nature human behaviour

Article

https://doi.org/10.1038/s41562-025-02324-0

Hybrid neural–cognitive models reveal how memory shapes human reward learning

Received: 4 October 2024

Accepted: 19 September 2025

Published online: 05 February 2026

Check for updates

Maria K. Eckstein[1], Christopher Summerfield[2], Nathaniel D. Daw[1,3] & Kevin J. Miller[1,4]

A long-standing challenge for psychology and neuroscience is to understand the transformations by which past experiences shape future behaviour. Reward-guided learning is typically modelled using simple reinforcement learning (RL) algorithms. In RL, a handful of incrementally updated internal variables both summarize past rewards and drive future choice. Here we describe work that questions the assumptions of many RL models. We adopt a hybrid modelling approach that integrates artificial neural networks into interpretable cognitive architectures, estimating a maximally general form for each algorithmic component and systematically evaluating its necessity and sufficiency. Applying this method to a large dataset of human reward-learning behaviour, we show that successful models require independent and flexible memory variables that can track rich representations of the past. Using a modelling approach that combines predictive accuracy and interpretability, these results call into question an entire class of popular RL models based on incremental updating of scalar reward predictions.

Reward-guided decisions are widely assumed to depend on a small number of latent variables that concisely summarize the history of actions and rewards and are calculated using simple incremental updates after each experience. For example, within the framework of reinforcement learning (RL), standard cognitive models posit that choices are based on '*Q*-values', which approximate the expected reward associated with each action and are calculated by repeatedly applying an incremental learning rule that compares the actual outcome to its previous estimate[1,2]. Such models are often simply called 'RL models', and they form the foundation for many studies investigating the psychology and neuroscience of reward-guided learning. These models have achieved an impressive record of success, providing computational explanations for basic as well as complex learning phenomena[3–9] and for neural correlates of reward-guided learning in a variety of tasks and species[10–12].

However, the literature has also accumulated a number of observations that these models do not easily account for. For example, individual events in the past can disproportionately affect behaviour[13–17], suggesting that task-relevant memory contains more than *Q*-value-like summary statistics of the reward history. Additionally, behaviour is often sensitive to global statistics of the past (for example, the range of rewards or the grouping of choice options) that are not easily captured by standard RL models[18–21]. Lastly, neural signals previously thought to relate straightforwardly to *Q*-values have been found to show marked diversity that is in tension with standard RL models[22–26]. These findings collectively suggest that the memory representations that humans and animals use to make reward-based choices go beyond incrementally learned summary statistics and may rely on a variety of additional internal memory mechanisms. However, a coherent computational account of such a learning algorithm is lacking.

Artificial neural networks (ANNs) are able to model highly expressive functions[27]. Sequential tasks can be modelled using recurrent neural networks (RNNs), which can learn to represent the past using high-dimensional internal states; these states are derived by memory mechanisms that are implemented in a potentially large number of trainable network parameters. With the ability to learn complex, time-dependent mapping functions, RNNs seem able to capture both the

[1]Google DeepMind, London, UK. [2]Department of Experimental Psychology, University of Oxford, Oxford, UK. [3]Princeton Neuroscience Institute and Department of Psychology, Princeton University, Princeton, NJ, USA. [4]Sainsbury Wellcome Centre, University College London, London, UK.
e-mail: mariaeckstein@google.com; kevinjmiller@google.com

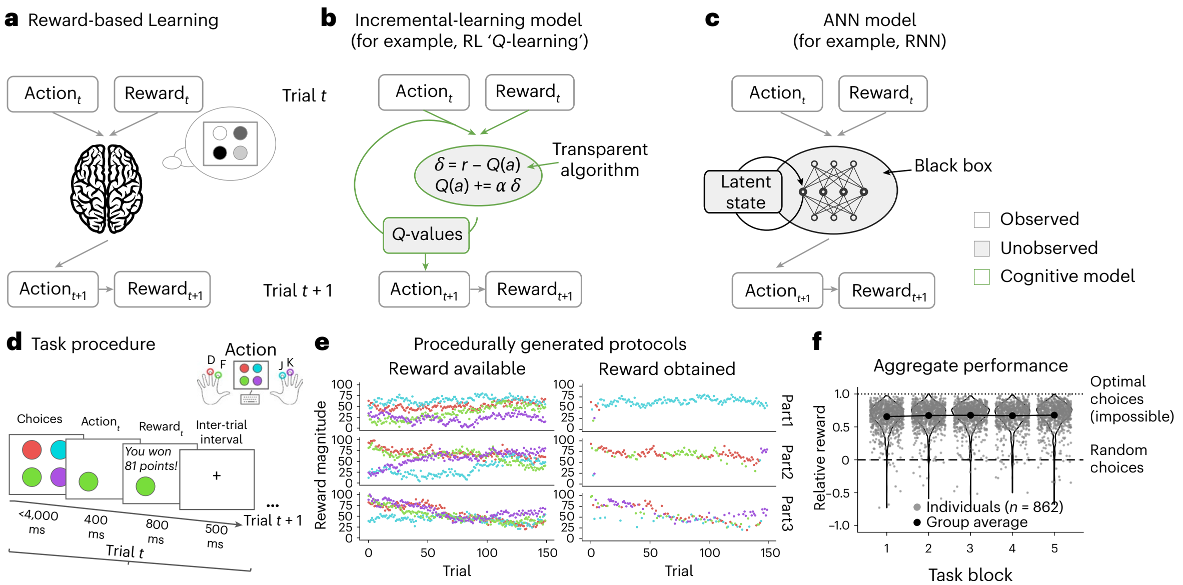


Fig. 1 | Overview. a–**c**, Cognitive modelling. Panel **a** illustrates reward-based learning. Reward-based learning tasks involve choosing one action at a time to win a reward, in an iterative fashion involving many trials. Panel **b** illustrates an incremental-learning model. Variants of RL, specifically Q-learning, are popular behavioural models for such tasks. Q-values approximate the expected reward associated with each action on the basis of an incremental, trial-wise delta-rule update. Panel **c** illustrates an ANN model. While classic cognitive models facilitate understanding of the underlying mechanism, ANNs typically predict empirical behaviour more accurately. **d**–**f**, Experimental design. Panel **d** shows the task procedure. On each trial, participants saw the same four stimuli, pressed a key to select one and obtained the corresponding reward (1–100 points). Each task lasted for 150 trials. Panel **e** shows examples of procedurally generated protocols. Each row shows the protocol for one of three example participants. The number of points available for each action diffused over time (left), independently for each action (colour). A different reward schedule was used for each participant and each task block. Participants' choices (right) reflected individual reward schedules. Panel **f** shows aggregate performance. Each participant (grey dots) performed multiple task blocks (horizontal axis). 'Relative reward' is a measure of task performance that is comparable across different reward schedules (see 'Behavioural analyses' in Methods). The black dots show means over participants, and the error bars (almost invisible) indicate standard errors.

long-term dependencies and the potentially complex learning mechanisms that underlie human behaviour during reward-based learning[28–31]. These networks have the advantage that they typically capture more behavioural variance than handcrafted cognitive models, providing an estimate of the model performance that is possible for a given dataset[30,32,33]. However, fitting behaviour with RNNs typically comes at the expense of interpretability—unlike in classic cognitive modes such as RL, in which each parameter serves a prescribed role, their computations typically require substantial additional work to interpret[34,35].

A budding research field has started to combine ANNs and classic cognitive models[28,31–33,36]. Whereas handcrafted cognitive models are interpretable but frequently underfit the data, ANNs are sufficiently expressive to model complex behaviours but usually hard to understand. For example, Peterson et al.[36] iteratively replaced components of a classic computational model with more expressive ANN counterparts to test increasingly general theories of human decision-making, using gambling tasks. Here we extend this approach to study reward-based learning and memory, which requires modelling both how information about the past is integrated into memory and how the contents of memory are used to guide choice. To do this, we created a hybrid neural–cognitive method that flexibly interpolates between a classic RL model (Fig. 1b) and an RNN (Fig. 1c). Iteratively replacing RL model components with flexible ANNs, we measured which relaxation of constraints improved the model's ability to capture human behaviour. We then inspected the best model's fitted ANN modules to shed light on the underlying mechanisms and to understand how experience shapes memory representations and how these representations drive choice.

Results

We collected a large dataset from a reward-learning task in which human participants repeatedly chose among four possible actions, which were rewarded according to noisy reward magnitudes that drifted over time (a non-stationary 'bandit' task; Fig. 1e)[37]. On each trial of the task, the participants selected one of the four actions and were given the corresponding reward (Fig. 1d). We collected the dataset online (880 participants, 862 of whom passed the inclusion criteria; 4,134 task blocks; 617,871 valid trials; all participants provided informed consent in accordance with Google DeepMind's Human Behavioural Research Ethics Committee, and the study complied with all relevant ethical regulations), which is comparable in size to the largest existing datasets from related tasks[38,39]. Participants tended to choose the actions with larger rewards, indicating that they successfully learned the task (average rewards exceeded chance ($t_{861} = 149.2$; $P < 0.001$; $d = 5.09$; 95% confidence interval (CI) of relative rewards, (66.2, 67.9)) and were numerically above chance on 4,085/4,134 task blocks; Fig. 1f). Both the large size of our dataset and the variability of reward contingencies between participants were crucial to our approach because they allowed RNNs and hybrid models to extract additional variance compared with basic RL models (Supplementary Fig. 7).

We first modelled this dataset using the two extreme approaches, a classic RL-based incremental-update model and a generic RNN. We identified the best RL model (Fig. 2a) through systematic comparison between many RL model variants, using standard methods[40,41] (Supplementary Table 2; implementation details are provided in 'Model architectures' in Methods). Specifically, we started with the simplest model (called 'Simple RL'), a tabular Q-learner with two free model parameters (learning rate and inverse decision temperature), and fitted it to participant behaviour by identifying the parameter values that maximized the negative log-likelihood of human behaviour under the model in the training split of the dataset. We then tested a variety of modifications to Simple RL that have been explored in the literature, including Q-value forgetting[4,42] and a parallel perseveration

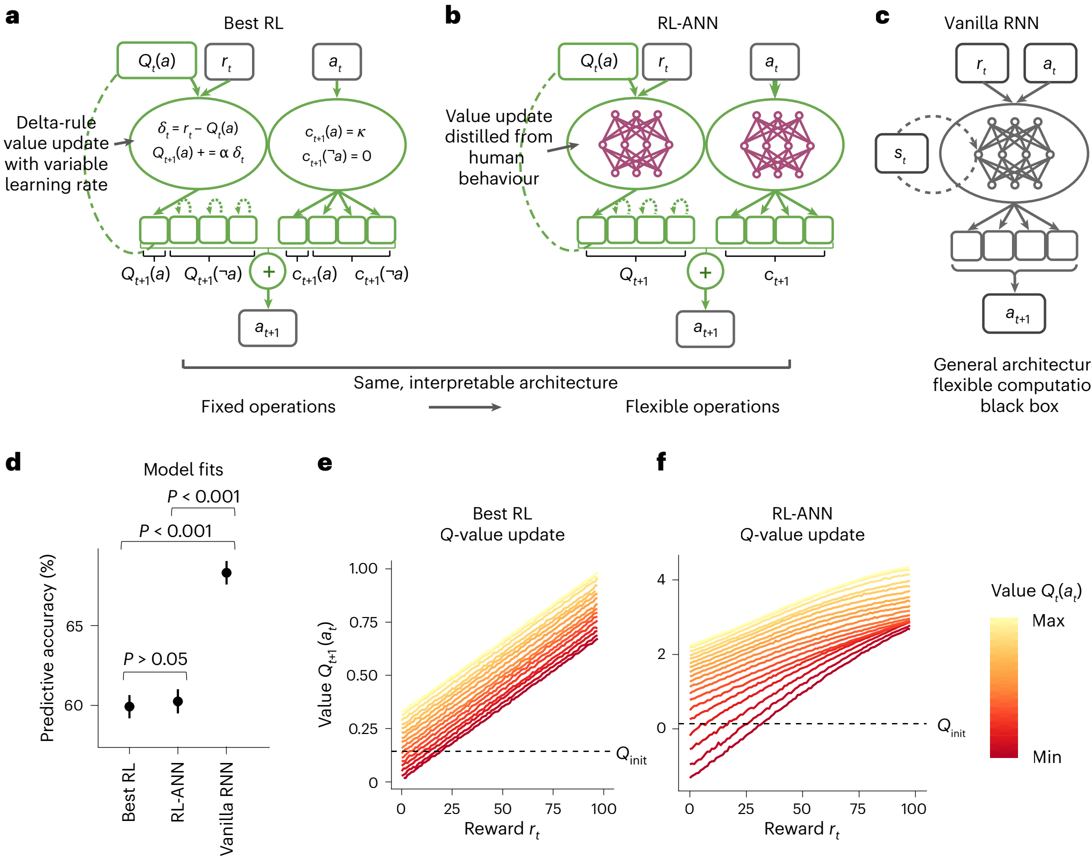


Fig. 2 | Best RL and RL-like models. a–c, Model architectures. Best RL (**a**) is the best handcrafted model based on *Q*-learning, identified using extensive model comparison (Supplementary Results and Supplementary Table 2). It contains a standard *Q*-value update (left oval) with decay of unchosen action values (left oval, dotted recurrent arrows for $Q(\neg a)$), as well as a reward-agnostic choice perseveration mechanism (right oval). The outputs of both computations are combined additively to sample the next choice. RL-ANN (**b**) has the same architecture as Best RL, consisting of a reward module that computes *Q*-values (left oval) and an action module that computes a perseveration kernel (right oval). However, RL-ANN uses ANNs to allow each module to perform any update rule, making it a generalization of linear update models that encompasses Best RL as a special case. Vanilla RNN (**c**) is a standard RNN and the most flexible model. It provides an upper bound in terms of behavioural prediction. **d**, Model fits. Predictive accuracy was derived from the loss of each fitted model to held-out participants not seen during training (see 'Model training' in Methods). Best RL and RL-ANN predicted human choices significantly worse than Vanilla RNN, with no significant difference between them, according to two-sided *t*-tests (see the main text for statistics). The data are presented as mean values over held-out blocks ($n = 413$) plus/minus s.e.m. **e**,**f**, Reward processing. In classic *Q*-learning as modelled by Best RL (**e**), updated values $Q_{t+1}(a_t)$ increase monotonically both in the observed reward r_t and in the previous value $Q_t(a_t)$ (colour), with strictly linear relationships (for model details and equations, see 'Model architectures' in Methods). After fitting to human behaviour, RL-ANN (**f**) acquired a qualitatively similar update rule with monotonically increasing and near-linear relationships. For ease of visualization, we averaged sampled values $Q_t(a_t)$ (colour) within quantile groups to obtain discrete lines.

