nature neuroscience

Article

https://doi.org/10.1038/s41593-024-01689-1

A feature-specific prediction error model explains dopaminergic heterogeneity

Received: 8 February 2022

Accepted: 22 May 2024

Published online: 03 July 2024

Check for updates

Rachel S. Lee[1], Yotam Sagiv[1], Ben Engelhard[1], Ilana B. Witten[1] ✉ & Nathaniel D. Daw[1,2] ✉

The hypothesis that midbrain dopamine (DA) neurons broadcast a reward prediction error (RPE) is among the great successes of computational neuroscience. However, recent results contradict a core aspect of this theory: specifically that the neurons convey a scalar, homogeneous signal. While the predominant family of extensions to the RPE model replicates the classic model in multiple parallel circuits, we argue that these models are ill suited to explain reports of heterogeneity in task variable encoding across DA neurons. Instead, we introduce a complementary 'feature-specific RPE' model, positing that individual ventral tegmental area DA neurons report RPEs for different aspects of an animal's moment-to-moment situation. Further, we show how our framework can be extended to explain patterns of heterogeneity in action responses reported among substantia nigra pars compacta DA neurons. This theory reconciles new observations of DA heterogeneity with classic ideas about RPE coding while also providing a new perspective of how the brain performs reinforcement learning in high-dimensional environments.

Among the more prominent hypotheses in computational neuroscience is that phasic responses from midbrain dopamine (DA) neurons report a reward prediction error (RPE)[1,2]. This account has impressive range, connecting neural events to behavior via interpretable computations over formally defined decision variables (Fig. 1a). Here, we ask whether and how the strengths of this account can be reconciled with a growing body of evidence challenging a core feature of the model: the assumption of a scalar, globally broadcast RPE signal in DA responses.

This scalar RPE is not a superficial claim of these theories but instead connects a key computational idea to several empirical observations. Computationally, scalar decision variables reflect the requirement that a decision-maker compare potential outcomes to choose which to take. Anatomically, the ascending DA projection has been argued to support a scalar 'broadcast' code: a relatively small number of neurons innervate a large area of forebrain via diffuse projections[3,4]. Physiologically, early reports stressed the homogeneity of responses of midbrain DA neurons on simple conditioning tasks, especially that most units respond to unexpected reward[5].

However, the physiological argument for a scalar RPE is increasingly untenable, as a body of recent work demonstrates a range of variation in DA responses. DA neurons can have heterogeneous and specialized responses to task variables during complex behavior[6–22], even while often having relatively homogenous responses to reward[1,9,15,23,24]. In some areas, the signature response to reward can itself be absent[6,10,16,25,26].

How can we reconcile such dopaminergic (DAergic) heterogeneity with evidence for the RPE theory? The majority of previous computational approaches to this question shares a common hypothesis, specifically that a set of different error signals could each serve to train different target functions. For example, distributional reinforcement learning (RL) suggests prediction errors (PEs) for different quantiles of reward expectation[27], action PE (APE) models envision PEs for movements[28–30], and successor representation (SR) models suggest PEs for different sensory stimuli[31]. We refer to these models as 'outcome-specific PEs' because they typically substitute different prediction targets in place of the reward outcome in classic RPE (Fig. 1b).

Two implications of these models limit their applicability. First, substituting different targets for reward means that these models cannot explain why most DA neurons respond to reward nor how such uniformity in reward responses can coexist with heterogeneity in other

[1]Princeton Neuroscience Institute, Princeton, NJ, USA. [2]Department of Psychology, Princeton University, Princeton, NJ, USA.
✉e-mail: iwitten@princeton.edu; ndaw@princeton.edu

https://doi.org/10.1038/s41593-024-01689-1

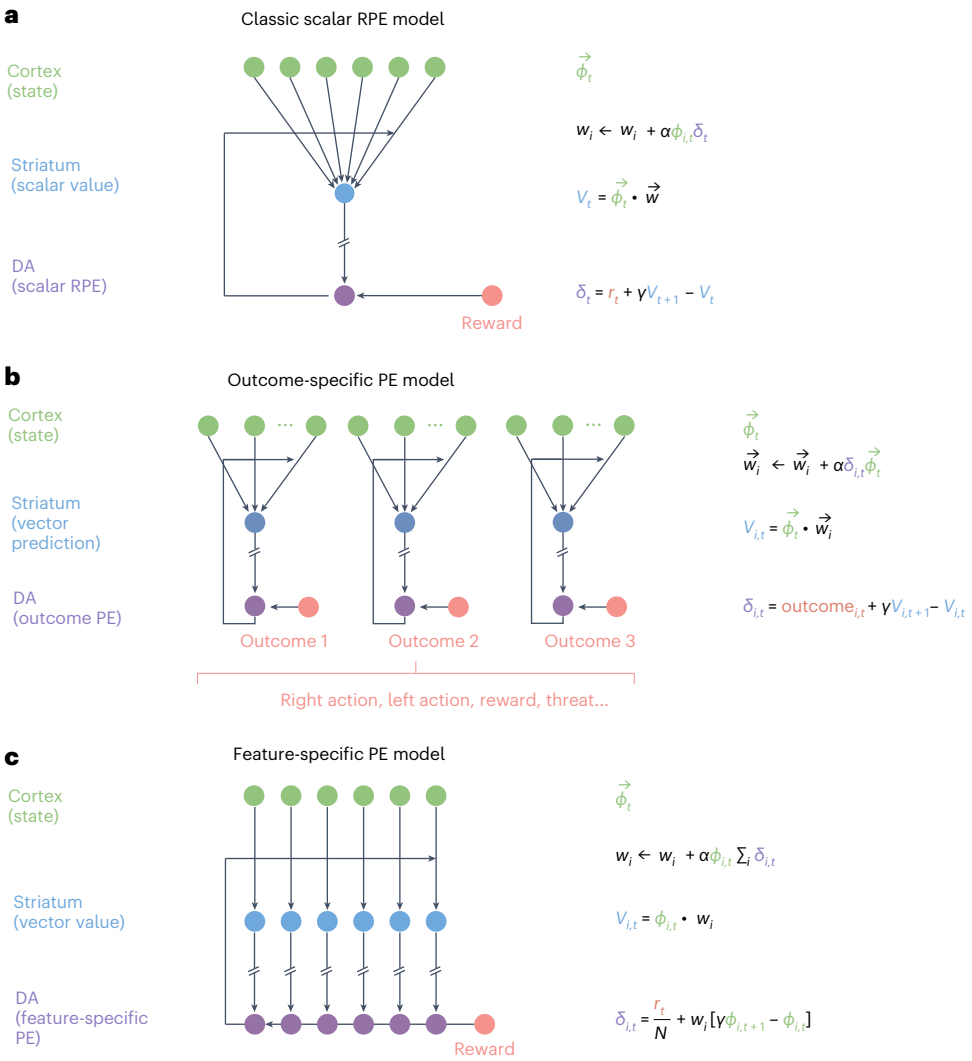


Fig. 1 | Feature-specific PE model updates classic TD learning to produce heterogeneous DA signals that reflect the state representation. a, Classic mapping between equations of the TD learning model and brain circuitry for the scalar RPE model[1,2]. **b**, The same scalar model can be replicated to use different outcome variables to generate vector values and PEs. These outcomes can be driven by different aspects of reward[27,41,70] (like in distributional RL models), specific actions (such as nosepoke, left and right action)[28–30] or other targets[31,43–45]. **c**, Our proposed feature-specific PE model, which remaps the scalar RPE algorithm onto brain circuitry such that value and PE are vectors but the overall computations are preserved.

aspects of responding[9]. Second, this approach amounts to duplicating the original scalar RPE model (Fig. 1a) for multiple error signals in parallel circuits (Fig. 1b). That different targets must be associated in roughly closed-loop fashion with distinct PEs suggests that DA neurons for different predictions must project to different targets.

We thus propose a complementary new 'feature-specific PE' model that addresses a distinct set of phenomena associated with heterogeneity within a projection-defined DA population rather than across projections (Fig. 1c). Within any PE circuit (Fig. 1a,b, classic RPE and any outcome-specific siblings, respectively), the predictions depend on an input variable known as 'state', a high-dimensional group of all sensory and internal variables relevant to the prediction. Because these are widely distributed throughout the brain, it is implausible that they converge uniformly on DA neurons to produce a scalar PE. Inputs to DA neurons are not homogenous but instead arise from cortico–basal–ganglionic circuits that are highly topographic[32–38]. Thus, in the feature-specific model, each DA neuron calculates a PE based on the subset of inputs that it receives, with the full PE signal reconstructed at the target site (Fig. 1c). In this way, corticostriatal circuits carry a distributed code for state, which transforms it into corresponding distributed codes for value and RPE that, in effect, decompose these scalar variables over state features[33]. Different striatal and DAergic neurons reflect the contribution of different state features to value and RPE, but the ensemble collectively represents canonical RL computations over the scalar variables. In turn, this circuit may be repeated between different targets, producing distinct aspects of heterogeneity between versus within projection-defined populations.

Here, we initially investigate the feature-specific PE model by focusing on a recent study from our labs[9], which provides one of the most detailed examples of variability between single neurons within the ventral tegmental area (VTA), the nucleus classically most identified with an RPE. We recorded DA neurons while mice performed an evidence accumulation task in a virtual reality (VR) T-maze. Although neurons respond relatively homogeneously to reward, during the preceding cue period, they respond heterogeneously to many task features[9]. Our feature-specific RPE model explains these data, whereas outcome-specific models cannot. We also demonstrate that our model can be integrated with the outcome-specific PE framework to explain aspects of substantia nigra pars compacta (SNc) DA neuron heterogeneity[26].

Our model offers several key insights. First, the contrast between the outcome- and feature-specific models predicts, testably, that distinct aspects of DAergic heterogeneity predominate between versus within projection-defined DA populations. Second, within the canonical VTA RPE circuit, the feature-specific model explains a striking contrast, specifically heterogeneous responses to task variables alongside uniform outcome period responses[9,15]. This is a key empirical signature of the distributed value code but is poorly explained by the earlier theories. Third, the model retains an algebraic mapping to the standard theory[1,2], preserving its successes while improving its match to data. Finally, the theory connects the puzzling empirical phenomena of DAergic heterogeneity to a major theoretical question in RL models, specifically the nature of state. The new model suggests that DA population heterogeneity provides an empirical window into how the brain represents task state.

Results

The feature-specific PE model

We consider extensions to the classic scalar RPE model (Fig. 1a). The classic model aims to learn the value function $V(s_t) = E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t]$ (the expected sum of γ-discounted rewards r_t starting in state s_t). In neuroscience and artificial intelligence, a typical assumption for high-dimensional tasks is that the learner approximates value linearly in some feature basis. That is, it represents the state s_t by a vector of features $\vec{\phi}(s_t)$ (henceforth $\vec{\phi}_t$) and approximates value as a weighted sum of those features, $V(s_t) \approx \vec{\phi}_t \cdot \vec{w}$. This reduces the problem of value learning (for some feature set) to learning the weights $\vec{w}$ and formalizes the state representation as a vector of time-varying features $\vec{\phi}_t$. The linearity assumption is not restrictive because the features may be complex and nonlinear in their inputs. For instance, this scheme is standard in deep RL models, which first derive features from video[39].

In a standard temporal difference (TD) learning model, weights are learned using the RPE $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ (ref. 40). A typical cartoon of how these are mapped onto brain circuitry is shown in Fig. 1a. A cortical input population vector for state features projects to value and RPE stages, corresponding, respectively (because both variables are scalar), to presumed uniform populations of striatal and DA neurons[1,2]. The RPE drives weight learning at corticostriatal synapses via ascending DA projections. Previous models that account for DA heterogeneity often replace the target r_t with additional target outcomes, replicating this scalar prediction model across parallel circuits ('outcome-specific PE models'; Fig. 1b)[27,31,41–46].

