020).
7. B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, A. Riddell, *Stan* : A Probabilistic Programming Language. *J. Stat. Soft.* **76** (2017).
8. W. Kool, S. J. Gershman, F. A. Cushman, Cost-Benefit Arbitration Between Multiple Reinforcement-Learning Systems. *Psychol Sci* **28**, 1321–1333 (2017).
9. G. Lefebvre, M. Lebreton, F. Meyniel, S. Bourgeois-Gironde, S. Palminteri, Behavioural and neural characterization of optimistic reinforcement learning. *Nat Hum Behav* **1**, 0067 (2017).
10. S. Palminteri, G. Lefebvre, E. J. Kilford, S.-J. Blakemore, Confirmation bias in human reinforcement learning: Evidence from counterfactual feedback processing. *PLoS Comput Biol* **13**, e1005684 (2017).
11. V. Chambon, H. Théro, M. Vidal, H. Vandendriessche, P. Haggard, S. Palminteri, Information about action outcomes differentially affects learning from self-determined versus imposed choices. *Nat Hum Behav* **4**, 1067–1079 (2020).

12. J. H. Decker, A. R. Otto, N. D. Daw, C. A. Hartley, From Creatures of Habit to Goal-Directed Learners: Tracking the Developmental Emergence of Model-Based Reinforcement Learning. *Psychol Sci* **27**, 848–858 (2016).
13. T. C. S. Potter, N. V. Bryce, C. A. Hartley, Cognitive components underpinning the development of model-based learning. *Developmental Cognitive Neuroscience* **25**, 272–280 (2017).
14. K. Nussenbaum, M. Scheuplein, C. V. Phaneuf, M. D. Evans, C. A. Hartley, Moving Developmental Research Online: Comparing In-Lab and Web-Based Studies of Model-Based Reinforcement Learning. *Collabra: Psychology* **6**, 17213 (2020).