module that learns from actions rather than rewards[7,43,44]. Among all tested RL model variants, we identified a winning model with six free parameters, called 'Best RL'. Best RL consists of two submodules. The 'reward module' takes as inputs the observed reward, denoted r_t, and the value, $Q_t(a_t)$, of the action a_t that led to this reward, and calculates an updated *Q*-value, $Q_{t+1}(a_t)$, for this action, using the equations specified in Fig. 2a (left). In Best RL, *Q*-values $Q_{t+1}(a_t)$ hence are linear in both the reward r_t and the previous value $Q_t(a_t)$, such that larger rewards and larger previous values lead to monotonically larger updated values (Fig. 2e). Best RL's forgetting mechanism gradually decays *Q*-values back to the initial value Q_{init}. The reward module hence captures pure reward-based learning. In addition, Best RL has an 'action module', which takes as input the previous action, a_t, and sets its perseveration indicator $c_{t+1}(a_t)$ to a value determined by a free parameter. This allows the model to express either action repetition ($c_t(a_t) > 0$) or action switching ($c_t(a_t) < 0$). Perseveration for all other actions, $c_{t+1}(a \neq a_t)$, is 0 (Fig. 2a, right). The outputs of both modules, 'reward logits' Q_{t+1} and 'action logits' c_{t+1}, are combined additively before sampling the action a_{t+1} that is taken on the next trial (for the model details and equations, see 'Model architectures' in Methods).

Best RL is a prime example of a classic handcrafted cognitive model: each mechanism is clearly defined by simple equations, which are modified by just a small number of interpretable model parameters (for example, the inverse decision temperature, β). However, these constraints limit the model's expressivity and potentially its ability to capture human behaviour. To assess whether this is the case, we compared Best RL to a highly expressive 'Vanilla RNN', which can employ a large number of free parameters to model increasingly complex functions. At the core of Vanilla RNN is a recurrent memory module that allows the model to directly share its high-dimensional hidden-layer activations, the latent state $\mathbf{s}_t$, with itself on subsequent trials (for details and equations, see 'Model architectures' in Methods; Fig. 2c). This allows Vanilla RNN to rely on a rich and flexible memory of past trials when making choices. Compared with Best RL, Vanilla RNN has the additional advantage of processing all inputs (a_t, r_t and $\mathbf{s}s_t$) jointly, allowing it to identify arbitrarily complex interactions between them. (Besides the basic RNN architecture, we also fitted more sophisticated sequence models such as long short-term memory networks (LSTMs[45] and transformers[46], which led to qualitatively similar results; see 'Additional model fits' in the Supplementary Information.)

We fit both models to our human data using a cross-entropy loss (equivalent to negative log-likelihood) that quantified how well each model predicted human choices. Note that the models were not trained to find the reward-maximizing policy for the task but to recreate the observed human data as accurately as possible. This approach is sometimes referred to as 'system identification' in engineering[47] or 'behavioural cloning' in machine learning[48,49]. We evaluated all models by cross-validating over participants. This amounts to using a subset of participants to identify the algorithm that best predicted the behaviour of the remaining participants, who completed a different set of task schedules. We trained all models on the same 80% of participants (690 participants; 3,302 task blocks) and tested all models' predictive performance on the same held-out 10% (86 participants; 413 blocks), using the remaining 10% (86 participants; 419 blocks) to select the best hyperparameters for each model (for example, the number of hidden units). Training, validating and testing on different sets of participants eliminates the risk that increasingly flexible models overfit to the training data, and it makes models with different numbers of free parameters directly comparable (see 'Model training' in Methods and Supplementary Table 1). We confirmed that different models were behaviourally distinguishable by generating synthetic behaviour from each model and confirming that the correct model could be identified; this was generally possible because less-flexible models were unable to imitate more-flexible ones (Supplementary Fig. 1). In terms of model comparison, we found that Vanilla RNN predicted the behaviour of unseen participants substantially better than Best RL, correctly anticipating 68.3% (95% CI, (66.9%, 69.7%)) of unseen participants' choices, compared with just 60.6% (95% CI, (59.2%, 62.0%)) by Best RL (chance is 25%; Vanilla RNN versus Best RL, paired t-test: $t_{412} = 28.9$, $P < 0.001$, $d = 1.39$; Fig. 2d). This confirms that, as expected, Vanilla RNN can predict human behaviour more accurately than the best classic RL model. A data sensitivity analysis (Supplementary Fig. 7a) showed that Vanilla RNN's advantage became increasingly prominent for increasing sizes of training data, indicating that collecting more data can improve the extraction of systematic behavioural variance.

Next, we created a series of models that interpolate between the extremes of Best RL and Vanilla RNN. We first created a hybrid model that inherits the architecture of Best RL (Fig. 2a) but replaces its handcrafted equations with flexible ANNs (Fig. 2b). As in Best RL, the reward module is responsible for updating the chosen action's Q-value at each time step. The module has access to the previous reward r_t (for example, 'received 70 points') and value $Q_t(a_t)$ (for example, 'expected 50 points'), but not the identity of the chosen action a_t (for example, 'pressed the D key'). In turn, the action module updates the chosen action's perseveration indicator, for which it has access only to the previous action a_t (for example, 'pressed the D key'). Unlike Best RL, both modules use flexible ANNs to map their respective inputs to the corresponding updated output. This model, which we call 'RL-ANN', is motivated by the insight that Best RL's strictly linear Q-value updates (Fig. 2e) (in conjunction with Best RL's restrictive perseveration mechanism; Supplementary Fig. 6) might be insufficient to capture human learning. For example, existing models propose that value updates might depend on reward in various nonlinear ways[19,50], but the strictly linear Q-learning model does not account for possibilities like these. Similarly, values might depend nonlinearly—or even non-monotonically—on previous values and rewards, but the model does not express this possibility. By replacing Best RL's linear update equations with generic ANNs, we were able to simultaneously test all nonlinear model variants of this kind, without the necessity of specifying each one by hand. During training, RL-ANN's value and action modules have the flexibility to acquire update rules of any functional form and will settle on the one that allows the model as a whole to best match human behaviour. In this sense, RL-ANN represents a whole class of cognitive models: any model that shares Best RL's architecture can in principle be instantiated by RL-ANN, independent of the specific functional form of its updates (for an example, see Supplementary Fig. 1). When we assessed how well RL-ANN predicts the behaviour of unseen participants, however, this added flexibility did not close the gap to Vanilla RNN (RL-ANN: 60.8%; 95% CI, (59.4%, 62.3%); Vanilla RNN: 68.3%; 95% CI, (66.9%, 69.7%); paired t-test: $t_{412} = 32.7$, $P < 0.001$, $d = 1.35$; Fig. 2d; also see Supplementary Fig. 3 for additional variants of Best RL). This suggests that there is no RL-like model—defined as a model that shares Best RL's cognitive architecture, albeit with complete flexibility in terms of the implemented functions—that can predict human behaviour on our task as well as Vanilla RNN. This shows that there exist no modifications to Best RL's update rules that improve the prediction of human task behaviour.

Perhaps surprisingly, RL-ANN did not significantly improve predictions compared to Best RL (RL-ANN: 60.8%; 95% CI, (59.4%, 62.3%); Best RL: 60.6%; 95% CI, (59.2%, 62.0%)); paired t-test: $t_{412} = 1.54$, $P = 0.12$, $d = 0.70$), suggesting that Best RL's original update rules might already be the best in its class. To see if this was the case, we conducted two analyses. We first inspected RL-ANN's learned update functions and compared them to their handcrafted counterparts in Best RL. This analysis can reveal whether among all possible mechanisms RL-ANN could implement, human behaviour lent the most support to the special case of Best RL. We visualized Best RL's Q-value update (Fig. 2e) by calculating the updated values $Q_{t+1}(a_t)$ for every combination of inputs ($0 < r_t < 100$ points; $0 < Q_t(a_t) < 1$), using the standard Q-value equations (Fig. 2a, left; see 'Model analysis' in Methods). We also visualized RL-ANN's Q-value update by extracting the fitted reward module and probing it across its range of inputs ($0 < r_t < 100$ points; $Q_t(a_t)$ between the 5th and the 95th percentile of observed Q-values), while recording its outputs $Q_{t+1}(a_t)$. Indeed, RL-ANN showed an update rule that was monotonic and approximately linear in both r_t and Q_t, similar to Best RL (Fig. 2f), suggesting that human behaviour was best approximated by an algorithm very similar to RL. The corresponding analysis of the action module is shown in the Supplementary Results (Supplementary Fig. 6a). Second, we generated and analysed synthetic behavioural data from both Best RL and RL-ANN, assessing whether the slight differences in the update rule between both would lead to meaningful differences in behaviour. We used each trained model to simulate a behavioural dataset with the same characteristics as the human dataset (the same sample size, reward schedules and train–test–validation split; see 'Model analysis' in Methods). We found that behavioural datasets from both models were qualitatively similar (Supplementary Figs. 10 and 11) but differed from human behaviour (Fig. 4). Thus, even when given the opportunity to learn new, more expressive operations for updating Q-values, RL-ANN approximately recovers the simple solution found in classic RL models and, like them, falls short in predicting human behaviour (Fig. 2d).

Our second hybrid model aims to address this issue by generalizing the architecture further and considering a broader space of models. It is inspired by the finding that learning is affected not only by properties of the chosen option but also by those of options that were available but not chosen, a notion commonly referred to as 'context'[18,19,21,51,52]. For example, an action that won 50 points might be processed differently depending on whether other available actions were expected to win 10 points or 90. To allow for this possibility, we provided the 'Context-ANN' model with additional connections that allow learned information about unchosen actions to modify the learning rule (Fig. 3b). Context-ANN's reward module receives as additional input its own value estimates Q_t (the previous trial's Q-values of all four actions); the action module receives as additional modulatory input c_t (the previous trial's perseveration indicators for all four actions). These modulatory inputs allow Context-ANN to adopt any learning algorithm that can be expressed as a function of the primary input (r_t, a_t) and the corresponding choice variables for all available actions (Q_t, c_t). In model comparison, Context-ANN fit human behaviour substantially better than RL-ANN, increasing the percentage of correctly predicted choices from 60.8% (95% CI, (59.4%, 62.3%)) to 65.4% (95% CI, (63.9%, 66.9%);