In contrast, we propose a new 'feature-specific RPE model'. This model relaxes the unrealistic anatomical assumption of complete, uniform convergence from vector state to scalar at the corticostriatal stage of this circuit (Fig. 1c). In fact, projections are topographic at each level of this circuit[32,34,47]. If different striatal units i preferentially receive input from particular cortical features $\phi_{i,t}$, then value will itself be represented by a distributed feature code[2] $V_{i,t} = w_i \phi_{i,t}$. In turn, DA neurons preferentially driven by each 'channel' will compute feature-specific RPEs

$$\delta_{i,t} = \frac{r_t}{N} + \gamma V_{i,t+1} - V_{i,t} = \frac{r_t}{N} + w_i(\gamma \phi_{i,t+1} - \phi_{i,t}), \quad (1)$$

where N is the number of channels. Due to linearity, the aggregate response (summed over channels i) at both the value and RPE stages reflects the original scalar variable. Thus, assuming that the ascending DAergic projection is sufficiently diffuse in order to mix the channels before the weight update, the model corresponds algebraically to the classic model but with a more realistic anatomical mapping.

In short, nonuniform convergence from state to value layers, and from there to RPE, implies value and RPE stages that reflect input feature variation but preserve the scalar RPE computational (Fig. 1a) due to linearity. This insight holds under more realistic models in which channels partially mix at each step due to limited convergence (for example, if a feature at any stage is the average of several input features).

Although still stylized, this account represents a more realistic picture than the traditional scalar model. It accommodates the known topography in the projections from cortical inputs to medium spiny neurons (MSNs) and from MSNs to DA units[32,33,38], as well as physiological findings that both input populations show heterogeneous tuning to similar features as do the DA neurons[26,48–51].

Even before specifying a feature basis, these properties imply several aspects of the heterogeneous DA response. Consider times when primary reward is not present, such as during the cue period in Engelhard et al.[9]. Here, $r_t = 0$, and equation (1) reduces to $\delta_{i,t} = w_i(\gamma \phi_{i,t+1} - \phi_{i,t})$; each DA unit reports the time-differenced activity in its feature weighted by w_i, its own association with value. Depending on what the features $\vec{\phi}_t$ are, this would explain DAergic response correlations with different, arbitrary covariates; if some feature ϕ_i is task relevant, it will have nonzero w_i (with a sign determined by ϕ_i's partial correlation with value given the other features), and its derivative will correlate with a subset of neurons.

Conversely, at outcome time, most modeled RPE units should respond differentially for reward versus nonreward (due to r_t being shared across channels in equation (1)), as in Engelhard et al.[9]. This reward response may also be modulated by its predictability due to each channel's share of the temporal difference, $w_i(\gamma \phi_{i,t+1} - \phi_{i,t})$, but is unlikely to be perfectly canceled by most individual features. Finally, because the standard RPE δ_t is equal to the sum over all channels $\Sigma_i \delta_{i,t}$, the model explains why neuron-averaged data (or bulk signals as in photometry or blood oxygen level-dependent signal), as often reported, resemble TD predictions even potentially in the presence of interneuron variation.

Deep RL network to simulate the feature-specific PE model

To simulate the feature-specific PE model in a specific task, we must specify basis functions for the task state. We consider our previously reported experiment in which mice performed an evidence accumulation task in VR while VTA DA neurons were imaged[9] (Fig. 2a). Mice navigated in a virtual T-maze while viewing towers that appeared transiently to the left and right and were rewarded for turning to the side with more towers.

To simulate the vector of feature-specific PEs (here, RPEs), we trained a deep RL agent on the same task that the mice performed in VR to derive a vector of features and, in turn, a corresponding distributed code for values and RPEs (Fig. 2b). We used a neural network to map the visual images from the VR task to 64 feature units (via convolutional layers for vision and then recurrent units for evidence accumulation; see Methods). These features were used as input for linear value prediction (as described above, producing the feature-specific RPE that sums to the traditional scalar RPE) and action selection (left, right or forward) at each step. We trained the network to perform the task using the A2C algorithm[39]. Note that neither A2C nor our model uses the feature-specific RPEs individually for training (only their sum).

After training, the agent accumulated evidence along the maze to choose the correct side with accuracy similar to mice (Fig. 2c; this and later results are consistent in an independent training run; Extended Data Fig. 1). A minimal abstract state space underlying the task is two dimensional, consisting of the position and the number of towers seen (left minus right) so far. State features from the network are tuned to different combinations of these features, implying that they span a relevant state representation for the task (Extended Data Fig. 2). Further, the scalar value function output by the trained agent is modulated by trial difficulty, meaning that the trained agent can predict the likelihood of reward (Fig. 2d).

https://doi.org/10.1038/s41593-024-01689-1

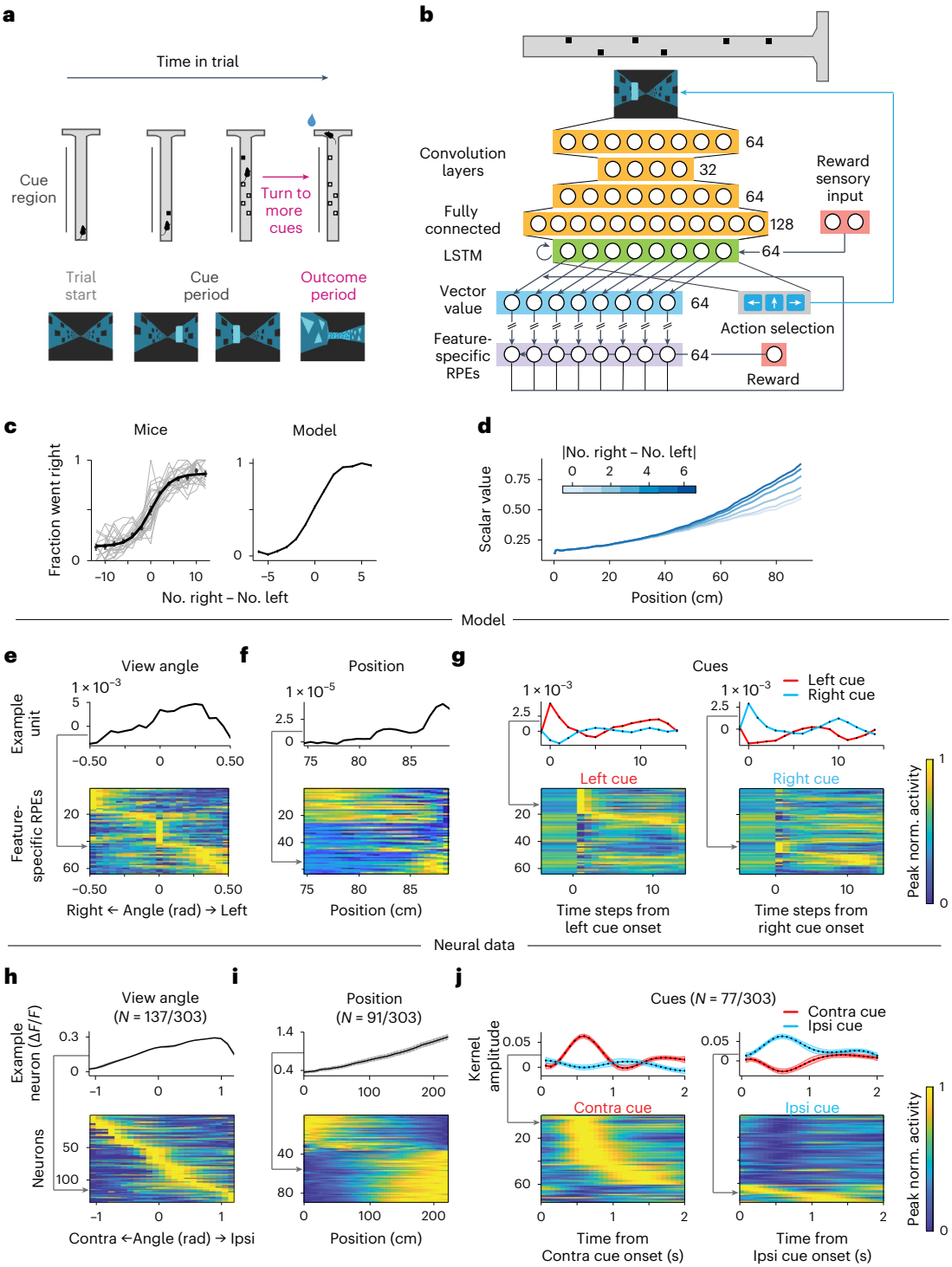


Fig. 2 | Feature-specific RPEs derived from a deep RL model trained on a VR evidence accumulation task are heterogeneously modulated by behavioral variables during the cue period, similar to DA neurons. a, Schematic of the VR task. Mice accumulated visual evidence (cues) as they ran down the stem of a T-maze and were rewarded if they turned to the side with more cues. **b**, A deep RL network was trained to play the game by transforming video frame input to value predictions and an action choice. The green layer (long short-term memory ('LSTM')) served as a feature set for per-feature value predictions and feature-specific RPEs. **c**, Psychometric curves showing the performance of the mice and model after training. The fraction of right choices is plotted as a function of the difference of the right and left towers presented on the trial. For the mice, the gray lines denote the average psychometric curves for individual sessions, and the black line denotes a logistic fit to the grand mean. Error bars indicate s.e.m. (mice, $N = 23$ sessions; model, $N = 5{,}000$ trials). **d**, The deep RL model's scalar value (sum over units in vector value) during the cue region decreases with trial difficulty. No., number. **e**, Feature-specific RPE unit activity with respect to the view angle of the agent (minimum–maximum normalization; top, example unit; bottom, all units). **f**, Same as **e** but for position of the agent in the final 15 cm of the maze. **g**, Same as **e** and **f** but for left (red) and right (blue) cues; norm., normalized. **h**–**j**, Same as **e**–**g** but for the subset of neurons from Engelhard et al.[9] tuned to view angle of the mice (**h**), position (**i**) and contralateral (red) and ipsilateral (blue) cues (**j**). Fringes represent ±1 s.e.m.; Contra, contralateral; Ipsi, ipsilateral. In **h** and **i**, peak-normalized $\Delta F/F$ signals are plotted, whereas in **j**, cue kernels are plotted from an encoding model.

https://doi.org/10.1038/s41593-024-01689-1

Heterogeneous cue period responses in feature-specific RPEs

In our prior imaging of DA neurons, our key finding during the cue period was heterogenous coding of variables such as view angle, position and cue side[9]. This heterogeneity was followed by relatively homogeneous responses to reward during the outcome period.

We first sought to determine if the simulated feature-specific RPEs had similar heterogenous tuning. We first considered the view angle during the central stem of the maze. Feature-specific RPEs displayed idiosyncratic selectivity across units for the range of view angles (Fig. 2e), which qualitatively resembled our previous results from DA neuron recordings (Fig. 2h). A subset of RPE units showed position selectivity, including both downward and upward ramps toward the end of the maze (Fig. 2f), again qualitatively resembling our neural recordings (Fig. 2i). Finally, as in the neural data (Fig. 2j), feature-specific units had idiosyncratic and heterogeneous cue selectivity, including preference for right versus left cues and diversity in response timing (Fig. 2g). Similar to the neural data[9], we observed clustering of these feature responses (Extended Data Fig. 3a), with different clusters responding most strongly to cues, position or choice (Extended Data Fig. 3b).

Reward-irrelevant responses in feature-specific RPEs

In neural data, DA units correlate with task features that appear to be reward irrelevant[9]. In the model, the network's need to extract relevant task state from high-dimensional video input implies the possibility that reward-irrelevant aspects of the input may 'leak' into the state features and then into the feature-specific RPEs, even if they average out in the scalar RPE. Although we do not intend backpropagation as a mechanistic account of how the brain learns features, it illustrates how a scalar objective (predicting scalar reward) imposes few constraints on upstream feature representations.

We thus sought to investigate the coding of reward-irrelevant visual information in the feature-specific RPEs of the agent. We focused on unambiguously reward-irrelevant visual structure in the task, specifically, the incidental background patterns on the wall in the maze stem (Fig. 3a). This pattern repeats every 43 cm. This period is visible as banding in the matrix of similarity between pairs of video frames and as peaks in the autocorrelation-like function showing the average similarity as a function of the distance between frames (Fig. 3b). To investigate whether these irrelevant features are present at the level of the feature-specific RPEs, we repeated the same analysis on them. To ensure that the network inputs reflect the structure, for this analysis, we exposed it to a maze traversal with a fixed view angle. The feature-specific RPEs show the same pattern of enhanced similarity at the 43-cm period, reflecting the irrelevant features (Fig. 3c,d). This effect remains a prediction for future neural experiments because, in the mouse experiment, the patterns cycled too quickly (more than once per second at typical running speed) for any correlations with DA responses to be resolvable. Note that this pattern is not present in the scalar RPE, indicating that training eliminates task-irrelevant features from the aggregated, but not the individual, RPEs (Extended Data Fig. 4).

Cue responses are consistent with feature-specific RPEs

Although our model implies idiosyncratic and task-irrelevant tuning in individual DA neurons, it also makes a fundamental prediction about the nature of these responses. In general, we expect that units that appear to respond to some feature (such as contralateral cues) do not simply reflect sensory responses (the presence of the cue) but rather should be further modulated by the component of RPE elicited by the feature. This could be particularly evident in the response averaged over units selective for a feature.

We thus performed a new analysis and subdivided cue-related responses (which are largely side selective in both the model and neural data; Fig. 2g,j) to determine whether they were additionally sensitive to the RPE associated with a cue on the preferred side.

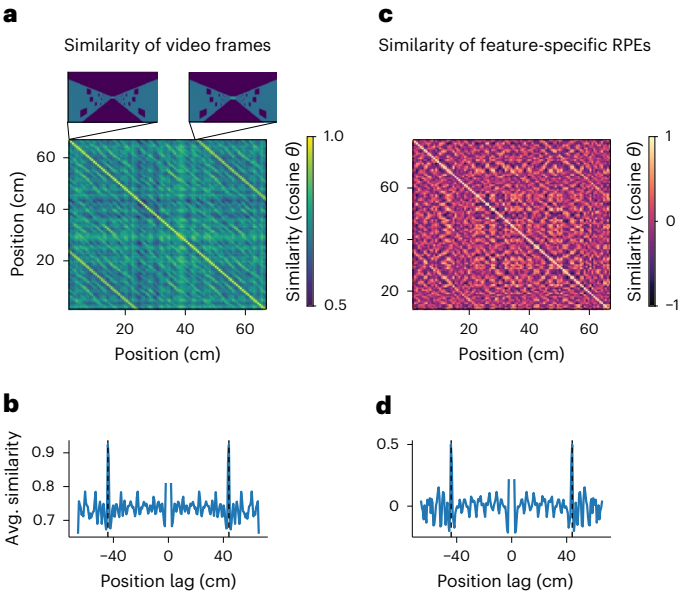


Fig. 3 | Feature-specific RPEs reflect incidental high-dimensional visual inputs. a, Similarity matrices of the video frames, which measure the similarity between pairs of video frames (quantified by the cosine of the angle between them when flattened to vectors) across different position combinations. The off-diagonal bands correspond to the wall pattern repetitions (see video frames for positions 0 cm and 43 cm at the insets above). **b**, Average similarity as a function of distance between frames, indicating that the average similarity peaked at the same position lag (43 cm) for video frames; Avg., average. The data in **c** and **d** are the same as in **a** and **b** but with feature-specific RPEs.

For this, we distinguished confirmatory cues, those which appear when their side has already had more cues than the other and therefore (due to the monotonic psychometric curve; Fig. 2c) imply an increase in the probability that the final choice will be correct and rewarded (that is, positive RPE), from disconfirmatory cues, whose side has had fewer towers so far and therefore imply decreased probability of reward (Fig. 4a). As expected, responses from the population of cue onset-responding feature-specific RPE units were stronger, on average, for confirmatory cues than for disconfirmatory cues (Fig. 4b), reflecting the component of RPE associated with the cue. We next reanalyzed our previous DA recordings based on this same insight. Consistent with a contralateral cue-specific RPE, the responses of cue-selective DA neurons were stronger for confirmatory contralateral cues than for disconfirmatory contralateral cues (Fig. 4c). Importantly, the fact that these cue-responsive neurons are overwhelmingly selective for contralateral cues implies that these responses, combined across hemispheres, simultaneously represent two separate components of a vector of feature-specific RPEs (as opposed to a classic scalar RPE). We confirmed this by further separating the neural response between neurons recorded from each hemisphere (Extended Data Fig. 5a,b). As in the model (Extended Data Fig. 5c,d), responses to cues contralateral to each side both distinguish confirmatory from disconfirmatory cues.

Uniform responses to reward at outcome period

In addition to explaining heterogeneity during the cue period, the feature-specific RPE model also explained the homogeneity of the neural responses to reward during the outcome period. Reflecting the standard properties of an RPE, the simulated scalar RPE (averaged over units) responded more for rewarded trials than for unrewarded trials (Fig. 5a). Because this aspect of the response arises (equation (1)) from a scalar reward input, it is consistent across units (Fig. 5b), matching the neural data from our experiment (Fig. 5c) and the widely reported reward sensitivity of DAergic units.

https://doi.org/10.1038/s41593-024-01689-1

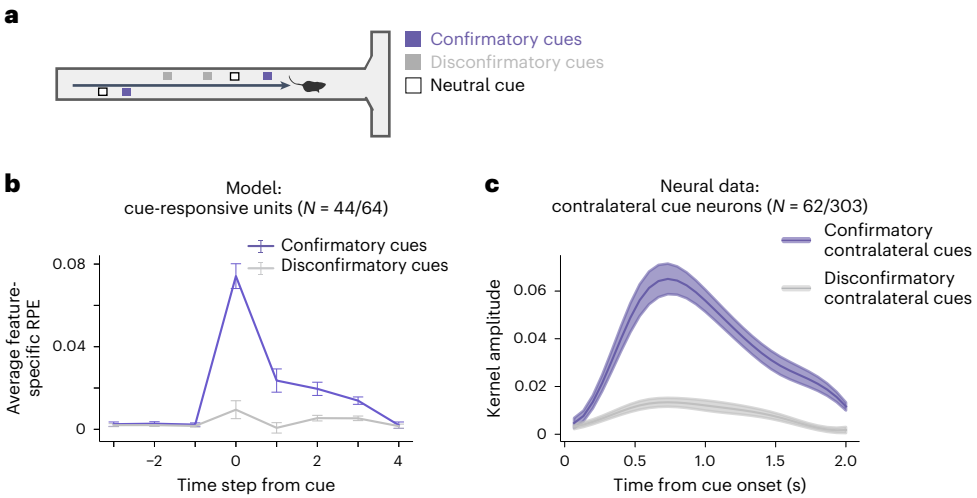


Fig. 4 | Cue responses in model and DAergic neurons reflect RPEs with respect to cues rather than simply their presence. a, Example trial illustrating confirmatory cues (purple), defined as cues that appear on the side with more evidence shown thus far, and disconfirmatory cues (gray), which are cues appearing on the side with less evidence shown thus far. Neutral cues (white) occur when there has been the same amount of evidence shown on both sides. **b**, Average response of feature-specific RPE units modulated by cues ($N = 44/64$) to confirmatory (purple) and disconfirmatory cues (gray). Error bars indicate ±1 s.e.m. **c**, Average responses of contralateral cue-responsive DA neurons for confirmatory (purple) and disconfirmatory (gray) contralateral cues. Colored fringes represent ±1 s.e.m. for kernel amplitudes ($N = 62$ neurons, subset of cue-responsive neurons from Fig. 2j that were modulated by contralateral cues only).

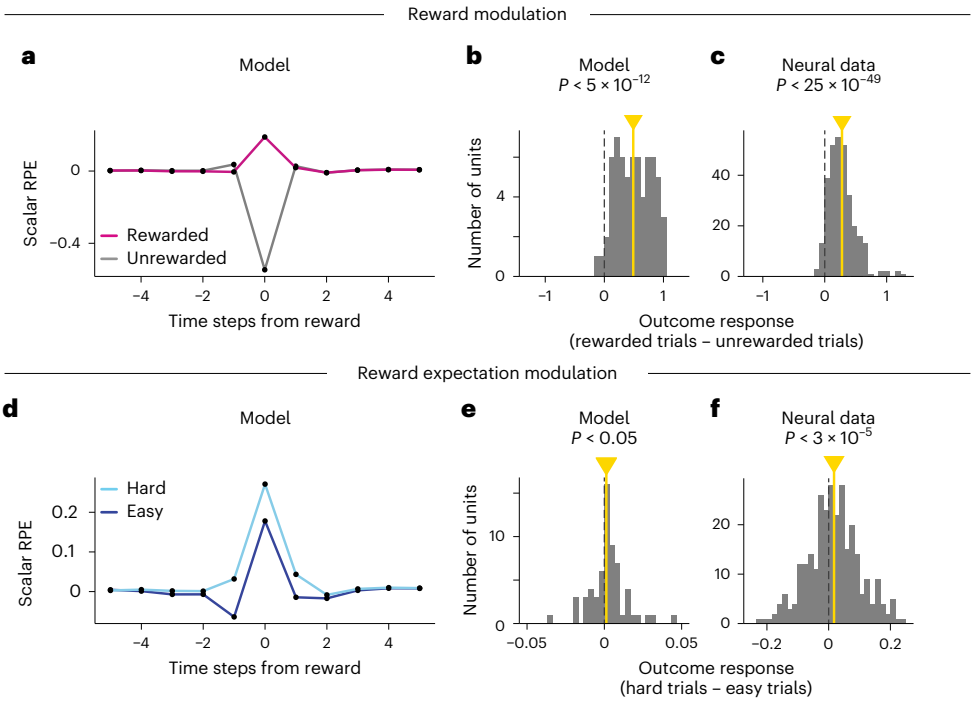


Fig. 5 | Feature-specific RPE units and DAergic neurons consistently respond to reward but show heterogeneous modulation by reward expectation. a, Scalar RPE (sum of feature-specific RPE units) time locked at reward time for rewarded (magenta) and unrewarded trials (gray). **b**, Histogram of feature-specific RPE units' responses to reward minus omission at reward time ($P = 4.47 \times 10^{-12}$ for a two-sided Wilcoxon signed rank test; $N = 64$). The yellow line indicates the median. **c**, Same as **b** but with all imaged DA neurons[9], using averaged activity for the first 2 s after reward delivery baseline corrected by subtracting the average activity 1 s before reward delivery ($P = 1.43 \times 10^{-49}$ for a two-sided Wilcoxon signed rank test; $N = 303$). **d**, Scalar RPE time locked at reward time for rewarded trials split by hard (light blue) and easy (dark blue) trials, defined as trials in the bottom or top tercile of trial difficulty and measured by the absolute value of the difference between towers presented on either side. **e**, Histogram of feature-specific RPE units' responses to difficult minus easy rewarded trials at reward time ($P = 0.033$ for a two-sided Wilcoxon signed rank test; $N = 64$), with the median indicated by the yellow line. There are two outlier data points at 0.12 and 0.29 for a feature-specific RPE unit, showing strong reward expectation modulation. **f**, Same as **e** but with the averaged activity of DA neurons[9] corrected by their baseline activity 1 s before reward delivery ($P = 2.95 \times 10^{-5}$ for a two-sided Wilcoxon signed rank test; $N = 303$).