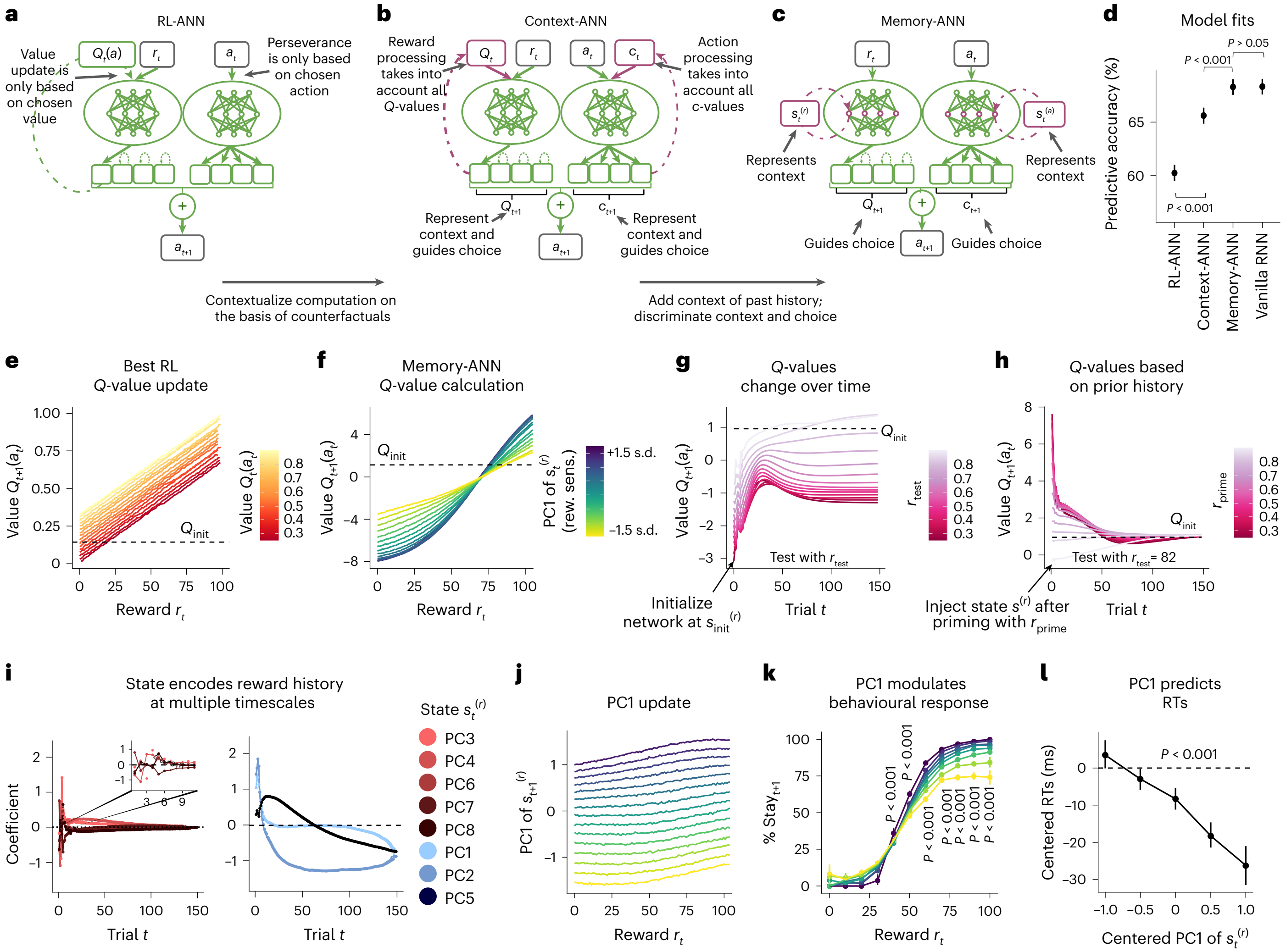


Fig. 3 | Interpretability. a–**c**, Model architectures. Context-ANN (**b**) and Memory-ANN (**c**) were incremental extensions of RL-ANN (**a**). **d**, Model fits. Both Context-ANN and Memory-ANN provided significant improvements, with Memory-ANN not differing from the predictive accuracy of Vanilla RNN, according to two-sided *t*-tests (see the main text for statistics). The data are presented as mean values over held-out blocks ($n = 413$) plus/minus s.e.m. **e**–**i**, Model algorithms. Panel **e** shows the Best RL *Q*-value update (details in Fig. 2e). Panel **f** shows the Memory-ANN *Q*-value calculation. Memory-ANN implements a monotonic but nonlinear, sigmoidal function (Supplementary Fig. 8b), mapping higher rewards onto higher values. The colours show the first principal component (PC1) of the memory state $\mathbf{s}_t^{(r)}$, which we interpret as reward sensitivity. The reference point (Q_{init}) indicates the initial *Q*-value, towards which values of non-chosen actions gradually decayed. As shown in **g**, *Q*-values change over time. When iteratively probed with the same reward magnitude r_{test} (colour), Memory-ANN produces different *Q*-values $Q_{t+1}(a_t)$ depending on the trial *t*. Panel **h** shows how *Q*-values are based on prior history. Different states $\mathbf{s}^{(r)}$ were obtained by subjecting Memory-ANN to 150 different iterations of observing reward r_{prime} (colour). Injecting different states $\mathbf{s}^{(r)}$ led to wide variations in the response to the same r_{test}, which was given for 150 time steps. As shown in **i**, the state encodes the reward history at multiple timescales. Coefficients were obtained from regressing past rewards $r_{1:t-1}$ against each PC of $\mathbf{s}_t^{(r)}$ (see 'Behavioural analyses' in Methods). Some PCs showed sensitivity to the most recent history of rewards, while others showed sensitivity to the long-term history of rewards. **j**, PC1 update. The memory representation $\mathbf{s}_t^{(r)}$ is affected by incoming rewards r_t. PC1 of $\mathbf{s}_t^{(r)}$ exhibits monotonic, near-linear, incremental integration. **k**,**l**, Behavioural relevance. PC1 modulates the behavioural response (**k**). The mapping from reward magnitudes to action reselection ('Percent stay') is modulated by PC1 of $\mathbf{s}_t^{(r)}$ (colour). $P < 0.001$ in paired, within-participant, Bonferroni-corrected, two-sided *t*-tests comparing stay frequency between PC1 ≤ 0 and PC1 > 0, separately for each bin of reward magnitudes (*x* axis). PC1 also predicts response times (**l**). The relation between PC1 of $\mathbf{s}_t^{(r)}$ (fitted to human behaviour) and response times (RTs) (log-transformed; both mean-centred within blocks) is shown. The data are presented as mean values over all blocks ($n = 4{,}134$) for each PC1 bin, plus/minus 95% bootstrapped CIs. $P < 0.001$ in mixed-effects linear regression (see the main text).

paired *t*-test: $t_{412} = 28.3$, $P < 0.001$, $d = 1.27$; Fig. 3d). Each module played a unique role in improving the prediction accuracy (Supplementary Tables 3 and 4). Nevertheless, Context-ANN still fell short of Vanilla RNN (68.3%; 95% CI, (66.9%, 69.7%); paired *t*-test: $t_{412} = 16.8$, $P < 0.001$, $d = 0.83$), indicating that the inclusion of context processing was not sufficient to capture human behaviour on our task and that an even more flexible architecture is required.

We hence turned to the role of memory processing, testing whether a model that can retain a richer representation of the past can explain human behaviour better than previous models. Indeed, several studies have shown that both recent[53] and distant[13,17] outcomes affect human learning in ways that cannot be explained by incremental updating alone. It has also been suggested that humans keep track of additional latent variables beyond *Q*-values—for example, remembering past prediction errors to adapt the future speed of learning[54]. (We implemented several versions of such variable-learning-rate models, which showed slightly better performance than Best RL but still fell far short of Vanilla RNN; see 'Model architectures' in Methods and

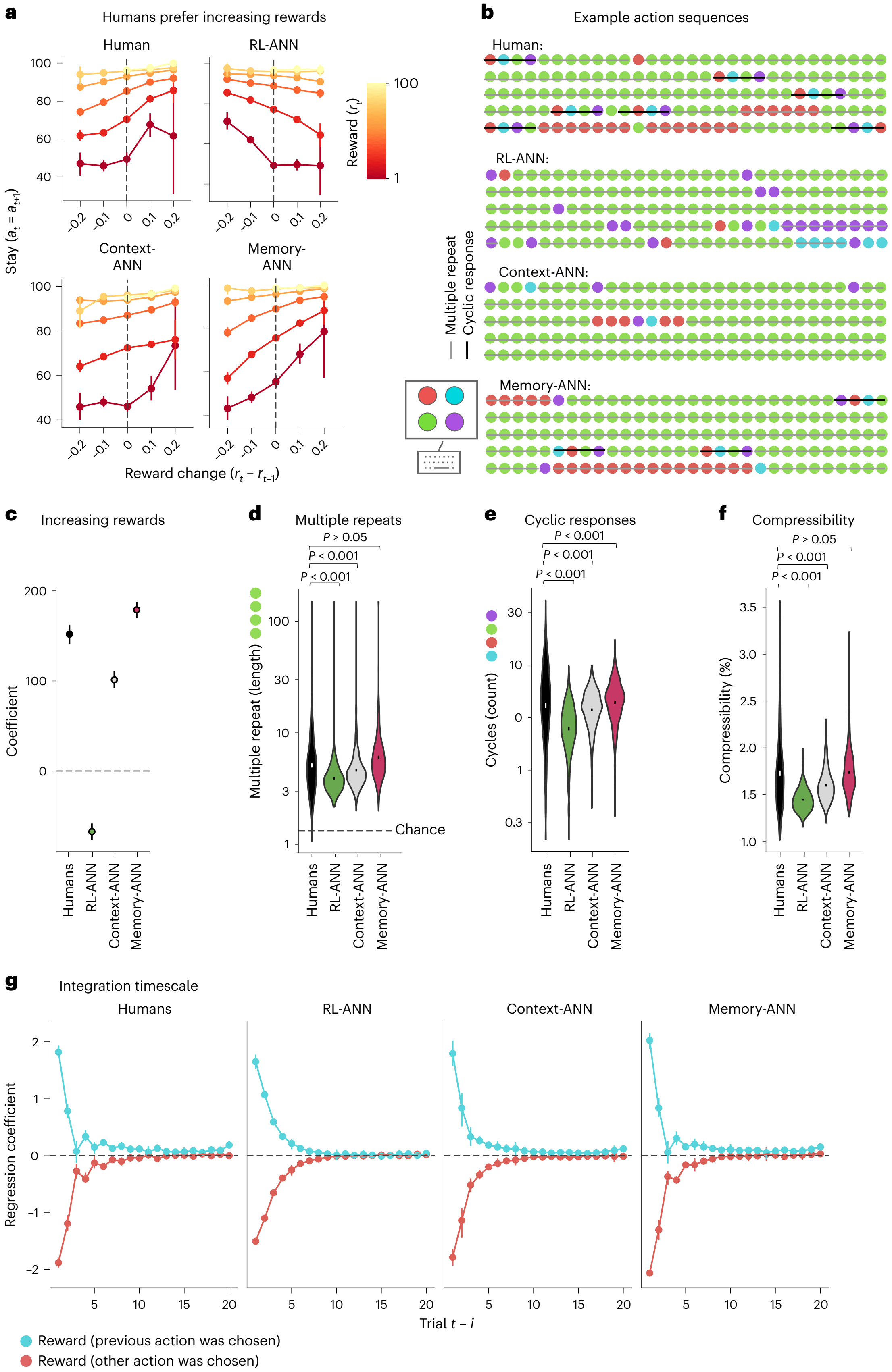
Supplementary Results). To test whether the ability to retain richer representations of the past is crucial to explain learning in our task, we created our final hybrid model: Memory-ANN. Whereas Context-ANN receives the modulatory inputs Q_t and c_t to account for unchosen actions, Memory-ANN requires inputs that have potential access to the entire task history and could represent any summary statistic thereof, including high-dimensional and nonlinear ones. The latent states of an RNN have precisely these properties. We hence replaced the reward module's inputs, Q_t and $Q_t(a_t)$, with the activities of the reward module's hidden units from the previous time step, which we denote $\mathbf{s}_t^{(r)}$ (this turns the reward ANN into a reward RNN). Likewise, we replaced the action module's input c_t with the previous activities of its hidden units, $\mathbf{s}_t^{(a)}$ (turning the action ANN into an action RNN; Fig. 3c). These modifications have the effect of explicitly separating memory variables ($\mathbf{s}_t^{(r)}$ and $\mathbf{s}_t^{(a)}$) from choice variables (Q_t and c_t), which in previous models were assumed to be identical. Hence, Memory-ANN has the ability to express a wide range of memory-based learning models that are based on modulating reward (and action) processing on the basis of any learned features of the reward (and action) history. Note, however, that Memory-ANN is still more constrained than Vanilla RNN: the same update applies regardless of which action is being updated, the values of all unchosen actions decay strictly exponentially, reward processing does not have access to past or present actions and vice versa for action processing, and the outputs of reward and action processing are combined by simple addition. Memory-ANN improved the prediction of human behaviour substantially compared with Context-ANN (Context-ANN: 65.4%; 95% CI, (63.9%, 66.9%); Memory-ANN: 68.3%; 95% CI, (66.9%, 69.7%); paired t-test: $t_{412} = 17.9$, $P < 0.001$, $d = 0.95$; Fig. 3d). Most importantly, Memory-ANN's predictions were not significantly different from those of Vanilla RNN, the most general model we tested (Memory-ANN: 68.3%; 95% CI, (66.9%, 69.7%); Vanilla RNN: 68.3%; 95% CI, (66.9%, 69.7%); paired t-test: $t_{412} = 0.32$, $P = 0.75$, $d = 0.14$). This indicates that Memory-ANN extracted all systematic variance in the dataset that can be extracted by an RNN, suggesting that its architectural constraints (Fig. 3c) identified relevant biases in human behaviour. Indeed, there was no constraint whose removal improved model prediction (Supplementary Tables 4, 5, 7 and 8). Taken together, these results suggest that our participants performed the task by creating rich memories of reward and action history and used them to guide reward learning.