Equation (1) also implies a subtler prediction about the feature-specific RPEs, which is that the modulation of this outcome response by the reward's predictability should be more variable. This is because it arises from the feature-specific value terms in equation (1). In this task, reward expectation can be operationalized using the absolute difference in tower counts (trial difficulty; Fig. 2c); that is, reward is

https://doi.org/10.1038/s41593-024-01689-1

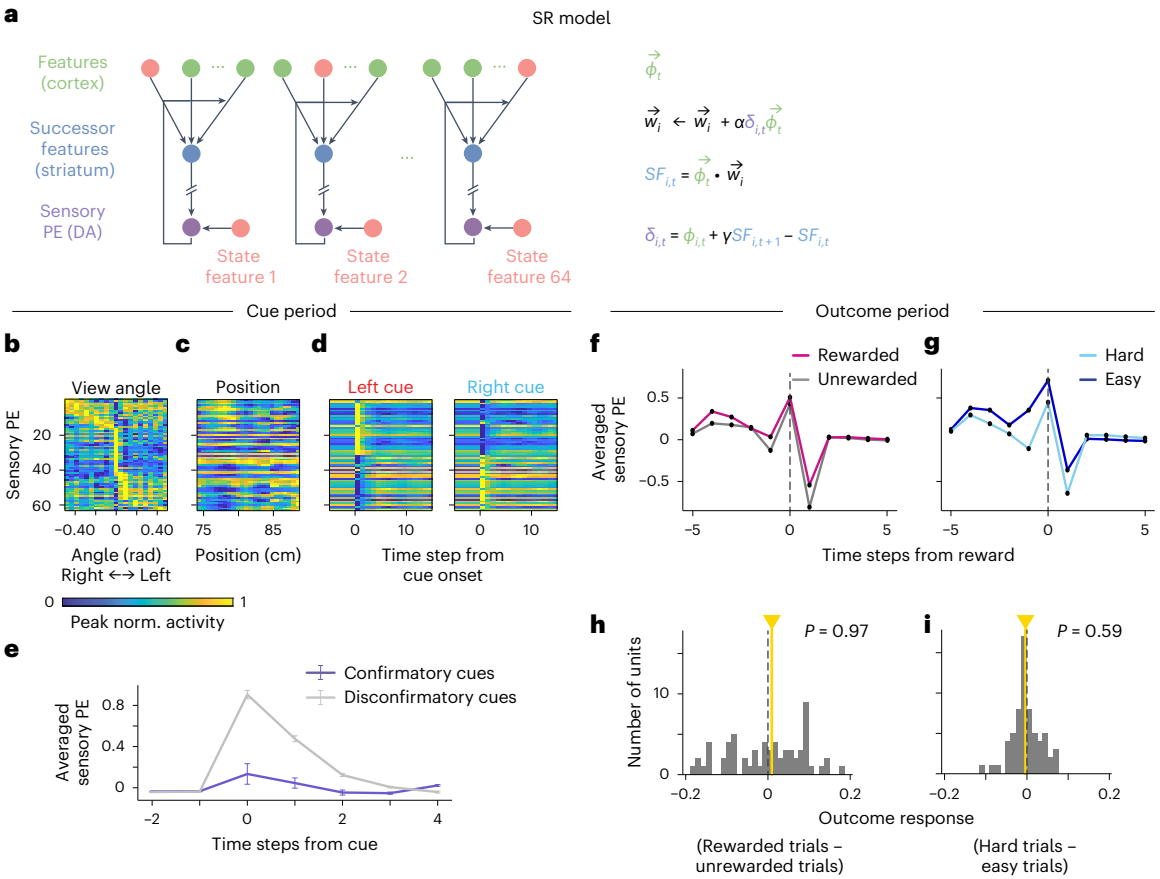


Fig. 6 | Sensory PEs from the SR model are heterogeneous during the cue period but are not responsive to confirmatory cues and reward. a, Schematic of the SR model to produce sensory PEs. The model was trained on the same state features as the feature-specific RPE model (green), with a vector of successor features (blue) and vector of sensory PEs (purple) with respect to each of the 64 features. **b**, Sensory PEs plotted with respect to view angle. Each row corresponds to a unit's peak-normalized response to view angle averaged across trials. **c**,**d**, Same as in **b** but for position (**c**) and left (red) and right (blue) cues (**d**). **e**, Sensory PE averaged across units in response to confirmatory (purple) and disconfirmatory (gray) cues shows a stronger response for disconfirmatory cues. Error bars indicate ±1 s.e.m. ($N = 64$). **f**, Sensory PE averaged across units split by rewarded (magenta) and unrewarded (gray) trials (at the time of reward). **g**, Same as **f** but for easy trials (dark blue) and hard trials (light blue), defined as in Fig. 5d. **h**, Histogram of the difference in sensory PE units' responses at reward time for rewarded minus unrewarded trials, with the yellow line indicating the median ($P = 0.97$ for a two-sided Wilcoxon signed rank test; $N = 64$). **i**, Same as **h** but for easy versus hard trials. Only rewarded trials are included ($P = 0.59$ for a two-sided Wilcoxon signed rank test; $N = 64$).

more unexpected following a more difficult discrimination (Fig. 2d). Accordingly, the simulated scalar RPE was larger for rewards on hard trials than on easy trials (Fig. 5d and Extended Data Fig. 6). However, when broken down over individual units, the effect varied widely (Fig. 5e), although the median of individual units was consistent with the scalar RPE ($P < 0.05$, two-sided Wilcoxon signed rank test). A similar finding emerged from the neural data; although, on average, the reward response was modulated by expectation, this effect varied across units (Fig. 5f).

SR cannot explain VTA DA heterogeneity

Another candidate account of VTA DA heterogeneity is the SR model (Fig. 6a)[31], which is an example of an outcome-specific PE model (Fig. 1b). This model posits that DA heterogeneity can arise from the prediction of different sensory features in parallel circuits, in addition to a circuit-predicting reward. We simulated a set of SR error signals by training an SR model to predict the features produced by the deep RL network trained on the task and computing their sensory PEs. Like the neural data and the feature-specific RPEs, the SR model produced heterogeneous responses during the cue period (Fig. 6b–d). However, these modulations were purely sensory rather than value driven and thus did not replicate the neural data. For example, responses to disconfirmatory cues were stronger than responses to confirmatory cues (Fig. 6e; opposite to the data and the feature-specific RPEs; Fig. 4b,c) because disconfirmatory cues are, on average, more surprising. Similarly, the SR model's outcome period responses are not consistently modulated by reward (Fig. 6f,h) or reward expectation (Fig. 6g,i), contrary to the hallmark feature of VTA DA neurons.

Distributional RL cannot explain VTA DA heterogeneity

Another prominent outcome-specific PE model is distributional RL (Fig. 7a)[27], which posits that parallel channels predict different expectiles of expected future reward. This model captures subtle variability in cue and outcome DA responses in classical conditioning[27], but it is unclear if it might explain broader variability in more complex tasks. We simulated expectile PEs using the features produced by the deep RL network trained on the task. Because distributional RL uses a common reward outcome across all channels, it was able to capture reward-related aspects of the neural data, including stronger responses to confirmatory over disconfirmatory cues (Fig. 7e), uniform response to reward (Fig. 7f,h) and modulation by reward expectation (Fig. 7g,i). Distributional RL was also able to produce some heterogeneity over responses during the cue period (Fig. 7b–d). Specifically, units varied in their response to features like cues or view angle because these are associated with different outcome distributions. However, because it combines all features for each PE, the model predicts invariant,

https://doi.org/10.1038/s41593-024-01689-1

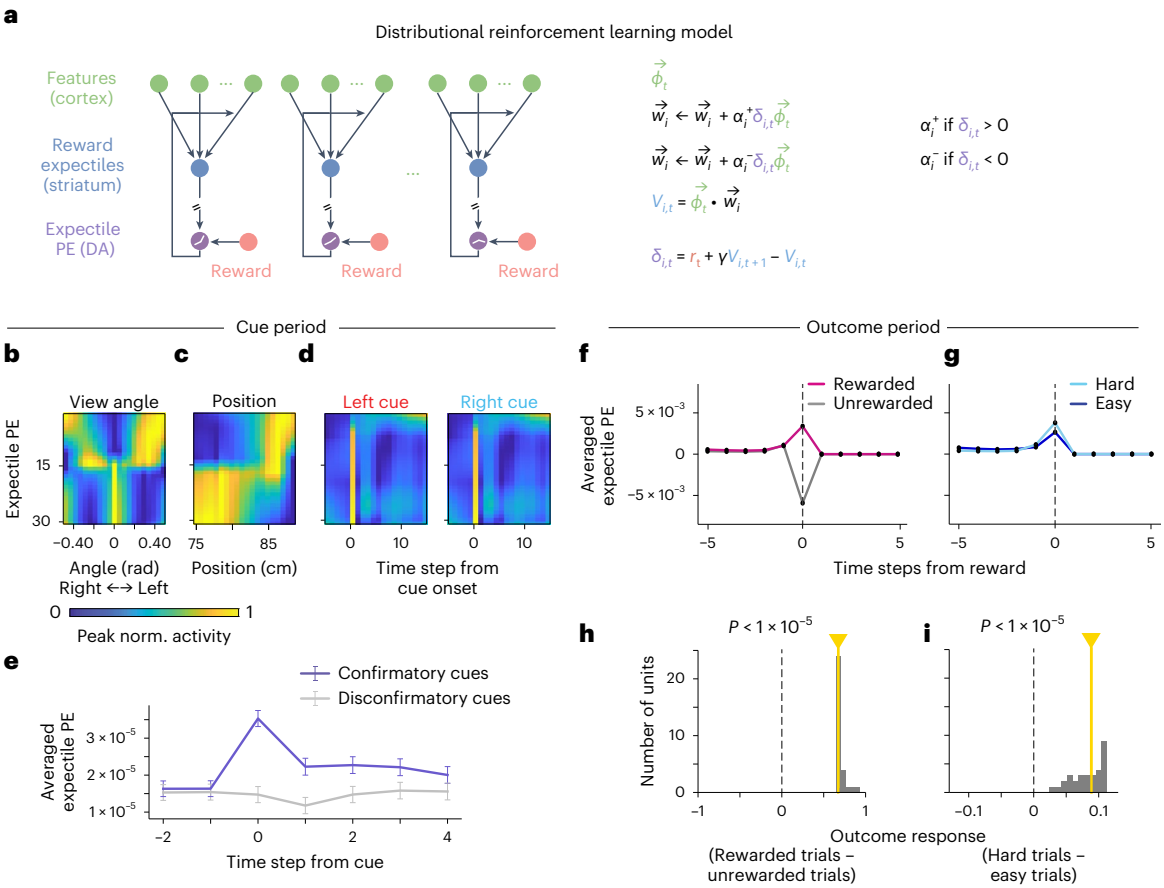


Fig. 7 | Expectile PEs from a distributional RL model do not fully capture cue period heterogeneity. a, Schematic for the distributional reinforcement learning model, which produces expectile PEs. The model uses the same state features as the feature-specific RPE model (green), with all the features input into 31 channels. These channels represent different expectiles of value (evenly spaced between 0 and 1) and 31 corresponding expectile PEs, which differentially weight positive and negative information. **b**, Expectile PE activity plotted with respect to view angle, peak normalized for each unit and sorted in the original distributional channel order. **c**,**d**, Same as in **b** but with position (**c**) and left (red) versus right (blue) cues (**d**). **e**, Expectile PEs averaged across units, responding to confirmatory (purple) and disconfirmatory (gray) cues. Error bars indicate ±1 s.e.m. ($N = 32$). **f**, Expectile PEs averaged across units time locked to reward and split by rewarded (magenta) versus unrewarded (gray) trials. **g**, Same as in **f** but split by easy trials (dark blue) and hard trials (light blue). Only rewarded trials are included, and trial difficulty is defined the same as in Fig. 5d. **h**, Histogram of expectile PEs at reward time, showing the difference in their response for rewarded and unrewarded trials ($P < 1 \times 10^{-5}$ for a two-sided Wilcoxon signed rank test; $N = 31$ units). Median response is indicated by the yellow line. **i**, Same as in **h** but for trial difficulty, taking the difference between responses for hard and easy rewarded trials ($P < 1 \times 10^{-5}$ for a two-sided Wilcoxon signed rank test; $N = 31$ units).

symmetric responses to features that are equivalent with respect to outcomes, for example, cues or view angles on either side. The prominent side selectivity of DA responses in the data (for example, Fig. 2h,j) thus suggests additional variability across neurons related to individual feature inputs over that accounted for by outcome statistics.

Feature-specific APEs explain SNc DA responses

So far, we have shown that the feature-specific PE model captures, and outcome-specific PE models fail to match, heterogeneity of DA responses within the VTA. This is consistent with our expectation that outcome-specific PEs are more relevant to heterogeneity between, rather than within, a DA subpopulation. How does each class of model explain data from other tasks and other DA populations? DA neurons in the SNc[6,7,16,52] and substantia nigra pars lateralis[28] can exhibit qualitatively different responses than those in the VTA, which may relate to their striatal projection target. Some SNc/substantia nigra pars lateralis DA neurons respond to actions, for example, displaying a transient increase in firing shortly before action onset, followed by a longer depression throughout execution[16,26,52–54]. In general, action selectivity in DA responses in the SNc tends to be accompanied by reduced response to reward[6,16,28,52]. For example, Parker et al.[6] recorded DA terminals in the dorsomedial striatum (DMS) (likely arising from the SNc) during a lever choice task (Fig. 8a). They responded robustly to contralateral movements (for example, press of the lever contralateral to the recording site) but (in contrast to classic RPEs observed in the nucleus accumbens (NAc), a VTA target) showed little modulation to reward or reward-predicting cues.

Can outcome-specific PE or feature-specific RPE models explain the apparent tradeoff, across regions, between movement and reward selectivity? One potential explanation is the APE model[28–30], an outcome-specific PE model comprised of parallel circuits that each predict different movements (for example, left versus right lever press) instead of reward (Fig. 8b and Methods).

We compared this model to the feature-specific RPE (Fig. 8c) in simulations of the Parker et al.[6] lever choice task, using a hand-designed feature space (Extended Data Fig. 7a and Methods). We simulated DA responses from both models, specifically focusing on PE neurons from the left and right subcircuits in the APE model and RPE neurons associated with the left and right choice features in the feature-specific RPE model (Fig. 8d). At choice, these behave similarly; the units fire more for their preferred action. At outcome, however, their predictions differ. Due to the r_t/N term in equation (1),

https://doi.org/10.1038/s41593-024-01689-1

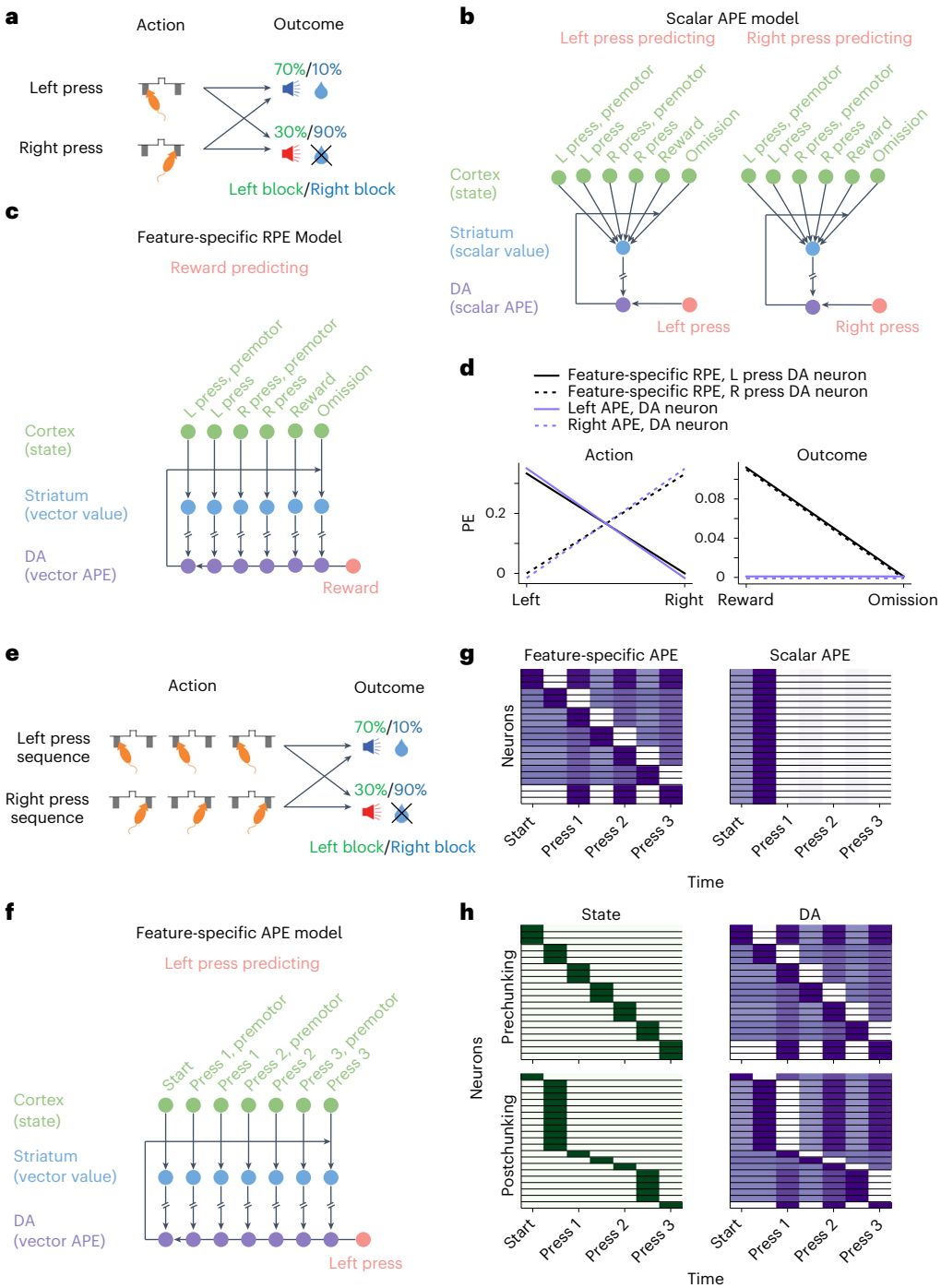


Fig. 8 | Feature-specific APEs provide a potential explanation for movement-related heterogeneity in SNc DA neurons. a, Schematic for the Parker et al.[6] reversal task. Mice chose between two levers. During a left block (green), the left lever rewarded with a probability of 0.7, and the right lever rewarded with a probability of 0.1. During a right block (blue), this contingency was reversed. Identity of the current block (left or right) was unsignaled to the mice and swapped with a pseudorandom schedule. **b**, Schematic of the scalar APE model. The model was trained with features encoding a one-hot representation of the state (green), scalar action value (blue) and scalar APE (purple). The left schematic displays the left action-preferring subcircuit, and the right schematic displays the right-preferring subcircuit; L, left; R, right. **c**, Schematic of the feature-specific RPE. Input features are the same as the scalar APE model in **b**. The value layer (blue) and the RPE layer (purple) were feature specific. **d**, DA predictions from the scalar APE model (purple) and the feature-specific RPE model (black) during action execution (left) and outcome (right) in the reversal learning task[6]. **e**, Schematic for a simplified version of the Jin and Costa task[26]. Mice chose between two levers, each of which provided a probabilistic reward after being pressed three times. Reward contingencies are the same as in **a**. **f**, Schematic for the left subcircuit of the feature-specific APE model used on the Jin and Costa[26] task. **g**, Activity of the DA neurons in the feature-specific (left) and scalar (right) APE models during the task. Neurons are sorted by peak onset. Black horizontal lines separate the neuronal activity profiles. **h**, Activity of the state (green) and DA neurons (purple) in the feature-specific APE model before (top) and after (bottom) chunking. Black horizontal lines separate the neuronal activity profiles.

feature-specific RPE neurons respond to reward, whereas APE neurons do not. Thus, although the feature-specific RPE model can produce units tuned to different movements, it does not explain the tradeoff between action selectivity and reward selectivity. The decoupling between reward and action selectivity suggests that for understanding why SNc–DMS responses differ from VTA–NAc responses, the principles of outcome-specific PE apply.

However, even within the movement-selective population in the SNc, there is additional heterogeneity. For example, Jin and Costa[26] recorded individual DA neurons during a task in which mice had to press a lever multiple times for reward. Most DA neurons had specific preferred presses (for example, the first or last presses within a sequence). Differential press-specific tuning is difficult to explain with a classic APE model. Because later presses are predicted by earlier presses, tuning for a standard APE unit should uniformly transfer to the start of the sequence the same way it classically does in the scalar RPE model. However, press-specific heterogeneity could reflect feature-specific computation within each APE circuit (Fig. 8f).

To assess whether a feature-specific APE model could account for these results, we augmented the choice task from Fig. 8a with a requirement of three presses of the lever to receive the reward (Methods and Fig. 8e). To create predictions for DA responses on this task, we compared the standard APE model to a feature-specific APE model (Fig. 8f) that functions like the feature-specific PE model but uses the state features to produce a set of PEs for (for example) left presses as outcomes instead of rewards (Methods). For simplicity, we only analyze the left-selective circuits within each of these, but both include parallel left- and right-selective circuits.

For both models, the state representation encoded the press count (Extended Data Fig. 7b). In the standard APE model, the resulting response peaked at the first press for all neurons (Fig. 8g, right). By contrast, the feature-specific APE model showed uniform selectivity for press counts across all DA neurons (Fig. 8g, left), with each neuron reflecting the tuning of its upstream state feature. Thus, the feature-specific APE recapitulates the experimentally reported press selectivity[26]. Furthermore, each feature-specific DA neuron exhibited a transient decrease in activity after its preferred press due to the feature-difference term in equation (1). This property has been observed in SNc DA neurons[16,26,52–54].

One prediction of our model is that DA heterogeneity should directly reflect the upstream corticostriatal state features. Jin and Costa[26] observed that from day 1 to day 12 of training, the proportion of first- and last-press neurons increased in tandem for both SNc DA neurons and striatal MSNs. If the population of DA responses is coupled to the upstream code, our model predicts that these changes imply a remodeling of the upstream population (simulated in Fig. 8h by changing the state representation in our model so that the first and last presses are overrepresented; as expected, this induces the corresponding representational shift at the DA level). This is consistent with evidence that overtraining is accompanied by a 'chunking' of corticostriatal representations of the action sequence, emphasizing its starting and concluding actions[55].