What mechanisms underlie the learning processes in Memory-ANN? To answer this question, we inspected the functions learned by the neural network modules during model fitting. We first considered reward processing, evaluating the reward module by probing it across its range of inputs (r_t and $\mathbf{s}_t^{(r)}$) while recording its outputs $Q_{t+1}(a_t)$ (see 'Model analysis' in Methods). We found that the reward module maps rewards r_r onto new values $Q_{t+1}(a_t)$ in a monotonic, roughly sigmoidal way (Fig. 3f and Supplementary Fig. 8b). Notably, the reward module does not have access to previous values $Q_t(a_t)$ (nor can it reconstruct them using its hidden state input $\mathbf{s}_t^{(r)}$), which means that Memory-ANN does not take into account previous values $Q_t(a_t)$ when calculating new values $Q_{t+1}(a_t)$. This is in stark contrast to most RL models, which posit that values are learned incrementally. Instead, Memory-ANN simply maps large rewards onto large Q-values and small rewards onto small Q-values, without calculating reward prediction errors or incremental updates. If Memory-ANN's latent state $\mathbf{s}_t^{(r)}$ was fixed over time (Supplementary Fig. 12a), this simple mapping mechanism would lead to somewhat rigid choice behaviour (Supplementary Fig. 12b). However, the flexibility of the latent state enables adaptive choices: $\mathbf{s}_t^{(r)}$ follows a stereotypical trajectory over the time course of a task (Supplementary Fig. 5f), which leads to a gradual change in the assignment of Q-values to rewards as the task progresses. To assess this, we initialized a fresh reward module and probed it with sequences of identical rewards, recording the resulting Q-values. Across the range of rewards, earlier trials lead to smaller Q-values than later ones, which can support a behavioural shift from more 'exploratory' to more 'exploitative' choices (Fig. 3g and Supplementary Results). $\mathbf{s}_t^{(r)}$ also adapts the calculation of future Q-values by encoding complex moments summarizing the history of rewards. We forced a fresh reward module into several extreme states by priming it with different reward sequences and tested its responses to a new reward. This reward elicited tremendously different Q-values depending on the injected state, an effect that took up to several dozen trials to disappear (Fig. 3h). We finally causally probed the role of state $\mathbf{s}^{(r)}$ by injecting activity into different principal components (PCs), observing the corresponding short- and long-term perturbations in the calculation of Q-values (Supplementary Fig. 12e), and testing the effects of individual trigger rewards (Supplementary Fig. 12c) or reward sequences (Supplementary Fig. 12d) on state $\mathbf{s}^{(r)}$.

We next identified the mechanisms by which $\mathbf{s}_t^{(r)}$ biases the calculation of Q-values. We conditioned the trained Memory-ANN on each participant's action sequence to obtain, for each participant, the trial-by-trial sequence of latent variables Q, c, $\mathbf{s}^{(r)}$ and $\mathbf{s}^{(a)}$ (Supplementary Fig. 5a–g), and applied principal component analysis to $\mathbf{s}^{(r)}$. The

Fig. 4 | Behavioural model validation. **a**–**g**, Comparison of human and simulated model behaviour. As shown in **a**, humans prefer increasing rewards. Humans (top left) repeated a choice ('stayed'; y axis) more often when the rewards for this choice had increased (positive reward change; x axis) rather than decreased (negative reward change) on the previous two trials. Best-RL (Supplementary Fig. 11a) and RL-ANN (top right) showed the inverse pattern, whereas Context-ANN and Memory-ANN qualitatively reproduced the effect. The data are presented as mean values over blocks ($n = 4{,}134$) for each reward change bin, plus/minus 95% bootstrapped CIs. Panel **b** shows example action sequences. These are raw sequences of chosen actions (coloured circles) from humans and models performing the reward schedule shown in the top row of Supplementary Fig. 5a. Humans showed two common patterns: multiple repeats, extended periods of the same action (grey lines); and cyclic responses, sets of four sequential trials in which each action was sampled once (black lines). Panel **c** shows the effect of reward change on stay probability (see **a**). The data are presented as the regression coefficients from the model `stay ~ reward × reward_change` ($n = 862$ participants), plus/minus standard errors of the coefficient estimates. As indicated in **d**, humans showed longer sequences of identical actions (average length, 6.9; 95% CI, (6.1, 7.6)) than expected by chance (chance length, 1.3; $t_{861} = 14.6$, $P < 0.001$, $d = 0.50$) or seen in RL-ANN (average length, 4.5; 95% CI, (3.9, 5.1); $t_{861} = 9.4$, $P < 0.001$, $d = 0.32$) and Context-ANN (average length, 5.5; 95% CI, (4.8, 6.1); $t_{861} = 5.4$, $P < 0.001$, $d = 0.19$). Memory-ANN sequence length did not differ from that of humans (average length, 7.5; 95% CI, (6.8, 8.3); $t_{861} = -1.8$, $P = 0.075$, $d = 0.06$). As shown in **e**, human choices contained more cyclic sequences than synthetic data (human mean, 5.37; 95% CI, (5.01, 5.68); RL-ANN mean, 2.68; 95% CI, (2.59, 2.76), $t_{861} = 10.9$, $P < 0.001$, $d = 0.37$; Context-ANN mean, 3.87; 95% CI, (3.77, 3.97), $t_{861} = 7.77$, $P < 0.001$, $d = 0.26$). Memory-ANN produced the qualitatively closest number of cyclic sequences compared to humans (Memory-ANN mean, 4.62; 95% CI, (4.48, 4.76), $t_{861} = 5.7$, $P < 0.001$, $d = 0.19$). As shown in **f**, we used the Lempel–Ziv–Welch algorithm to compress human and model action sequences (see 'Behavioural analyses' in Methods), quantifying systematic temporal structure. Human sequences were substantially more compressible than those of RL-ANN and Context-ANN (human mean, 1.73; 95% CI, (1.70, 1.76); RL-ANN mean, 1.45; 95% CI, (1.44, 1.45), $t_{861} = 20.48$, $P < 0.001$, $d = 0.70$; Context-ANN mean, 1.60; 95% CI, (1.59, 1.61), $t_{861} = 9.47$, $P < 0.001$, $d = 0.322$). Memory-ANN compressibility did not differ from that of humans (mean, 1.74; 95% CI, (1.72, 1.76), $t_{861} = 0.69$, $P = 0.49$, $d = 0.02$). In **d**–**f**, the data are presented as violin plots of the raw data distribution ($n = 862$ participants) and error bars indicating 95% CIs of the mean. $P < 0.001$ in paired, two-sided t-tests. Panel **g** shows the integration timescale. Weights of trial-history regression models trained to predict future choices on the basis of past choices and rewards (see 'Behavioural analyses' in Methods) are plotted. Only Memory-ANN reproduced the patterns seen in human behaviour qualitatively. The data are presented as mean values over participants ($n = 862$) for each trial, plus/minus 95% bootstrapped CIs. (The corresponding plots for Simple RL, Best RL and Vanilla RNN for all these measures are shown in Supplementary Figs. 10 and 11).

first component (PC1) modified the gain of the sigmoidal mapping from rewards to *Q*-values, effectively controlling the sensitivity of *Q*-values to reward magnitude (Fig. 3f). At high gains (blue), even small reward differences lead to large differences in *Q*-values, whereas at small gains (yellow), large reward differences are required to produce moderate differences (Fig. 3f). We therefore interpreted PC1 as tracking



the model's current sensitivity to reward. Confirming this interpretation, we found that large values of PC1 are associated with a high probability of repeating a choice that led to a large reward and a small probability of repeating a choice that led to a small reward, while low values of PC1 are associated with a shallower relationship (Fig. 3k). The value of PC1 also correlated with participants' response times, showing that a model-derived variable predicted behaviour in a dimension that was not included in the training data (mixed-effects regression, slope = −0.968.9, $z = -13.3$, $P < 0.001$; Fig. 3l). These modulations occurred within participants, showing that fluctuations in reward sensitivity captured gradual changes in participants' behavioural strategies, rather than individual differences between participants. (For more discussion of individual differences, see Supplementary Results.) Subsequent PCs affected the gain, range, bias and scale of the sigmoid (Supplementary Fig. 8). This mechanism enables Memory-ANN to flexibly adapt its behaviour to the current reward context. The corresponding analysis for Memory-ANN's action module is shown in the Supplementary Results (Supplementary Fig. 6a).

How does $\mathbf{s}_t^{(r)}$ represent the past history? We first determined how new information alters existing representations. We probed the reward module across its range of inputs (r_t and $\mathbf{s}_t^{(r)}$), this time collecting the latent state $\mathbf{s}_{t+1}^{(r)}$ as the output. We found that PC1 (reward sensitivity) integrated rewards in a monotonic, near-linear way, increasing slightly after big rewards and decreasing slightly after small ones (Fig. 3j). Several other state PCs showed similar monotonic, near-linear integration patterns, exhibiting steeper (for example, PC3 and PC4) or shallower slopes (for example, PC5) (Supplementary Fig. 8). This supports the notion that $\mathbf{s}_t^{(r)}$ integrates new rewards using parallel update rules with a variety of integration timescales (see also Supplementary Fig. 12). We next assessed the contents of the representations, using a decoding analysis. For each delay i, we used lagged regression to predict each state PC from the reward r_{t-i}. We found that a subset of PCs showed large regression weights to just a handful of the most recent rewards, consistent with the idea that these PCs track individual recent outcomes while being insensitive to all earlier events (Fig. 3i, left). Some other PCs were sensitive to the entire history of rewards, potentially providing a baseline for how reward-rich the environment is overall, and whether this is changing for better or for worse (Fig. 3i, right). These results were consistent across multiple independent runs of Memory-ANN (Supplementary Fig. 2) and were recoverable when Memory-ANN was fit to synthetic data (Supplementary Fig. 1). These findings indicate that Memory-ANN flexibly modulates the mapping from rewards and Q-values, continuously adjusting to the time on task and the reward history.

Finally, we tested whether Memory-ANN captured qualitative features of human behaviour that more restricted models were unable to capture[55]. We used each fitted model to simulate task behaviour 'open-loop' (without knowledge of human choices), and on the same tasks as humans (see 'Model analysis' in Methods). First, we sought a behavioural signature of the history-dependent processing of reward sequences. For this, we considered pairs of trials in which the same action was selected twice in a row, and we quantified the tendency to select that action again on the following trial as a function of the change in reward magnitudes (Fig. 4a). Best-RL (Supplementary Fig. 11a) and RL-ANN preferred actions for which the second-most-recent reward r_{t-1} was larger than the most recent reward r_t (colour), on which we conditioned. This arises because these models make choices on the basis of running averages, and a larger reward in the past increases this average. Humans, in contrast, preferred actions whose second-most-recent rewards were lower[56,57], as if anticipating that a recent increase in reward magnitudes will continue in the future. Only Context-ANN and Memory-ANN reproduced this effect qualitatively (Fig. 4a,c). This shows that memory representations need to contain information about unchosen actions or task history to capture how participants modify their responses. Second, we assessed behavioural patterns related to the history-dependent processing of actions. We focused on stereotyped action sequences, such as multiple repeats (AAAA) and cyclic responses (ABCD), in which the time horizon extends for multiple trials[58]. Memory-ANN was able to capture the strong human preference for such multi-trial patterns, while no other model was able to do so (Fig. 4b,d–f). The prevalence of these behavioural motifs implies that human participants committed to stretches of exploiting an action they believed was best (AAAA), interspersed with brief episodes of systematically exploring whether a different action might be better (ABCD)[59]. We then characterized the overall structure within the observed choice patterns, computing the compressibility of all action sequences using a standard algorithm (see 'Behavioural analyses' in Methods) and comparing humans to model predictions (Fig. 4f). Only Memory-ANN achieved a similarly high compression ratio as humans; Context-ANN showed intermediate compressibility, and RL-ANN showed the lowest compressibility. This reveals that the choices of humans and of Memory-ANN had structured relationships with other choices nearby, which was less the case in simpler models. Finally, we assessed the history dependence of actions using lagged regression[60]. We found that participants showed shallow and non-monotonic history dependence that was reproduced by Memory-ANN but not by other models (Fig. 4g). Memory-ANN hence captures a range of patterns that are characteristic of human behaviour, including many that violate classic models. While some of these patterns have been described in the past[14,56–59,61], they have not previously been captured in a single model. It is a challenge in computational cognitive science that the identification of new patterns often leads to the creation of idiosyncratic model features and a multiplication of model architectures, rather than consolidation in a single framework.

Discussion

In psychology and neuroscience, reward-learning behaviour is commonly understood using computational models based on Q-learning, in which memory consists entirely of a set of incrementally updated decision variables. We have shown that this family of models cannot adequately account for reward-guided learning in humans, using a large dataset from a classic reward-learning task and a systematic model comparison approach that integrates deep neural networks into classic cognitive architectures. We identify instead a model that contains both decision variables that drive choice directly and a set of latent memory variables that modulate the update of these decision variables but do not directly drive choice. These memory variables track a complex history of rewards and choices over multiple timescales. We show that this model captures human behaviour in detail, both reproducing a number of intricate features of the dataset and matching generic neural networks in quantitative quality of fit. At the same time, it is interpretable as an algorithmic model of human reward learning.

Recent work implicitly recognizes the complexity of how humans use memory in reward-learning tasks, highlighting that learning processes often operate at multiple different timescales. This has been framed as a multiplicity of memory mechanisms[13,14,53,62–65] and is consistent with evidence that the brain represents task-history information at a diversity of timescales[26,66–68]. Memory beyond decision variables is also present in several handcrafted models of human reward learning. For example, Bayesian inference models[37,69,70] track a measure of the model's uncertainty that creates non-Markovian dependencies between choice variables, variable-learning-rate models[71,72] track a measure of environmental volatility, and actor-critic models[73,74] and reference-point models[75] track an action-independent measure of expected value. However, all these models are based on handcrafted equations, and the ones we have tested here fall short compared with more flexible ANN-based models. Memory-ANN reveals that learning at different timescales is supported by a flexible recurrent memory system that is one step removed from behavioural choice, and it shows that the way in which observed outcomes are mapped to future choices is a complex, yet interpretable, function of task history.