Discussion

We propose a new theory that reconciles recent empirical reports of DA response heterogeneity to the classic idea of DA neurons as encoding RPEs. Most previous work has considered 'outcome-specific' PEs that describe heterogeneity between parallel circuits for predicting different outcomes. We introduce a complementary 'feature-specific' model, where additional heterogeneity arises from the high-dimensional state input to each such circuit. This model, but not alternative outcome-specific PE models, produces heterogeneous responses to task variables but relatively uniform responses to reward, recapitulating empirical results[9,15]. We also test the model's prediction that heterogeneous DA responses are not simply responses to sensory and behavioral features of the task but instead reflect components of the RPE with respect to the features. Finally, we show how outcome- and feature-specific PEs combine to explain different aspects of movement-related heterogeneity in SNc DA neurons.

Feature- versus outcome-specific PEs: physiology

Our central observation is that TD models imply two sources of heterogeneity across DA units, specifically heterogeneity arising from the target outcome (for example, reward) and that arising from the state input (for example, cues). Most previous theories have explained DAergic heterogeneity by positing multiple distinct error signals, typically specialized for predicting a different outcome[27,31,41–43,45,46,56,57]. We suggest that within any such circuit, nonuniform input projections from an arbitrary population code for state can give rise to diverse responses to different task events. Collectively, these responses constitute a population code over feature-specific PEs for otherwise standard TD learning to predict (in each circuit) each scalar outcome like reward.

Both sorts of heterogeneity likely coexist, capturing different aspects of DA heterogeneity. We can differentiate their contributions in several ways. First, they imply different relationships between heterogeneous responses at outcome (to rewards, omissions or punishments[10,12,16,19,25,27,58–63]) versus heterogeneous responses to other task events, including stimuli and movements[6,7,9,15,16,18,22,26,52,62–65]. Outcome-specific PEs imply that variation between neurons in responses to stimuli ultimately arises from those neurons specializing in different outcomes. Thus, these models fail to capture the coexistence in VTA DA between heterogeneous responses to task variables alongside a more uniform effect of reward[9]. This is a hallmark of a feature-specific RPE, in which different neurons represent different stimulus-specific components of a PE for a common reward. Our simulations show that different outcome-specific models can either produce heterogenous coding of cues and task variables by predicting many outcomes but then fail to respond to reward (Fig. 6, SR) or instead maintain the overall reward effect by focusing on predicting a narrower range of outcomes but with limited variability in cue responses (Fig. 7, distributional RL).

A related prediction of the feature-specific RPE is that DA encoding of task features reflects components of RPE rather than strictly reporting each feature itself. For instance, in our data and in the feature-specific RPE, DA neurons are tuned for different cues (left versus right), but each population further distinguishes confirmatory from disconfirmatory cues (Fig. 4b,c), which differ in their reward consequences. By contrast, different SR neurons respond to the cues themselves (Fig. 6d,e). In the data, these responses are tied to individual features, consistent with the feature-specific model's unique claim that different neurons represent RPE components for left and right cues separately. This stands in contrast to a distributional RL model (Fig. 7a) and to classic scalar TD (Fig. 1a), both of which also predict that cue responses should be modulated by their reward associations (for example, confirmatory versus disconfirmatory; Fig. 7e) but wrongly predict that all neurons should respond similarly to left versus right cues, which do not have distinct reward associations (Fig. 7d).

Because both frameworks are so general (different PEs can be defined for any vector of outcomes and also features), we have emphasized qualitative predictions that aim to capture what is common to each family. We have also, for concreteness, used a deep network model to learn a high-dimensional feature set with which to simulate the model. However, this model involves several simplifications (for instance, it does not vary running speed), which means that it cannot capture some responses reported in the neural data[9] and differs with respect to others (for example, head direction, which is more constrained in simulation). Another aspect that is different in simulation is position-related ramps. Although these arise in simulation and demonstrate how feature-specific PEs can capture temporally extended responses, they tend to occur over shorter distances in the model than

in DA recordings. A full explanation of positional ramping in DA likely involves additional considerations (for example, uncertainty) beyond the current model[66].

Feature-specific versus outcome-specific PEs: anatomy

Feature- and outcome-specific heterogeneity should also relate to anatomy. DA neurons reporting different outcome-specific PEs should be linked to their targets in a closed loop, meaning this form of heterogeneity should be segregated across projection-defined DA populations. By contrast, heterogeneity arising from feature-specific inputs may be nested within each outcome-specific circuit and should exist even within a projection-defined DA population.

We thus predict that the characteristics of outcome- versus feature-specific PE variation should predominate, respectively, between versus within different projection-defined DA populations. Investigating this prediction (by comparing responding within and between projection-defined DA populations) is the key open empirical test of our framework. But existing data generally suggest that heterogeneity related to outcome-specific differences tends to occur between distinct DA nuclei or targets[6,67,68], in contrast to the hallmarks of feature-specific PEs we study here, which occur within the VTA[9,15] and SNc[26]. Thus, in SNc and DMS, compared to the VTA and NAc, movement responses tend to emerge, whereas reward responses decline (a combination consistent with outcome- rather than feature-specific variation; Fig. 8)[6,16,18,26,65]. Similarly, DA neurons responding to threat rather than reward have been reported in projections from the lateral SNc to the tail of the striatum[25], and the differential responsivity to RPE magnitudes as in distributional RL emerges across DA projections to distinct striatal subregions[67].

Although anatomical data remain incomplete, they are also consistent with our framework. Consistent with feature-specific variation, descending inputs to DA neurons (especially from the striatum) are topographic[32,33,37,38]. These afferent neurons reflect similar heterogeneity with respect to task features as seen at the DA level[26,48–51]. Meanwhile, individual DA neurons innervate large areas of the striatum[4], and volumetric propagation of released DA may further blur DA's effect[3,69]. This suggests that a limited number of sufficiently nonoverlapping outcome-specific PE circuits are possible.

APE

In extending our framework to SNc/DMS DA movement signals (Fig. 8), we used an APE model[28,29] in which DA neurons hypothetically represent PEs for future actions. Fusing this idea with feature-specific RPE variation can, in a unified framework, explain features of DA movement sensitivity, both within the SNc and between regions (Fig. 8).

However, evidentiary support for a DA APE in the brain is not yet definitive. Although action-evoked DA responses can decrease over time[28], as though modulated by predictability, the full signatures of TD PEs, as observed for reward responses, have not yet been tested. For instance, a TD APE should transfer to cues that elicit actions, and 'omission' of an action predicted by a cue (say, following a no-go signal) should inhibit DA.

Notwithstanding this, evidence is also emerging for other outcome-specific PEs. For instance, recent work on DA neurons projecting to the tail of the striatum identifies responses to threat or novelty (again modeled with an outcome-specific PE for these variables) that decline with learning and are associated with responses such as avoidance[25,70]. Overall, how large-scale DA circuits are organized remains a major open question. Our introduction of a distinction between outcome- and feature-based heterogeneity, nested at different spatial scales, offers a starting point for organizing that discourse.

Credit assignment and outcome- versus feature-specific PEs

Outcome-specific PE models have typically been motivated as enabling enhanced function: additional PEs compute additional quantities[27,31,41,42,44–46]. By contrast, feature-specific PEs arise from two practical issues. First, distinct closed-loop circuits are needed for each outcome PE, constraining the scale of such heterogeneity. Second, each of these circuits requires converging input information about many state features. These two issues are related; although it seems inevitable that inhomogeneous state feature input drives response variation within a projection-defined DA population, the same overlapping ascending projections that constrain the outcome-specific PEs also promote convergence across these feature-specific PEs.

Such convergence addresses the following central computational function: credit assignment over multidimensional feature spaces. A common RPE enables stimuli to compete to explain observed rewards, each learning to explain what cannot be explained by the others. Thus, if RPE heterogeneity arises from feature inputs rather than outcome targets, it can be advantageous that they mix (Fig. 1c) rather than remain separate (Fig. 1b). In Pavlovian conditioning, such competition is observable behaviorally in the blocking effect[71,72].

Credit assignment can also be addressed in separate parallel channels using an outcome-specific architecture via 'mixture of expert' modules that may each simplify the problem by focusing on different subsets of features. This can offer more individualized control to 'divide and conquer' and focus learning on context-relevant dimensions. One recent model[20] uses this architecture to suggest a different set of mechanisms addressing some aspects of the problems we discuss here. Overall, to what extent the brain relies on either credit assignment mechanism remains a question, but the present framework clarifies how these issues could be examined via behavioral, projection-specific and neurophysiological studies. For instance, to whatever extent different DA subpopulations represent RPEs specific to different features, but without anatomically overlapping projections, their respective cues should fail to block one another behaviorally.

Population codes for state

Although much RL modeling in neuroscience assumes a hand-constructed state representation[1,2], in general, the brain must learn or construct appropriate state features while solving a task. How it does this is a major open question. There have been several recent theoretical hypotheses addressing how the brain might build states[73–76], but there exist relatively few experimental results to assess these ideas. For instance, behavior alone is relatively uninformative about state, whereas in the brain, it is unclear which neural representations directly play this role. The feature-specific PE model implies that the heterogeneous DAergic population itself gives a new experimental window, from the RL system's perspective, into the brain's population code over state features. Although the framework is agnostic to the feature set, the various DA responses should reflect it. This should enable new experiments and analyses to infer the brain's state representation from DA recordings and to test ideas about how it changes across tasks and as tasks are acquired.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41593-024-01689-1.

References

1. Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. *Science* **275**, 1593–1599 (1997).
2. Montague, P. R., Dayan, P. & Sejnowski, T. J. A framework for mesencephalic dopamine systems based on predictive Hebbian learning. *J. Neurosci.* **16**, 1936–1947 (1996).
3. Arbuthnott, G. W. & Wickens, J. Space, time and dopamine. *Trends Neurosci.* **30**, 62–69 (2007).