The cognitive architecture of Memory-ANN is modular in two ways. First, reward-based learning and action-based learning are divided into two parallel modules. This idea has origins in early work on the psychology of learning—for example, in the distinction between Thorndike's[76] law of effect (actions that lead to good outcomes should be repeated) and law of exercise (actions that have been taken in the past should be repeated). A separation of reward-based from action-based learning is present in a number of computational models of behaviour[5,43,70,77], and evidence from neuroscience suggests that the brain may incorporate such modularity[78–81]. These models typically imagine that action-based learning takes the form of perseveration, in which actions that have been taken in the past are more likely to be taken in the future[43], and that reward-based learning takes the form of incremental RL[1,2]. Memory-ANN retains the basic separation between reward-based and action-based learning but allows for each module to implement substantially more sophisticated mechanisms. This uses Memory-ANN's second kind of modularity: both reward-based and action-based learning are divided into a 'deep' memory component, which learns rich hidden representations of the past but does not drive choice, and a 'shallow' choice component that guides action selection. This architecture shares features with models of more complex reward-learning tasks, many of which draw on hierarchical cognitive architectures[6,82–85]. Evidence from neuroscience also supports the idea of a gradient of abstraction in the neural architecture[82,86,87]. Our results suggest that humans may use hierarchically structured algorithms even in superficially simple reward-learning tasks.

One limitation of the current work is a lack of focus on individual differences. We fit a single model to the whole population, which allows us to infer the likely mechanisms that characterize the behaviour of all participants but does not provide insight into individual differences between them. Others have modelled individual differences within RNN-based frameworks[31,88], and similar approaches could be used to extend the current work. However, RNN-like models implicitly capture individual differences even when they are not modelled explicitly[89], which means that in principle, some of our results concerning differential performance between Memory-ANN and Best RL might reflect the network better capturing aspects of between-participant differences, rather than (as we interpret it) improved modelling of the progression of learning within each participant. While additional analyses ruled out the possibility that this difference between the models accounts for our key results (for example, that Memory-ANN outperforms Best RL and that aspects of its architecture and latent state dynamics capture within-participant learning), it remains possible that some of our conclusions reflect a contribution of both between- and within-participant effects. Additional work, both experimental and analytical, will be required to fully tease apart these possibilities. Overall, this direction offers intriguing new prospects for studying individual differences as well as the dynamic fluctuations that occur within individuals over time (Supplementary Information).

Science faces a theory discovery problem: it is fundamentally more difficult to create new models than to evaluate existing ones[90,91]. In psychology and neuroscience, new laboratory technologies have enabled scientists to collect larger datasets than ever before, a development that might provide new solutions to this problem[5,32,92–95]. We used a combination of hypothesis-driven architecture search and data-driven function approximation[36] to successfully identify a predictive yet interpretable model of human reward-based learning. With the rich tradition of classic cognitive modelling providing the theoretical framework to guide our model search, machine learning tools contributed the ability to approximate any functional form on the basis of sufficient data. This approach allowed us to compare the most relevant model classes in the most general case. The same approach could be applied to a wide range of open questions, both within the cognitive sciences and beyond. There is a ubiquitous need for models that can capture the complexity in rich datasets and also provide interpretable explanations.

Methods

Dataset

Participants. We recruited 880 participants on Prolific (app.prolific.co). No statistical methods were used to predetermine the sample size, but our sample size is orders of magnitude larger than those of most traditional lab-based human experimental studies and similar to those reported in previous publications focused on large-scale experiments[36,39,96,97]. In agreement with the ethical guidelines of the Google DeepMind Human Behavioral Research Ethics Committee, all participants were local to the UK and fluent in English. The participants provided informed consent and were paid at a rate of 12 pounds per hour; there was no performance-based bonus payment. The study was not preregistered.

Experimental procedure. The participants completed one training block and several testing blocks of our bandit paradigm (see below), each using different visual stimuli. After each block, the participants were truthfully informed how many points they had won, how many points they could have won (the sum of points from each trial's best choice option) and how many points they would have won by choosing randomly (the average points of all choice options). At the end of the study, the participants were asked for their highest level of education and offered the opportunity to voice thoughts and concerns. The experimental task was written using jsPsych[98] and served on cognition.run.

Exclusion criteria. Eighty participants were asked to complete one training and three testing blocks of 150 trials each. The remaining 800 participants were asked to complete one training block of 50 trials and five testing blocks of 150 trials, for a total of 4,240 task blocks. Four participants in the first (5%) and 14 participants (1.75%) in the second sample failed to finish the experiment and were excluded, leading to an initial sample of 880 − 18 = 862 participants who collectively finished (80 − 4) × 3 + (800 − 14) × 5 = 4,158 task blocks. We further excluded blocks in which participants missed more than 15 of the 150 trials (10%), 24 blocks in total (0.58%). Hence, our final dataset comprised 4,134 blocks (with 617,871 valid trials) from 862 participants. Of these 862 participants, 858 (99.5%) provided valid demographic information: 341 (39.7%) were female, and 517 (60.3%) were male; the average age was 39.7 years, with a range of 18–88 and a standard deviation of 13.1 years.

Task

The participants performed a classic four-armed drifting bandit task[37,99]. On each trial t of this task, participants chose one of four bandits and observed the corresponding reward r_t. At the first trial $t = 1$, each arm was initialized independently and uniformly at random between 1 and 100 points. The mean reward $\mu_{t,i}$ at each trial t and arm i was determined by a Gaussian random walk that evolved according to standard deviation σ_d and centrality λ:

$$\mu_{t,i} \sim \mathrm{N}(\lambda \times \mu_{t-1,i} + (1 - \lambda) \times 50, \sigma_\mathrm{d})$$

The actual reward $r_{t,i}$ observed by participants was sampled from a Gaussian distribution with mean $\mu_{t,i}$ and standard deviation σ_o:

$$r_{t,i} \sim \mathrm{N}(\mu_{t,i}, \sigma_\mathrm{o})$$

Following prior work[37,99], we used $\lambda = 0.9836$, $\sigma_\mathrm{d} = 2.8$ and $\sigma_\mathrm{o} = 4$. Unlike prior work[37,99], we created a new reward schedule for each participant for each task to increase the behavioural variation in the dataset and facilitate the fitting of neural network models.

On each trial, the participants saw four visual stimuli on the screen, one representing each bandit (Fig. 1d). Each bandit was presented in the same location on each trial, but new stimuli were used on each task iteration, and their positions were randomly shuffled between participants. Participants had four seconds to select a bandit using the keys 'D', 'F', 'J' and 'K'. When participants failed to make a response within this time window, they were encouraged to respond faster on the next trial and reminded of the response keys. The participants were also told that they had received zero points for that trial. Only a very small percentage of trials in the final sample were missed (0.36%). When participants made a valid selection, the chosen bandit remained on the screen for 400 milliseconds while the others disappeared. The trial outcome was then presented in addition to the chosen bandit (for example, 'You won 79 points.'). After another 800 milliseconds, an inter-trial interval of 500 milliseconds began, after which the next trial started.

Behavioural analyses

Task performance. We first aimed to assess participant performance. The raw number of points is not a good measure of performance because each task block is based on a different reward schedule (see above), and hence the same number of points can indicate good or bad performance. To obtain a performance measure that is comparable between blocks, we calculated relative rewards. The relative reward $r_{\mathrm{rel},t}$ is the number of points r_t obtained on trial t, normalized between the maximum number of points available on that trial ($\max(p_t)$) and the number of points expected on that trial by random selection ($\mathrm{mean}(p_t)$):

$$r_{\mathrm{rel},t} = \frac{r_t - \mathrm{mean}(p_t)}{\max(p_t) - \mathrm{mean}(p_t)}$$

Averaging $r_{\mathrm{rel},t}$ across all trials t gives the relative reward of a block r_{rel}, shown in Fig. 1f. A block's relative reward would be 1 if a participant chose the best bandit on each trial (which is impossible); the relative reward is close to 0 when a participant chooses randomly and smaller than 0 when a participant systematically prefers bandits with smaller-than-average rewards.

Lagged regression. We next focused on learning, assessing how past task events affected participants' future behaviour. Following a model-free approach, we used logistic regression to quantify the effects of past actions a_{t-i} and outcomes r_{t-i} on participant choices a_t and to compare the time courses of these effects between cognitive models (Fig. 4g). For each cognitive model, we calculated four regression models, one per bandit. There was no reason to respond differently to the four bandits, and indeed, the four regression models produced nearly identical results in all cases; hence, we averaged the results for visualization. Each regression model predicted the time course of choices for one particular bandit, $a_{1:n}$ (number of trials $n = 150$), coding trials as 1 when the bandit was chosen and 0 otherwise. We used two sets of regressors to predict $a_{1:n}$. 'Bandit-reward' regressors contain the time course of the number of points obtained in the past after choosing the current bandit: $r_{i:n+i} \times a_{1:n}$. For example, the bandit-reward regressor at $t-1$ contains the sequence of points obtained on the previous trial for those trials in which participants had chosen the current bandit; trials in which a different bandit was chosen contain the value 0. The second set of regressors are 'other-reward' regressors, which indicate the number of points obtained in the past after choosing a bandit other than the current bandit: $r_{i:n+i} \times (1 - a_{1:n})$. We predicted choices $a_{1:n}$ from past events up to 20 trials in the past, $1 < i < 21$, such that our models contained 40 regressors (20 bandit-reward and 20 other-reward regressors).

Mixed-effects regression. We next assessed how PC1 of participants' reward state $\mathbf{s}_t^{(r)}$ (reward sensitivity) affected subsequent choices a_{t+1} and response times r_{t+1}. To this aim, we ran a mixed-effects regression model specifying random effects of participants, including trial number and block number as nuisance predictors. For Fig. 3i, we preprocessed response times by log-transforming and then centring on the mean, individually for each participant and each block. We preprocessed PC1 of $\mathbf{s}_t^{(r)}$ by centring on the mean, individually for each participant and each block. Centring both measures across participants allows us to directly test for within-participant differences. This rules out the possibility that all observed differences arose from differences between participants, such that different participants occupied different states, which were also associated with differences in response times. Instead, the same participants transitioned through different regions of the space, which also captured differences in response times.

Multiple repeats and cyclic responses. We then focused on the structure within participants' choice sequences. We calculated the average length of multiple repeats (continuous streaks that repeat the same action; Fig. 4d), and we counted the number of cyclic responses (four subsequent trials in which each of the four available actions is chosen once; Fig. 4e).

Compressibility ratio. We finally quantified the structure within participants' choice sequences by estimating sequence compressibility (Fig. 4f). We used the Lempel–Ziv–Welch (LZW) algorithm, a relatively simple standard compression algorithm for sequential data[100,101]. LZW first identifies the subsequences (for example, ABCD or AAAA) that an original sequence is composed of and then re-expresses the original sequence in terms of these subsequences, hence reducing the sequence length by taking advantage of repetitions. Sequences that are composed of a small number of subsequences (for example, ABCDABCD) are more compressible than random sequences without such structure (for example, DADDCBDB). To estimate the compressibility of participants' choice sequences, we first compressed each block's original choice sequence using LZW, obtaining the compressed sequence length l_{LZW}. For comparison, we also sampled random sequences of the same length as the original blocks ($n = 150$) using the same four elements (A, B, C and D). We also compressed these random sequences to obtain the baseline compressibility, b_{LZW}, expected for sequences of the same length and with the same number of elements, just by chance. Finally, we calculated the ratio between the length of compressed random sequences and that of participants' blocks, obtaining the compressibility score $\frac{b_{\mathrm{LZW}}}{l_{\mathrm{LZW}}}$.

Model architectures

***Q*-learning model architectures.** We obtained our Best RL model by comparing many variants of Q-learning[41]. In (tabular) Q-learning, each action a is associated with a value $Q(a)$, which approximates the expected reward of a (ref. 2). Values are learned incrementally over trials, on the basis of the observed reward. On each trial t, the value of the chosen action is updated by a fraction α (called the 'learning rate') of the reward prediction error, the discrepancy between the reward r_t and the action value going into this trial, $Q_t(a)$:

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

The standard formulation of Q-learning applies to environments with multiple states, where taking an action a in state $\mathbf{s}$ leads the agent to state $\mathbf{s}'$. In such environments, the Q-value update includes a term corresponding to the Q-value of the subsequent state, including a discount factor $0 < \gamma < 1$. For example, the on-policy SARSA algorithm performs the following Q-value update:

$$Q_{t+1}(\mathbf{s}, a) = Q_t(\mathbf{s}, a) + \alpha \times (r_t + \gamma \times Q_t(\mathbf{s}', a') - Q_t(\mathbf{s}, a))$$

In this paper, because the environment does not provide state transitions (for example, the subsequent state s' does not depend on the previous state $\mathbf{s}$ and action a), we use a simplified algorithm without the term $\gamma \times Q_t(\mathbf{s}', a')$, following standard conventions in cognitive modelling[40,41].