4. Matsuda, W. et al. Single nigrostriatal dopaminergic neurons form widely spread and highly dense axonal arborizations in the neostriatum. *J. Neurosci.* **29**, 444–453 (2009).
5. Schultz, W. Predictive reward signal of dopamine neurons. *J. Neurophysiol.* **80**, 1–27 (1998).
6. Parker, N. F. et al. Reward and choice encoding in terminals of midbrain dopamine neurons depends on striatal target. *Nat. Neurosci.* **19**, 845–854 (2016).
7. Lee, R. S., Mattar, M. G., Parker, N. F., Witten, I. B. & Daw, N. D. Reward prediction error does not explain movement selectivity in DMS-projecting dopamine neurons. *eLife* **8**, e42992 (2019).
8. Choi, J. Y. et al. A comparison of dopaminergic and cholinergic populations reveals unique contributions of VTA dopamine neurons to short-term memory. *Cell Rep.* **33**, 108492 (2020).
9. Engelhard, B. et al. Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons. *Nature* **570**, 509–513 (2019).
10. Lerner, T. N. et al. Intact-brain analyses reveal distinct information carried by SNc dopamine subcircuits. *Cell* **162**, 635–647 (2015).
11. Collins, A. L. & Saunders, B. T. Heterogeneity in striatal dopamine circuits: form and function in dynamic reward seeking. *J. Neurosci. Res.* **98**, 1046–1069 (2020).
12. Verharen, J. P. H., Zhu, Y. & Lammel, S. Aversion hot spots in the dopamine system. *Curr. Opin. Neurobiol.* **64**, 46–52 (2020).
13. Hassan, A. & Benarroch, E. E. Heterogeneity of the midbrain dopamine system. *Neurology* **85**, 1795–1805 (2015).
14. Marinelli, M. & McCutcheon, J. E. Heterogeneity of dopamine neuron activity across traits and states. *Neuroscience* **282**, 176–197 (2014).
15. Kremer, Y., Flakowski, J., Rohner, C. & Lüscher, C. Context-dependent multiplexing by individual VTA dopamine neurons. *J. Neurosci.* **40**, 7489–7509 (2020).
16. Howe, M. W. & Dombeck, D. A. Rapid signalling in distinct dopaminergic axons during locomotion and reward. *Nature* **535**, 505–510 (2016).
17. Anderegg, A., Poulin, J.-F. & Awatramani, R. Molecular heterogeneity of midbrain dopaminergic neurons—moving toward single cell resolution. *FEBS Lett.* **589**, 3714–3726 (2015).
18. Barter, J. W. et al. Beyond reward prediction errors: the role of dopamine in movement kinematics. *Front. Integr. Neurosci.* **9**, 39 (2015).
19. Cai, L. X. et al. Distinct signals in medial and lateral VTA dopamine neurons modulate fear extinction at different times. *eLife* **9**, e54936 (2020).
20. Hamid, A. A., Frank, M. J. & Moore, C. I. Wave-like dopamine dynamics as a mechanism for spatiotemporal credit assignment. *Cell* **184**, 2733–2749.e16 (2021).
21. Mohebi, A., Wei, W., Pelattini, L., Kim, K. & Berke, J. D. Dopamine transients follow a striatal gradient of reward time horizons. *Nat. Neurosci.* **27**, 737–746 (2024).
22. Zolin, A. et al. Context-dependent representations of movement in *Drosophila* dopaminergic reinforcement pathways. *Nat. Neurosci.* **24**, 1555–1566 (2021).
23. Eshel, N., Tian, J., Bukwich, M. & Uchida, N. Dopamine neurons share common response function for reward prediction error. *Nat. Neurosci.* **19**, 479–486 (2016).
24. Cohen, J. Y., Haesler, S., Vong, L., Lowell, B. B. & Uchida, N. Neuron-type-specific signals for reward and punishment in the ventral tegmental area. *Nature* **482**, 85–88 (2012).
25. Menegas, W., Akiti, K., Amo, R., Uchida, N. & Watabe-Uchida, M. Dopamine neurons projecting to the posterior striatum reinforce avoidance of threatening stimuli. *Nat. Neurosci.* **21**, 1421–1430 (2018).
26. Jin, X. & Costa, R. M. Start/stop signals emerge in nigrostriatal circuits during sequence learning. *Nature* **466**, 457–462 (2010).
27. Dabney, W. et al. A distributional code for value in dopamine-based reinforcement learning. *Nature* **577**, 671–675 (2020).
28. Greenstreet, F. et al. Action prediction error: a value-free dopaminergic teaching signal that drives stable learning. Preprint at *bioRxiv* https://doi.org/10.1101/2022.09.12.507572 (2024).
29. Bogacz, R. Dopamine role in learning and action inference. *eLife* **9**, e53262 (2020).
30. Lindsey, J. and Litwin-Kumar, A. Action-modulated midbrain dopamine activity arises from distributed control policies. In *Proc. 36th International Conference on Neural Information Processing Systems* (eds. Koyejo, S. et al.) 5535–5548 (2022).
31. Gardner, M. P. H., Schoenbaum, G. & Gershman, S. J. Rethinking dopamine as generalized prediction error. *Proc. Biol. Sci.* **285**, 20181645 (2018).
32. Alexander, G. E., DeLong, M. R. & Strick, P. L. Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annu. Rev. Neurosci.* **9**, 357–381 (1986).
33. Lau, B., Monteiro, T. & Paton, J. J. The many worlds hypothesis of dopamine prediction error: implications of a parallel circuit architecture in the basal ganglia. *Curr. Opin. Neurobiol.* **46**, 241–247 (2017).
34. Haber, S. N., Fudge, J. L. & McFarland, N. R. Striatonigrostriatal pathways in primates form an ascending spiral from the shell to the dorsolateral striatum. *J. Neurosci.* **20**, 2369–2382 (2000).
35. Hintiryan, H. et al. The mouse cortico–striatal projectome. *Nat. Neurosci.* **19**, 1100–1114 (2016).
36. Hunnicutt, B. J. et al. A comprehensive excitatory input map of the striatum reveals novel functional organization. *eLife* **5**, e19103 (2016).
37. Pan, W. X., Mao, T. & Dudman, J. T. Inputs to the dorsal striatum of the mouse reflect the parallel circuit architecture of the forebrain. *Front. Neuroanat.* **4**, 147 (2010).
38. Cox, J. & Witten, I. B. Striatal circuits for reward learning and decision-making. *Nat. Rev. Neurosci.* **20**, 482–494 (2019).
39. Mnih, V. et al. Asynchronous methods for deep reinforcement learning. In *Proc. 33rd International Conference on Machine Learning* (eds. Balcan, M. F. & Weinberger, K. Q.) 1928–1937 (jmlr.org, 2016).
40. Sutton, R. S. Learning to predict by the methods of temporal differences. *Mach. Learn.* **3**, 9–44 (1988).
41. Daw, N. D., Kakade, S. & Dayan, P. Opponent interactions between serotonin and dopamine. *Neural Netw.* **15**, 603–616 (2002).
42. Lloyd, K. & Dayan, P. Safety out of control: dopamine and defence. *Behav. Brain Funct.* **12**, 15 (2016).
43. Lak, A., Nomoto, K., Keramati, M., Sakagami, M. & Kepecs, A. Midbrain dopamine neurons signal belief in choice accuracy during a perceptual decision. *Curr. Biol.* **27**, 821–832 (2017).
44. Daw, N. D., Courville, A. C. & Touretzky, D. S. Timing and partial observability in the dopamine system. In *Proc. 15th International Conference on Neural Information Processing Systems* (eds. Becker, S. et al.) 99–106 (MIT Press, 2003).
45. Kurth-Nelson, Z. & Redish, A. D. Temporal-difference reinforcement learning with distributed representations. *PLoS ONE* **4**, e7362 (2009).
46. Gershman, S. J., Pesaran, B. & Daw, N. D. Human reinforcement learning subdivides structured action spaces by learning effector-specific values. *J. Neurosci.* **29**, 13524–13531 (2009).
47. Voorn, P., Vanderschuren, L. J. M. J., Groenewegen, H. J., Robbins, T. W. & Pennartz, C. M. A. Putting a spin on the dorsal–ventral divide of the striatum. *Trends Neurosci.* **27**, 468–474 (2004).
48. Rueda-Orozco, P. E. & Robbe, D. The striatum multiplexes contextual and kinematic information to constrain motor habits execution. *Nat. Neurosci.* **18**, 453–460 (2015).
49. Parker, N. F. et al. Choice-selective sequences dominate in cortical relative to thalamic inputs to NAc to support reinforcement learning. *Cell Rep.* **39**, 110756 (2022).

50. Matsumoto, N., Minamimoto, T., Graybiel, A. M. & Kimura, M. Neurons in the thalamic CM–Pf complex supply striatal neurons with information about behaviorally significant sensory events. *J. Neurophysiol.* **85**, 960–976 (2001).
51. Choi, K. et al. Distributed processing for value-based choice by prelimbic circuits targeting anterior–posterior dorsal striatal subregions in male mice. *Nat. Commun.* **14**, 1920 (2023).
52. da Silva, J. A., Tecuapetla, F., Paixão, V. & Costa, R. M. Dopamine neuron activity before action initiation gates and invigorates future movements. *Nature* **554**, 244–248 (2018).
53. Dodson, P. D. et al. Representation of spontaneous movement by dopaminergic neurons is cell-type selective and disrupted in parkinsonism. *Proc. Natl Acad. Sci. USA* **113**, E2180–E2188 (2016).
54. 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).
55. Jog, M. S., Kubota, Y., Connolly, C. I., Hillegaart, V. & Graybiel, A. M. Building neural representations of habits. *Science* **286**, 1745–1749 (1999).
56. Ribas-Fernandes, J. J. F. et al. A neural signature of hierarchical reinforcement learning. *Neuron* **71**, 370–379 (2011).
57. Jiang, L. & Litwin-Kumar, A. Models of heterogeneous dopamine signaling in an insect learning and memory center. *PLoS Comput. Biol.* **17**, e1009205 (2021).
58. Matsumoto, H., Tian, J., Uchida, N. & Watabe-Uchida, M. Midbrain dopamine neurons signal aversion in a reward-context-dependent manner. *eLife* **5**, e17328 (2016).
59. de Jong, J. W. et al. A neural circuit mechanism for encoding aversive stimuli in the mesolimbic dopamine system. *Neuron* **101**, 133–151 (2019).
60. Matsumoto, M. & Hikosaka, O. Two types of dopamine neuron distinctly convey positive and negative motivational signals. *Nature* **459**, 837–841 (2009).
61. Lammel, S. et al. Input-specific control of reward and aversion in the ventral tegmental area. *Nature* **491**, 212–217 (2012).
62. Syed, E. C. J. et al. Action initiation shapes mesolimbic dopamine encoding of future rewards. *Nat. Neurosci.* **19**, 34–36 (2016).
63. O'Doherty, J. et al. Dissociable roles of ventral and dorsal striatum in instrumental conditioning. *Science* **304**, 452–454 (2004).
64. Moss, M. M., Zatka-Haas, P., Harris, K. D., Carandini, M. & Lak, A. Dopamine axons in dorsal striatum encode contralateral visual stimuli and choices. *J. Neurosci.* **41**, 7197–7205 (2021).
65. Saunders, B. T., Richard, J. M., Margolis, E. B. & Janak, P. H. Dopamine neurons create Pavlovian conditioned stimuli with circuit-defined motivational properties. *Nat. Neurosci.* **21**, 1072–1083 (2018).
66. Mikhael, J. G., Kim, H. R., Uchida, N. & Gershman, S. J. The role of state uncertainty in the dynamics of dopamine. *Curr. Biol.* **32**, 1077–1087.e9 (2022).
67. Tsutsui-Kimura, I. et al. Distinct temporal difference error signals in dopamine axons in three regions of the striatum in a decision-making task. *eLife* **9**, e62390 (2020).
68. Avvisati, R. et al. Distributional coding of associative learning in discrete populations of midbrain dopamine neurons. *Cell Rep.* **43**, 114080 (2024).
69. Gonon, F. et al. Geometry and kinetics of dopaminergic transmission in the rat striatum and in mice lacking the dopamine transporter. *Prog. Brain Res.* **125**, 291–302 (2000).
70. Akiti, K. et al. Striatal dopamine explains novelty-induced behavioral dynamics and individual variability in threat prediction. *Neuron* **110**, 3789–3804.e9 (2022).
71. Rescorla, R. A. and Wagner, A. R. A theory of Pavlovian conditioning: variations in the effectiveness of reinforcement and nonreinforcement. In *Classical Conditioning II: Current Research and Theory* (eds. Black, A. H. & Prokasy, W. F.) 64–99 (Appleton-Century-Crofts, 1972).
72. Kamin, L. J. 'Attention-like' processes in classical conditioning. *Miami Symposium on the Prediction of Behavior: Aversive Stimulation* (ed. Jones, M. R.) 9–31 (Univ. Miami Press, 1968).
73. Gershman, S. J., Norman, K. A. & Niv, Y. Discovering latent causes in reinforcement learning. *Curr. Opin. Behav. Sci.* **5**, 43–50 (2015).
74. Russek, E. M., Momennejad, I., Botvinick, M. M., Gershman, S. J. & Daw, N. D. Predictive representations can link model-based reinforcement learning to model-free mechanisms. *PLoS Comput. Biol.* **13**, e1005768 (2017).
75. Stachenfeld, K. L., Botvinick, M. M. & Gershman, S. J. The hippocampus as a predictive map. *Nat. Neurosci.* **20**, 1643–1653 (2017).
76. Niv, Y. Learning task-state representations. *Nat. Neurosci.* **22**, 1544–1553 (2019).