We compared our RL models head-to-head with neural networks. To make this comparison fair, we included a bias parameter b in the RL models. b allows a linear offset in value updates, a freedom that the neural-network models have by design:

$$Q_{t+1}(a) = Q_t(a) + \alpha \times (r_t - Q_t(a)) + b \tag{2}$$

On any trial t, Q-learning agents select an action by transforming the vector $\mathbf{Q}_t$ of all four action values into a vector of choice probabilities $\mathbf{p}_t$ of the same length, using the softmax function. This transformation can have a 'lower temperature', leading to more deterministic choices by exaggerating differences between action values, or a 'higher temperature', leading to increasingly random choice. The inverse decision temperature β is a free parameter of the model:

$$\mathbf{p}_t = \mathrm{softmax}(\beta \times \mathbf{Q}_t) \tag{3}$$

We call the model based on just equations (1) and (3) 'Basic RL'. With only two free parameters (α and β), a Basic RL model typically does not predict human choices very accurately. Many extensions have been proposed to improve behavioural fit. We focus on three here: perseveration, forgetting and variable learning rates. Perseveration enables action repetition (or switching) independently of rewards and is the simplest form of reward-independent action-history processing. The perseveration term c adds a small bonus (of size $\varkappa$) to the value of the action a that was chosen on the previous time step, but not to all other actions $\neg a$:

$$\begin{aligned} c_t(a) &= \varkappa \\ c_t(\neg a) &= 0 \end{aligned} \tag{4}$$

Q-learning agents that track both perseveration and action values have an additive choice rule. The vectors of action values and perseveration are added (to form 'choice logits' h_t) and pass through the softmax rule for action selection:

$$\mathbf{h}_t = \mathbf{Q}_t + \mathbf{c}_t$$

$$\mathbf{p}_t = \mathrm{softmax}(\beta \times \mathbf{h}_t)$$

Forgetting was implemented as the exponential decay of each action value back to Q_{init}, at which each action value is initialized on the first trial. Q_{init} is a free model parameter that is fitted to participant behaviour. The decay parameter f, a free model parameter, determined the rate of decay. On each trial, all action values underwent forgetting, according to:

$$Q_t(a) = (1 - f) \times Q_t(a) + f \times Q_{\mathrm{init}} \tag{5}$$

Variable learning rates were implemented following a variant of the classic Pearce–Hall learning rule[102], adapted to instrumental tasks[54]. In this model, each trial t's learning rate α_t is updated on the basis of the previous trial's reward prediction error δ_t. The larger the absolute value of δ_t, that is, the greater the 'surprise' about an outcome, the larger the learning rate:

$$\delta_t = r_t - Q_t(a) \tag{6}$$

$$Q_{t+1}(a) = Q_t(a) + \alpha_t \times \delta_t \tag{7}$$

$$\alpha_{t+1} = w \times |\delta_t| + (1 - w) \times \alpha_t \tag{8}$$

w, a free parameter of the model, is a weighting parameter that determines how variable (larger w) versus stable (smaller w) α_t is over time—a learning rate on the learning rate. At $w = 0$, learning rates are stable, and the model reduces to simpler RL model variants. Variable-learning-rate model variants replace the standard learning rate parameter α with α_{init}, the model's initial learning rate on the first trial.

In the main text, we sometimes obliterate the subscript t in equations for better readability. Following common practice, we restricted the ranges of the free parameters of our Q-learning models to ensure interpretability. For example, a negative learning rate or negative forgetting would not be interpretable. We used common transforms (sigmoid, relu and tanh) to enforce the following ranges for RL models' free parameters:

Learning rate / initial learning rate: $0 < \alpha < 1$, $0 < \alpha_{\mathrm{init}} < 1$
Update bias: $-1 < b < 1$
Inverse decision temperature: $0 < \beta < \infty$
Perseveration: $-1 < \neg < 1$
Forgetting: $0 < f < 1$
Weighting parameter: $0 < w < 1$
The initial value Q_{init} was not restricted.

Q-learning model comparison. To identify the best Q-learning model for our data, we performed a systematic model comparison. We created $7^2 - 1 = 48$ model variants based on all parameter combinations. Supplementary Table 2 shows the results for the most relevant subset of model variants. Basic RL included only two parameters, α and β. Best RL included six parameters (α, β, f, $\neg$, b and Q_{init}). We fitted all models to the training split of our dataset, using the methods described in the following sections, and selected the winner on the basis of the model fit on the held-out test data.

RL-ANN architecture. RL-ANN has the same structure as Best RL but contains two neural networks instead of Best RL's value update and perseveration operations (Fig. 2b). We first focus on the value update module, the model's Reward ANN, and then turn to the perseveration network, the model's Action-History ANN. The Reward ANN receives the same inputs as the classic value update (equation (1)), $Q_{t-1}(a)$ and r_{t-1}, and produces the same output, $Q_t(a)$. On each trial t, the Reward ANN's input layer vector $\mathbf{i}_t^{(r)}$ contains the concatenation of its two scalar inputs:

$$\mathbf{i}_t^{(a)} = [Q_{t-1}(a), r_{t-1}]$$

The activations in the hidden layer (the state vector $\mathbf{s}_t^{(r)}$) are obtained by passing the input vector through the first fully-connected layer of the network. Inputs are multiplied with the matrix of weights $W_1^{(r)}$, the bias vector $\mathbf{b}_1^{(r)}$ is added and the result is passed through a tanh nonlinearity:

$$\mathbf{s}_t^{(r)} = tanh\left(W_1^{(r)} \mathbf{i}_t^{(r)} + \mathbf{b}_1^{(r)}\right)$$

The output of the network, $Q_t(a)$, is obtained by passing the state through a second fully connected layer, parameterized by weights $W_2^{(r)}$ and bias $b_2^{(r)}$ (there is no nonlinearity in the second layer; hence, values Q can be interpreted as logits):

$$Q_t(a) = W_2^{(r)} \mathbf{s}_t^{(r)} + b_2^{(r)} \tag{9}$$

Like Best RL, RL-ANN maintains a vector $\mathbf{Q}_t$ over trials, which contains one value per action. $Q_t(a)$ is replaced by the output of equation (9). All actions in $\mathbf{Q}_t$ undergo forgetting according to equation (8). The Reward ANN's input layer has size 2 (containing $Q_{t-1}(a)$ and r_{t-1}), and the output layer has size 1 ($Q_t(a)$). The size of the hidden layer was determined by a hyperparameter sweep (see below).

RL-ANN's Action-History ANN also is a three-layer, fully connected Multi-Layer Perceptron (MLP). The Action-History ANN receives the same input as classic perseveration (equation (4)), a_{t-1}, and returns the same output, a vector $\mathbf{c}_t$ with one perseveration scalar per action. The network is parameterized by weight matrices $W_1^{(a)}$ and $W_2^{(a)}$, and biases $\mathbf{b}_1^{(a)}$ and $\mathbf{b}_2^{(a)}$:

$$i^{(a)}t = a_{t-1}$$

$$\mathbf{s}_t^{(a)} = tanh\left(W_1^{(a)} \times i_t^{(a)} + \mathbf{b}_1^{(a)}\right)$$

$$\mathbf{c}_t = W_2^{(a)} \mathbf{s}_t^{(a)} + \mathbf{b}_2^{(a)}$$

The Action-History ANN's input layer has size 1, and the output layer has size 4 (one per action). The size of the hidden layer was identical to the reward module's hidden layer.

Like before, values $\mathbf{Q}_t$ and perseveration $\mathbf{c}_t$ are combined additively before passing through the softmax for action selection:

$$\mathbf{h}_t = \mathbf{Q}_t + \mathbf{c}_t$$

$$\mathbf{p}_t = \text{softmax}(\mathbf{h}_t)$$

Context-ANN architecture. Context-ANN is an extension of RL-ANN that adds the ability to condition operations on the context (Fig. 3b). Context-ANN represents the reward context with the vector $\mathbf{Q}_{t-1}$ and the action context with the vector $\mathbf{c}_{t-1}$. We chose $\mathbf{Q}_{t-1}$ and $\mathbf{c}_{t-1}$ as context representations because they are the most succinct summaries of the past history and represent all four actions. Conditioning is performed by adding $\mathbf{Q}_{t-1}$ and $\mathbf{c}_{t-1}$ as inputs to the reward module and choice-MLP, respectively. In this way, the networks can learn to modify their operations on the basis of the additional context information (if this is supported by human behaviour):

$$\mathbf{i}_t^{(r)} = [Q_{t-1}(a), r_{t-1}, \mathbf{Q}_{t-1}]$$

$$\mathbf{i}_t^{(a)} = [a_{t-1}, \mathbf{c}_{t-1}]$$

Everything else remains the same as in RL-ANN (see above).

Memory-ANN architecture. Memory-ANN is our winning model. It is an extension of Context-ANN that allows a more flexible context representation. Instead of conditioning on the output vectors $\mathbf{Q}_{t-1}$ and $\mathbf{c}_{t-1}$, Memory-ANN conditions on their precursors, the hidden states $\mathbf{s}_{t-1}^{(r)}$ and $\mathbf{s}_{t-1}^{(a)}$. As a simplification, it removes the dependence on $Q_{t-1}(a)$:

$$\mathbf{i}_t^{(r)} = [r_{t-1}, \mathbf{s}_{t-1}^{(r)}]$$

The remaining processing steps are unchanged:

$$\mathbf{s}_t^{(r)} = \tanh\left(W_1^{(r)}\mathbf{i}_t^{(r)} + \mathbf{b}_1^{(r)}\right)$$

$$Q_t(a) = W_2^{(r)}\mathbf{s}_t^{(r)} + b_2^{(r)}$$

Vanilla RNN model architecture. Vanilla RNN is a basic RNN. On each trial t, the model receives information about the most recent action a_{t-1} and the reward received after choosing this action, r_{t-1}, and returns a vector of choice logits $\mathbf{h}_t$, with one element for each action. Like before, choice logits guide the selection of the next action a_t, after transformation into action probabilities using the softmax function:

$$\mathbf{p}_t = \text{softmax}(\mathbf{h}_t)$$

Vanilla RNN is a simple, fully connected, recurrent three-layer network. It concatenates the inputs $\mathbf{a}_{t-1}$ (a one-hot vector indicating the chosen action with 1 and all others with 0) and r_{t-1} (a scalar) into a joint vector $\mathbf{i}_t$, the input activations of the network:

$$\mathbf{i}_t = [\mathbf{a}_{t-1}, r_{t-1}]$$

The hidden layer (or recurrent state $\mathbf{s}_t$) is obtained by passing the input activations through the first layer of fully connected neurons, parameterized by weight matrix W_1 and biases b_1, in the same way as above:

$$\mathbf{s}_t = \tanh(W_1\mathbf{i}_t + \mathbf{b}_1)$$

The final output, the vector of logits $\mathbf{h}_t$, is the result of passing the state through another fully connected layer, parameterized by weight matrix W_2 and biases b_2:

$$\mathbf{h}_t = W_2\mathbf{s}_t + \mathbf{b}_2$$

Action choices are made like before, by passing choice logits through a softmax function to determine choice probabilities:

$$\mathbf{p}_t = \text{softmax}(\mathbf{h}_t)$$

Model training

Data splits. We randomly split our dataset into three partitions: training (80% (690) of participants; 3,302 blocks), testing (10% (86) of participants; 413 blocks) and validation (10% (86) of participants; 419 blocks). We used the same train–validation–test splits for testing all models. In other words, the same exact sessions went into the training split for each model, a different set of sessions went into the testing set for each model and a third set was used for validation of all models. We did this to ensure that the resulting model fits were comparable between models.

The training data were used to fit the model parameters (for example, α, β, W_1 and b_2) of a wide range of models, including all combinations of all hyperparameters (for example, the number of hidden units; see below). The validation data were used to identify the optimal set of hyperparameters for each model. The test data were used to determine the fit of each selected model (Figs. 2d and 3c). The three-way split was necessary for two reasons. The validation split allowed us to find the best hyperparameters for each model. This ensured that differences in model fits reflected differences between model architectures rather than differences in the optimality of the chosen hyperparameters. For example, we can be sure that no Context-ANN—whatever its hyperparameters—could ever beat Memory-ANN, because there is no Context-ANN that fits the data better than the one we report. The test split was necessary to ensure that models did not overfit to the training data.

Model fitting. All models, both classic variants of Q-learning and neural networks, were trained with the Adam optimizer, using the optax package (https://github.com/google-deepmind/optax) for jax (https://github.com/google/jax). The optimizer learning rate, batch size, number of training steps, weight decay and number of hidden units (if applicable) for each model were determined by a hyperparameter sweep. Each training batch was sampled randomly and with replacement from the training data. We systematically assessed the following space of hyperparameters: learning rate, 1×10^{-3}, 1×10^{-4}, 1×10^{-5}; L2 weight decay, 1×10^{-3}, 1×10^{-4}, 1×10^{-5}; number of the hidden units, 16, 32, 64; batch size, 32, 64, 128. We trained each model for 1,000,000 steps on the training data, using five instantiations of each combination of hyperparameters, and identified the number of training steps (≤1,000,000) and hyperparameters that led to the best fit on the validation data. The chosen hyperparameters for each model are shown in Supplementary Table 1.

Fitting objective. The goal of training was to create models that behave as similarly as possible to humans (rather than to perform the task as well as possible). We followed standard practices[41] to achieve this. We minimized the negative log-likelihood loss (also called cross-entropy) of each model with respect to the training data. This loss incentivizes model parameters that maximize the (log) probability of jointly predicting the choices $a_{\{t,i\}}$ of each participant p on each trial t in a training batch (of size bs), by following stochastic gradient descent over training batches:

$$L = -\sum_{i=1}^{\text{bs}}\sum_{t=1}^{n_{\text{trials}}} log(p(a_{t,i}))$$

The optimal batch size bs was determined individually for each model on the basis of a hyperparameter sweep (see above). Each task had $n_{\mathrm{trials}} = 150$.

To obtain the final fit for each model (Figs. 2d and 3c), we calculated the loss of the variant with the best hyperparameters on the held-out test data. We calculated the loss separately for each task block, so that we could assess the variability between participants. We also transformed model losses into the trial-wise prediction accuracy, an estimate of what percentage of human choices are predicted accurately:

$$\mathrm{acc} = \exp\left(\frac{-L}{bs \times n_{trials}}\right)$$

Model analysis

Qualitative model fit. We created a synthetic dataset for each model, using the hyperparameters (for example, batch size; Supplementary Table 1) and parameters (for example, learning rate α and connection weights W_1) we obtained in model fitting. We simulated behaviour on the same 4,134 tasks (with the same reward schedules) as human participants, using 'open-loop' simulation (which means that human choices are unknown to the behaving models). We then subjected human and model behaviour to the same statistical analyses to uncover qualitative similarities and differences (Fig. 4).

Model dynamics. We also created 'closed-loop' simulations for each model. Also called 'teacher forcing', this means that a model is forced to make the same choices as a participant. The model does not sample its action from the action probabilities it calculates on each trial but instead automatically selects the teacher's choice. We used this method to inspect the internal dynamics (for example, trial-by-trial trajectories of values Q and choice kernel c or memory states $\mathbf{s}$) that our models assigned to individual participants (Supplementary Fig. 5).

Model inspection. The reward module (described above) determines how observed rewards r_{t-1} map onto values Q_t. We analysed this mapping by probing reward modules with the full range of inputs and measuring their output (Fig. 2e). We first extracted the relevant parameters ($W_1^{(r)}$, $W_2^{(r)}$, $\mathbf{b}_1^{(r)}$ and $\mathbf{b}_2^{(r)}$) from the fitted model (RL-ANN or Memory-ANN). We then initialized a new MLP with the same shape as the original reward module (for example, for Memory-ANN: 2 input units, 32 hidden units and 1 output unit) and injected the fitted parameters. We uniformly sampled rewards r_{t-1} between 1 and 100 points. For RL-ANN, we also sampled values $Q_{t-1}(a)$ between the 10% and 90% quantiles of the values observed in the closed-loop dataset. For Memory-ANN, we sampled hidden state vectors $\mathbf{s}_{t-1}^{(r)}$ along the first (or a different) principal component of the hidden states visited in the closed-loop data; samples were taken up to 1.5 standard deviations from the mean. We finally collected the outputs $Q_t(a)$ of this MLP in response to each combination of inputs.

The same method was used to analyse the action-history module. We obtained the corresponding fitted parameters ($W_1^{(a)}$, $W_2^{(a)}$, $\mathbf{b}_1^{(a)}$ and $\mathbf{b}_2^{(a)}$) and injected them into a newly initialized MLP. We sampled actions a uniformly; for Memory-ANN, we also sampled hidden state vectors $\mathbf{s}_{t-1}^{(r)}$, using the same method as above. We then collected the output $\mathbf{c}_t^{(a)}$ of the network and visualized the relationship between inputs and outputs (Supplementary Fig. 6).

To assess the contents of $\mathbf{s}_t^{(r)}$ (Fig. 3j,k), we calculated a separate regression model for each delay i, predicting the reward observed on trial $t - i$ on the basis of a PC of the current state $\mathbf{s}_t^{(r)}$. We repeated this analysis individually for each PC.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The dataset generated for this study is available via the Open Science Framework at https://osf.io/8xz3w/.

Code availability

The code generated for this study is available via GitHub at https://github.com/google-deepmind/hybrid_rnns_reward_learning. Please refer to the Colab provided for a simple demo of how to fit the models presented here.

References

1. Rescorla, R. A. & Wagner, A. R. in *Classical Conditioning II: Current Research and Theory* (eds Black, A. H. & Prokasy, W. F.) Vol. 2, 64–99 (Appleton-Century-Crofts, 1972).
2. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* 2nd edn (MIT Press, 2017).
3. Erev, I. & Roth, A. E. Predicting how people play games: reinforcement learning in experimental games with unique, mixed strategy equilibria. *Am. Econ. Rev.* **88**, 848–881 (1998).
4. Lee, D., McGreevy, B. P. & Barraclough, D. J. Learning and decision making in monkeys during a rock–paper–scissors game. *Cogn. Brain Res.* **25**, 416–430 (2005).
5. Miller, K. J., Botvinick, M. M. & Brody, C. D. From predictive models to cognitive models: separable behavioral processes underlying reward learning in the rat. Preprint at *bioRxiv* https://doi.org/10.1101/461129 (2018).
6. Eckstein, M. K. & Collins, A. G. E. Computational evidence for hierarchically structured reinforcement learning in humans. *Proc. Natl Acad. Sci. USA* **117**, 29381–29389 (2020).
7. Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P. & Dolan, R. J. Model-based influences on humans' choices and striatal prediction errors. *Neuron* **69**, 1204–1215 (2011).
8. Frank, M. J. & Badre, D. Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: computational analysis. *Cereb. Cortex* **22**, 509–526 (2012).
9. Collins, A. G. E. & Koechlin, E. Reasoning, learning, and creativity: frontal lobe function and human decision-making. *PLoS Biol.* **10**, e1001293 (2012).
10. Doya, K. Reinforcement learning: computational theory and biological mechanisms. *HFSP J.* **1**, 30–40 (2007).
11. O'Doherty, J. P., Hampton, A. & Kim, H. Model-based fMRI and its application to reward learning and decision making. *Ann. N. Y. Acad. Sci.* **1104**, 35–53 (2007).
12. Lee, D., Seo, H. & Jung, M. W. Neural basis of reinforcement learning and decision making. *Annu. Rev. Neurosci.* **35**, 287–308 (2012).
13. Duncan, K. D. & Shohamy, D. Memory states influence value-based decisions. *J. Exp. Psychol. Gen.* **145**, 1420–1426 (2016).
14. Plonsky, O., Teodorescu, K. & Erev, I. Reliance on small samples, the wavy recency effect, and similarity-based learning. *Psychol. Rev.* **122**, 621–647 (2015).
15. Schulz, E. & Gershman, S. J. The algorithmic architecture of exploration in the human brain. *Curr. Opin. Neurobiol.* **55**, 7–14 (2019).
16. Gershman, S. J. & Daw, N. D. Reinforcement learning and episodic memory in humans and animals: an integrative framework. *Annu. Rev. Psychol.* **68**, 101–128 (2017).
17. Bornstein, A. M. & Norman, K. A. Reinstated episodic context guides sampling-based decisions for reward. *Nat. Neurosci.* **20**, 997–1003 (2017).
18. Palminteri, S., Khamassi, M., Joffily, M. & Coricelli, G. Contextual modulation of value signals in reward and punishment learning. *Nat. Commun.* **6**, 8096 (2015).

19. Lefebvre, G., Lebreton, M., Meyniel, F., Bourgeois-Gironde, S. & Palminteri, S. Behavioural and neural characterization of optimistic reinforcement learning. *Nat. Hum. Behav.* **1**, 0067 (2017).
20. Louie, K., Khaw, M. W. & Glimcher, P. W. Normalization is a general neural mechanism for context-dependent decision making. *Proc. Natl Acad. Sci. USA* **110**, 6139–6144 (2013).
21. Khaw, M. W., Glimcher, P. W. & Louie, K. Normalized value coding explains dynamic adaptation in the human valuation process. *Proc. Natl Acad. Sci. USA* **114**, 12696–12701 (2017).
22. Yaple, Z. A. & Yu, R. Fractionating adaptive learning: a meta-analysis of the reversal learning paradigm. *Neurosci. Biobehav. Rev.* **102**, 85–94 (2019).
23. Gerraty, R. T. et al. Dynamic flexibility in striatal–cortical circuits supports reinforcement learning. *J. Neurosci.* **38**, 2442–2453 (2018).
24. Langdon, A. J., Sharpe, M. J., Schoenbaum, G. & Niv, Y. Model-based predictions for dopamine. *Curr. Opin. Neurobiol.* **49**, 1–7 (2018).
25. Coddington, L. T. & Dudman, J. T. The timing of action determines reward prediction signals in identified midbrain dopamine neurons. *Nat. Neurosci.* **21**, 1563–1573 (2018).
26. Engelhard, B. et al. Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons. *Nature* **570**, 509–513 (2019).
27. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
28. Dezfouli, A., Griffiths, K., Ramos, F., Dayan, P. & Balleine, B. W. Models that learn how humans learn: the case of decision-making and its disorders. *PLoS Comput. Biol.* **15**, e1006903 (2019).
29. Fintz, M., Osadchy, M. & Hertz, U. Using deep learning to predict human decisions and using cognitive models to explain deep learning models. *Sci. Rep.* **12**, 4736 (2022).
30. Ger, Y., Shahar, M. & Shahar, N. Using recurrent neural network to estimate irreducible stochasticity in human choice behavior. *eLife* **13**, e90082 (2024).
31. Song, M., Niv, Y. & Cai, M. Using recurrent neural networks to understand human reward learning. *Proc. Annu. Meet. Cogn. Sci. Soc.* **43**, 1388–1394 (2021).
32. Agrawal, M., Peterson, J. C. & Griffiths, T. L. Scaling up psychology via scientific regret minimization. *Proc. Natl Acad. Sci. USA* **117**, 8825–8835 (2020).
33. Kuperwajs, I., Schütt, H. H. & Ma, W. J. Using deep neural networks as a guide for modeling human planning. *Sci. Rep.* **13**, 20269 (2023).
34. Botvinick, M. M. & Plaut, D. C. Short-term memory for serial order: a recurrent neural network model. *Psychol. Rev.* **113**, 201–233 (2006).
35. Sussillo, D. & Barak, O. Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* **25**, 626–649 (2013).
36. Peterson, J. C., Bourgin, D. D., Agrawal, M., Reichman, D. & Griffiths, T. L. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science* **372**, 1209–1214 (2021).
37. Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B. & Dolan, R. J. Cortical substrates for exploratory decisions in humans. *Nature* **441**, 876–879 (2006).
38. Gillan, C. M., Kosinski, M., Whelan, R., Phelps, E. A. & Daw, N. D. Characterizing a psychiatric symptom dimension related to deficits in goal-directed control. *eLife* **5**, e11305 (2016).
39. Hunter, L. E., Meer, E. A., Gillan, C. M., Hsu, M. & Daw, N. D. Increased and biased deliberation in social anxiety. *Nat. Hum. Behav.* **6**, 146–154 (2022).
40. Daw, N. D. "Trial-by-trial data analysis using computational models" in Mauricio R. Delgado, Elizabeth A. Phelps, and Trevor W. Robbins (eds). *Decision Making, Affect, and Learning* 3–38 (2011), Oxford, 2011.
41. Wilson, R. C. & Collins, A. G. Ten simple rules for the computational modeling of behavioral data. *eLife* **8**, e49547 (2019).
42. Ito, M. & Doya, K. Validation of decision-making models and analysis of decision variables in the rat basal ganglia. *J. Neurosci.* **29**, 9861–9874 (2009).
43. Miller, K. J., Shenhav, A. & Ludvig, E. A. Habits without values. *Psychol. Rev.* **126**, 292–311 (2019).
44. Lau, B. & Glimcher, P. W. Dynamic response-by-response models of matching behavior in rhesus monkeys. *J. Exp. Anal. Behav.* **84**, 555–579 (2005).
45. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
46. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. Attention Is All You Need. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), 5998-6008 (2017).
47. Söderström, T. & Stoica, P. *System Identification* (Prentice Hall, 1989).
48. Sammut, C., Hurst, S., Kedzier, D. & Michie, D. in *Machine Learning Proceedings 1992* (eds Sleeman, D. & Edwards, P.) 385–393 (Morgan Kaufmann, 1992).
49. Argall, B. D., Chernova, S., Veloso, M. & Browning, B. A survey of robot learning from demonstration. *Rob. Auton. Syst.* **57**, 469–483 (2009).
50. Frank, M. J., Seeberger, L. C. & O'Reilly, R. C. By carrot or by stick: cognitive reinforcement learning in parkinsonism. *Science* **306**, 1940–1943 (2004).
51. Rosas, J. M., Todd, T. P. & Bouton, M. E. Context change and associative learning. *WIREs Cogn. Sci.* **4**, 237–244 (2013).
52. Klein, T. A., Ullsperger, M. & Jocham, G. Learning relative values in the striatum induces violations of normative decision making. *Nat. Commun.* **8**, 16033 (2017).
53. Collins, A. G. E. & Frank, M. J. How much of reinforcement learning is working memory, not reinforcement learning? A behavioral, computational, and neurogenetic analysis. *Eur. J. Neurosci.* **35**, 1024–1035 (2012).
54. Li, J., Schiller, D., Schoenbaum, G., Phelps, E. A. & Daw, N. D. Differential roles of human striatum and amygdala in associative learning. *Nat. Neurosci.* **14**, 1250–1252 (2011).
55. Palminteri, S., Wyart, V. & Koechlin, E. The importance of falsification in computational cognitive modeling. *Trends Cogn. Sci.* **21**, 425–433 (2017).
56. Kovach, C. K. et al. Anterior prefrontal cortex contributes to action selection through tracking of recent reward trends. *J. Neurosci.* **32**, 8434–8442 (2012).
57. Wittmann, M. K. et al. Predictive decision making driven by multiple time-linked reward representations in the anterior cingulate cortex. *Nat. Commun.* **7**, 12327 (2016).
58. Schönberg, T., Daw, N. D., Joel, D. & O'Doherty, J. P. Reinforcement learning signals in the human striatum distinguish learners from nonlearners during reward-based decision making. *J. Neurosci.* **27**, 12860–12867 (2007).
59. Ebitz, R. B., Albarran, E. & Moore, T. Exploration disrupts choice-predictive signals and alters dynamics in prefrontal cortex. *Neuron* **97**, 475 (2018).
60. Lee, D., Conroy, M. L., McGreevy, B. P. & Barraclough, D. J. Reinforcement learning and decision making in monkeys during a competitive game. *Brain Res. Cogn. Brain Res.* **22**, 45–58 (2004).
61. Tuzsus, D., Brands, A., Pappas, I. & Peters, J. Exploration–exploitation mechanisms in recurrent neural networks and human learners in restless bandit problems. *Comput. Brain Behav.* **7**, 314–356 (2024).
62. Seymour, B. & McClure, S. M. Anchors, scales and the relative coding of value in the brain. *Curr. Opin. Neurobiol.* **18**, 173–178 (2008).

63. Rangel, A. & Clithero, J. A. Value normalization in decision making: theory and evidence. *Curr. Opin. Neurobiol.* **22**, 970–981 (2012).
64. Collins, A.G.E. A habit and working memory model as an alternative account of human reward-based learning. Nat Hum Behav https://doi.org/10.1038/s41562-025-02340-0 (2025).
65. Lengyel, M. & Dayan, P. Hippocampal contributions to control: the third way. *Adv. Neural Inf. Process. Syst.* **21**, 889–896 (2007).
66. Miller, J. A. & Constantinidis, C. Timescales of learning in prefrontal cortex. *Nat. Rev. Neurosci.* https://doi.org/10.1038/s41583-024-00836-8 (2024).
67. Spitmaan, M., Seo, H., Lee, D. & Soltani, A. Multiple timescales of neural dynamics and integration of task-relevant signals across cortex. *Proc. Natl Acad. Sci. USA* **117**, 22522–22531 (2020).
68. Dabney, W. et al. A distributional code for value in dopamine-based reinforcement learning. *Nature* **577**, 671–675 (2020).
69. Gershman, S. J. A unifying probabilistic view of associative learning. *PLoS Comput. Biol.* **11**, e1004567 (2015).
70. Beron, C., Neufeld, S., Linderman, S. & Sabatini, B. Efficient and stochastic mouse action switching during probabilistic decision making. *Neuroscience* **10**, 13–444094 (2021).
71. Behrens, T. E. J., Woolrich, M. W., Walton, M. E. & Rushworth, M. F. S. Learning the value of information in an uncertain world. *Nat. Neurosci.* **10**, 1214–1221 (2007).
72. Piray, P. & Daw, N. D. A simple model for learning in volatile environments. *PLoS Comput. Biol.* **16**, e1007963 (2020).
73. Joel, D., Niv, Y. & Ruppin, E. Actor-critic models of the basal ganglia: new anatomical and computational perspectives. *Neural Netw.* **15**, 535–547 (2002).
74. Chen, R. & Goldberg, J. H. Actor-critic reinforcement learning in the songbird. *Curr. Opin. Neurobiol.* **65**, 1–9 (2020).
75. Bavard, S., Lebreton, M., Khamassi, M., Coricelli, G. & Palminteri, S. Reference-point centering and range-adaptation enhance human reinforcement learning at the cost of irrational preferences. *Nat. Commun.* **9**, 4503 (2018).
76. Thorndike, E. L. *Animal Intelligence: Experimental Studies* (Macmillan, 1911).
77. Ashby, F. G., Ennis, J. M. & Spiering, B. J. A neurobiological theory of automaticity in perceptual categorization. *Psychol. Rev.* **114**, 632–656 (2007).
78. Balleine, B. W. & O'Doherty, J. P. Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action. *Neuropsychopharmacology* **35**, 48–69 (2010).
79. Akaishi, R., Umeda, K., Nagase, A. & Sakai, K. Autonomous mechanism of internal choice estimate underlies decision inertia. *Neuron* **81**, 195–206 (2014).
80. Greenstreet, F. et al. Dopaminergic action prediction errors serve as a value-free teaching signal. *Nature* **643**, 1333–1342 (2025).
81. Lebedeva, A. et al. Dorsal prefrontal cortex drives perseverative behavior in mice. Preprint at *bioRxiv* https://doi.org/10.1101/2024.05.02.592241 (2024).
82. Botvinick, M. M., Niv, Y. & Barto, A. C. Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition* **113**, 262–280 (2009).
83. Ho, M. K., Abel, D., Griffiths, T. L. & Littman, M. L. The value of abstraction. *Curr. Opin. Behav. Sci.* **29**, 111–116 (2019).
84. Badre, D. & Nee, D. E. Frontal cortex and the hierarchical control of behavior. *Trends Cogn. Sci.* **22**, 170–188 (2018).
85. Tomov, M. S., Yagati, S., Kumar, A., Yang, W. & Gershman, S. J. Discovery of hierarchical representations for efficient planning. *PLoS Comput. Biol.* **16**, e1007594 (2020).
86. Badre, D. & Frank, M. J. Mechanisms of hierarchical reinforcement learning in cortico-striatal circuits 2: evidence from fMRI. *Cereb. Cortex* **22**, 527–536 (2012).
87. Alexander, W. H. & Brown, J. W. Frontal cortex function as derived from hierarchical predictive coding. *Sci. Rep.* **8**, 3843 (2018).
88. Dezfouli, A. et al. Disentangled behavioural representations. *Adv. Neural Inf. Process. Syst.* **32**, 2254–2263 (2019).
89. Katahira, K. Excessive flexibility? Recurrent neural networks can accommodate individual differences in reinforcement learning by capturing higher-order history dependencies. *Comput. Brain Behav.* https://doi.org/10.1007/s42113-025-00254-8 (2025).
90. Navarro, D. J. Between the devil and the deep blue sea: tensions between scientific judgement and statistical model selection. *Comput Brain Behav.* **2**, 28–34 (2019).
91. Nassar, M. R. & Frank, M. J. Taming the beast: extracting generalizable knowledge from computational models of cognition. *Curr. Opin. Behav. Sci.* **11**, 49–54 (2016).
92. Ji-An, L., Benna, M. K. & Mattar, M. G. Discovering cognitive strategies with tiny recurrent neural networks. *Nature* **644**, 993–1001 (2025).
93. Miller, K. J., Eckstein, M., Botvinick, M. & Kurth-Nelson, Z. Cognitive model discovery via disentangled RNNs. *Adv. Neural Inf. Process. Syst.* **36**, 61377–61394 (2024).
94. Castro, P. S. et al. Discovering symbolic cognitive models from human and animal behavior. Proceedings of the 42nd International Conference on Machine Learning, in Proceedings of Machine Learning Research 267:6849-6890 (2025).
95. Binz, M. et al. A foundation model to predict and capture human cognition. *Nature* **644**, 1002–1009 (2025).
96. Dubois, M. & Hauser, T. U. Value-free random exploration is linked to impulsivity. *Nat. Commun.* **13**, 4542 (2022).
97. Zorowitz, S., Solis, J., Niv, Y. & Bennett, D. Inattentive responding can induce spurious associations between task behaviour and symptom measures. *Nat. Hum. Behav.* **7**, 1667–1681 (2023).
98. de Leeuw, J. R. jsPsych: a JavaScript library for creating behavioral experiments in a Web browser. *Behav. Res. Methods* **47**, 1–12 (2015).
99. Bahrami, B. & Navajas, J. 4 arm bandit task dataset. *OSF* https://doi.org/10.17605/OSF.IO/F3T2A (2020).
100. Ziv, J. & Lempel, A. Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory* **24**, 530–536 (1978).
101. Welch, T. A. A technique for high-performance data compression. *Computer* (June 1984).
102. Pearce, J. M. & Hall, G. A model for Pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychol. Rev.* **87**, 532–552 (1980).

Acknowledgements

We thank P. Humphreys, G. Wayne, Z. Kurth-Nelson and J. Wang for fruitful discussions about this project, and P. Dayan, A. Collins, S. Chan, M. Mozer, K. Stachenfeld, N. Éltető and M. Botvinick for helpful comments on an earlier version of this manuscript. We acknowledge the Wellcome Trust Discovery Award No. 227928/Z/23/Z to C.S. The funder had no role in study design, data collection and analysis, decision to publish or preparation of the manuscript.

Author contributions

Conceptualization: M.K.E. and K.J.M. Experiment design: M.K.E., K.J.M. and N.D.D. Formal analysis: M.K.E. Methodology: M.K.E., C.S., N.D.D. and K.J.M. Visualization: M.K.E. Project administration: M.K.E. Model conception and implementation: M.K.E. Writing—original draft: M.K.E., C.S., N.D.D. and K.J.M. Writing—review and editing: M.K.E., C.S., N.D.D. and K.J.M.

Competing interests

M.K.E., N.D.D. and K.J.M. are employed by Google DeepMind. C.S. declares no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41562-025-02324-0.

Correspondence and requests for materials should be addressed to Maria K. Eckstein or Kevin J. Miller.

Peer review information *Nature Human Behaviour* thanks Daeyeol Lee and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Maria Eckstein, Kevin Miller

Last updated by author(s): Oct 18, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
□	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
□	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
□	☒	A description of all covariates tested
□	☒	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
□	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
□	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
□	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	jsPsych
Data analysis	python 3.5

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The dataset generated for this study is available in the Open Science Framework repository at https://osf.io/8xz3w/files/osfstorage
The code written for this study is openly accessible at: https://github.com/google-deepmind/hybrid_rnns_reward_learning/tree/main/hybrid_rnns_reward_learning

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender

Our analyses did not focus on gender and/or sex differences. Both males and females were recruited equally to participate in the study.

Reporting on race, ethnicity, or other socially relevant groupings

No reporting was done on race, ethnicity, or other socially relevant groupings.

Population characteristics

Our study did not focus on population characteristics or individual differences.

Recruitment

Participants were recruited using the standard approach of Prolific.com. The only restrictions were the limitation to adulthood (age >= 18), located in the UK, and mastery of the English language.

Ethics oversight

HuBREC, Google DeepMind's Human Behavioural Research Committee

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☐ Life sciences ☒ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description

We collected a large dataset from a reward-learning task in which human participants repeatedly chose among four possible actions, which were rewarded according to noisy reward magnitudes that drifted over time (a non-stationary 'bandit' task; Fig. 1E; Daw et al., 2006). On each trial of the task, participants selected one of the four actions and were given the corresponding reward. We collected a large dataset online (880 participants, 862 of whom passed inclusion criteria; 4,134 task blocks; 617,871 valid trials). The dataset is quantitative.

Research sample

We recruited 880 participants on Prolific (app.prolific.co). No statistical methods were used to pre-determine the sample size but our sample size is orders of magnitude larger than most traditional lab-based human experimental studies, and similar in size to those reported in previous publications focused on large-scale experiments. In agreement with the ethical guidelines of the Google DeepMind Human Behavioral Research Committee (HuBReC), all participants were local to the UK and fluent in English. Participants provided informed consent and were paid at a rate of 12 pounds per hour; there was no performance-based bonus payment. The study was not preregistered. Dedicated studies have shown that prolific samples rank high in terms of representativeness. Eighty participants were asked to complete one training and three testing blocks of 150 trials each. The remaining 800 participants were asked to complete one training block of 50 trials and five testing blocks of 150 trials, for a total of 4,240 task blocks. Four participants in the first (5%) and 14 participants (1.75%) in the second sample failed to finish the experiment and were excluded, leading to an initial sample of 880 - 18 = 862 participants who collectively finished (80 - 4) * 3 + (800 - 14) * 5 = 4,158 task blocks. We further excluded blocks in which participants missed more than 15 of the 150 trials (10%), 24 blocks in total (0.58%). Hence, our final dataset comprised 4,134 blocks (with 617,871 valid trials) from 862 participants. Of these 862 participants, 858 (99.5%) provided valid demographic information: 341 (39.7%) were female and 517 (60.3%) were male; the average age was 39.7 years, with a range of 18 - 88 and a standard deviation of 13.1 years.

Sampling strategy

Our sample size was chosen to be similar to the largest datasets in the literature concerning similar tasks. Our planned analysis methods (based on neural networks) required a larger dataset than typical in the field. Whereas most existing studies collect on the order of 30-50 data points, our study contains more than 4.000.

Data collection

Participants took the study on a computer in the privacy of their own homes, using the prolific platform. No researchers were present with research participants.

Timing

All participants were collected within approximately two days on 2023-04-20.

Data exclusions

Eighty participants were asked to complete one training and three testing blocks of 150 trials each. The remaining 800 participants were asked to complete one training block of 50 trials and five testing blocks of 150 trials, for a total for 4,240 task blocks. Four participants in the first (5\%) and 14 participants (1.75\%) in the second sample failed to finish the experiment and were excluded, leading to an initial dataset of $(80-4) \cdot 3+(800-14) \cdot 5=4,158$ task blocks. We further excluded blocks in which participants missed more than 15 of the 150 trials (10\%), 24 blocks in total (0.58\%). Hence, our final dataset comprised 4,134 blocks (comprising 617,871 valid trials) from 862 participants. Of these 862 participants, 858 (99.5\%) provided valid demographic information: 341

(39.7\%) were female and 517 (60.3\%) were male; the average age was 39.7 years, with a range of 18-88 and a standard deviation of 13.1 years.

Non-participation

171 dropped out of the study. Reasons included running out of time (which was provided generously) and voluntarily dropping the task. Participants did not face negative consequences for opting out of the study.

Randomization

NA.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study	
☒	□	Antibodies
☒	□	Eukaryotic cell lines
☒	□	Palaeontology and archaeology
☒	□	Animals and other organisms
☒	□	Clinical data
☒	□	Dual use research of concern
☒	□	Plants

Methods

n/a	Involved in the study	
☒	□	ChIP-seq
☒	□	Flow cytometry
☒	□	MRI-based neuroimaging

Plants

Seed stocks

NA

Novel plant genotypes

NA

Authentication

NA