Methods

Behavioral task

Simulations. At every trial, the agent was placed at the start of a virtual T-maze, with cues randomly appearing on either side of the stem of the T-maze as the agent moved down the maze. On each trial, one side was randomly determined to be correct, and the number of cues on each side was then sampled from a truncated Poisson distribution, with a mean of 2.29 cues on the correct side and 0.69 cues on the incorrect side. To prevent the agent from forming a side bias, we used a debiasing algorithm to ensure that the identity of the high probability side changed if the agent kept choosing one side[77]. To match the procedure from the mouse experiment, we also oversampled easy trials (trials in which only one side had six total cues, and the other had no cues) by ensuring that they were 5% of the trials. The agent moved down the maze at a constant speed of 0.638 cm per time step and could also modulate its view angle with two discrete actions corresponding to left and right rotation. A third discrete action moved forward without changing the view angle. The cue region was 85 cm, and the cues were placed randomly along the cue region under a uniform distribution but with the restriction that cues on either side had a minimal spatial distance of 14 cm between them. Each cue first appeared when the agent was 10 cm from the cue location and disappeared once the agent passed the cue by 4 cm. After the cue region, there was a short 5-cm delay region before the agent's final left or right action determined their choice of entering either arm in the T-maze. If the agent turned to the arm on the side where more cues appeared, they received a reward. The agent was also given a sensory input in the model indicating whether it made the correct or wrong turn.

Neural data. The task simulated above was a streamlined version of that used in the mouse recordings in Engelhard et al.[9]. In particular, the rules for spacing and visual appearance of cues were the same, but the model's simulated controls were simplified (to discrete actions), and the maze was shorter (to facilitate neural network training). For the neural recordings, the mice could control their speed and direction of movement more continuously by running on a trackball and traversed a maze that had a 30-cm start region (with no cues), a 220-cm cue region and an 80-cm delay region before the T-maze arms. The mean numbers of cues were correspondingly larger: 6.4 on the correct side and 1.3 on the incorrect side. At reward time, the mice received a water reward if they made the correct choice; if a mouse made an incorrect choice, it was given a pulsing 6- to 12-kHz tone for 1 s. Before the next trial, the VR screen froze for 1 s during reward delivery and blacked out for 2 s if the mouse was rewarded or for 5 s if the mouse failed.

VR system and deep RL model

A deep RL network was trained on the evidence accumulation task. As input, the network took in 68 × 120 pixel video frames in grayscale. The model had three convolution layers to analyze the visual input, an LSTM layer to allow for memory and output to three action units and one value unit. The first convolutional layer had 64 filters, a filter size of 8 pixels and a stride of 2 pixels; the second convolutional layer had 32 filters, a filter size of 2 pixels and a stride of 1 pixel; and the third convolutional layer had 64 filters, a filter size of 3 pixels and a stride of 2 pixels. The convolution layers fed into a fully connected layer of 128 units, which fed into the LSTM layer with 64 units along with a second input, a one-hot vector of length 2, which flagged whether or not the agent was rewarded at the end of the trial. The reward input into the LSTM was meant to replicate the sensory input that the mouse experienced when it was rewarded with water or received a tone for failing the trial. The hyperparameters for the convolutional layers were optimized with a grid search of various filter numbers and sizes trained on supervised learning for recognizing towers.

We used the same MATLAB VR program (ViRMEn software engine[78]) from the original neural recordings[9], which we altered to accommodate the agent's movement choices of forward, left and right. While in the stem of the T-maze, the agent always moved forward at a constant rate per time step. The constant speed ensured that at every trial, the agent always took the same number of time steps to traverse the stem of the T-maze. The agent could choose to rotate left or right, which would alter the view angle 0.05 radians up to the limit of $-\pi/6$ and $\pi/6$ radians. The agent could also choose to move forward without changing their view angle. After the delay region in the T-maze, the agent's left and right movement would no longer alter its view angle but instead determined which arm the agent chose.

For the deep RL agent to interact with the ViRMEn software, we created a custom gym environment using OpenAI Gym[79]. Our custom VR gym environment defined the forward, left and right movements the agent could make and sent in the movement choices to the ViRMEn software, which in turn returned updated video frames.

We trained the network to maximize obtained reward using the Stable Baselines[80] (version 2.10.1, with Tensorflow 1.14.0) implementation of the Advantage Actor Critic (A2C) algorithm[39]. The loss function for training using the A2C algorithm with weight parameters θ, action at time t a_t, state s_t and reward r_t is given as

$$Loss(\theta) = \log \pi(a_t|s_t,\theta)[A(s_t,a_t|\theta)] + \beta[V(s_t|\theta) - G(s_t|\theta)] - \eta H[\pi(\cdot|s_t,\theta)].$$

The first term is the actor loss, given as the log of the policy π multiplied by a sample of the advantage $A(s_t,a_t|\theta) = G(s_t|\theta) - V(s_t|\theta)$, where $G(s_t|\theta) = \sum_{i=0}^{n_{steps}-1}[\gamma^i r_{t+i}] + \gamma^{n_{steps}} V(s_{t+n_{steps}}|\theta)$ is an n-step (defined as n_{steps} in the model) Bellman estimator for the return and V is the value (critic) function. The second term is the critic loss or the squared error of the value function estimate with respect to the return G. The final term is an entropy term, which regularized the policy to increase exploration. β and η are hyperparameters (listed below) to trade off the effects of the regularization and losses.

All hyperparameters used for the deep RL agent can be found in Supplementary Table 1. We trained the model until it reached a performance of 80% or higher correct choices, which took 140.8 million time steps or approximately 900,000 trials. To show that the results were consistent across multiple starting weights, we retrained the agent with fresh random weights and found that the new features replicated the main results of the paper (Extended Data Fig. 1).

We also ran a few additional simulations to learn more about the training process. First, pretraining the convolutional layers of the network to recognize towers and shortening the delay region from the final tower appearance to the final tower location substantially reduced training time. The agent was also sensitive to the n_{steps} hyperparameter, which determined the batch gradient size; specifically, a shorter n_{steps} early in training helped with the agent learning the task above chance, but a longer n_{steps} was needed to master the task to reach the performance we see in the paper. We also trained the network with only eight LSTM units, but the agent could not learn the task even after 200 million time steps. A larger and overparameterized network was evidently needed for the agent to understand the task and keep track of the towers seen, even though the theoretical state space sufficient for the task is three dimensional (left tower count, right tower count and position of the agent).

After training, we froze the weights and took the final output layer of the network before the action and value units (that is, the LSTM output) as the features for vector value and feature-specific RPE (equation (1) and Fig. 2b). Note that this just corresponds to decomposing the scalar value and RPE units in the original A2C network into vectors, with one value component for every LSTM-to-value weight, and a corresponding RPE component; that is, the feature-specific RPE model, being algebraically equivalent to TD, is just a more detailed view of the A2C critic. In this way, we calculated the feature-specific RPE at every point in the trial, for a total of 5,000 simulated trials. We defined the outcome period to be the five time steps before and after the reward.

We defined the cue period as the first 140 time steps of the maze, which occurred at the same positions on every trial because the agent always moved forward the same amount at each time step.

Neural data

All experimental procedures were conducted in accordance with the National Institutes of Health guidelines and were reviewed by the Princeton University Institutional Animal Care and Use Committee. This article reanalyzes data originally reported in Engelhard et al.[9], the methods of which we briefly summarize here and below[9]. We primarily reanalyzed the neural recordings during the VR experiments, in which we used male DAT::cre mice ($N = 14$; Jackson Laboratory, 006660) and male mice that were a cross between DAT::cre mice and the GCaMP6f reporter line Ai148 ($N = 6$; Ai148 × DAT::cre; Jackson Laboratory, 030328). All mice were 2–6 months old during their use in the study.

VTA DA neurons were imaged at 30 Hz using a custom-built, VR-compatible two-photon microscope equipped with a pulsed Ti:sapphire laser (Chameleon Vision, Coherent) that was tuned to 920 nm. After imaging, we removed trials in which mice were not engaged in the task, primarily those found close to the end of the session when animal performance typically decreased. Average performance across sessions on all retained trials was 77.6 ± 0.9% after removing trials (compared to 73.3 ± 1.1% including all trials). Ultimately, we used 23 sessions from 20 mice (1 session per imaging field, each session with at least 100 trials and minimal performance of 65%).

After preprocessing the imaging data, we performed motion correction procedures to eliminate spatially uniform motion and spatially nonuniform slow drifts. The $\Delta F/F$ was derived by subtracting the scaled version of the annulus fluorescence from the raw trace (correction factor of 0.58) and smoothing using a zero-phased filter with 25-point center Gaussian with 1.5-sample point standard deviation. We then divided $\Delta F/F$ by the eighth percentile of the smoothed and neuropil-corrected trace based on the preceding 60 s of recording. After examining $\Delta F/F$, we only included neurons that were stable for at least 50 trials. The full dataset we used for reanalysis has 303 neurons spread across 23 sessions from 20 mice.

Cue period responses

For the heat maps in Fig. 2e–j, each row represents a feature-specific RPE unit or neuron's response to the behavioral variable rescaled so that their maximum value is at 1 and minimum value is at 0. The example plots above show the unnormalized response. For the neural data, we used the same encoding model from our previous work[9] to predict neural activity with a linear regression based on predictors—including cues, accuracy, previous reward, position and kinematics—to isolate temporal kernels that reflect the response to each cue. The code for this encoding model can be found at https://github.com/benengx/encodingmodel. To determine which neurons are displayed for the heat maps in Fig. 2h–j, we used the same criterion as in Engelhard et al.[9]. Specifically, we included neurons with a statistically significant contribution of that behavioral variable in the full encoding model relative to a reduced model based on an F-test ($P = 0.01$), with comparison to null distributions produced by randomly shifted data to account for slow drift in the data.

Encoding model and clustering analysis

For Extended Data Fig. 3, we adapted the encoding and clustering models from Engelhard et al.[9] to estimate the relative contributions of behavioral variables to the model's simulated DA responses and then clustered them using a Gaussian mixture model. First, we used multiple linear regression to model cue period activity, per simulated neuron, as a function of three behavioral features: cues, position and the agent's choice. Each feature was coded with a group of regressors; we used the regression to measure the relative contribution (defined in terms of variance explained) of each group to each neuron's responses for clustering (below). Coding of the variables followed that used for DA neurons in Engelhard et al.[9]. Specifically, the cue variables were coded as event-locked timeseries, with the appearance of left and right cues convolved with an 8-d.f. regression spline basis across ten time steps of duration. Position was entered as a third degree polynomial series on the continuous variable (that is, we included the first, second and third powers of position). We then chose the optimal number of predictors using a fivefold cross-validation over trials. The agent's final choice was coded with three variables as one-hot vectors, indicating the decision time for a left choice, and one and two steps before that. There were no separate variables for choosing the right arm because we also included a bias variable in the regression.

For the outcome period, we modeled time steps following reward using two sets of regressors, representing (by one-hot event-triggered lagged indicators) the time of reward (on rewarded trials only) and time of outcome (comprising the time of either nonreward or reward). Each comprised seven one-hot vectors indicating the event time (reward and/or nonreward) and the six time steps following. No bias variable was included.

After regressing cue and outcome period activity, the relative contribution and Gaussian mixture model clustering procedures were performed as in Engelhard et al.[9]. Following Engelhard et al., we defined the relative contribution for each group of regressors by the reduction in variance explained when setting their betas to 0 (referred to in Engelhard et al.[9] as the 'no refitting' method) and then taking the ratio of these relative to the sum over all features.

Clustering was performed using MATLAB's 'fitgmdist' function on the matrix of neuron × feature contributions, using 1,000 maximum iterations, a regularization value of 0.30, 100 replicates and the covariance matrix constrained to diagonal. To select the number of clusters, we used the fitgmdist function to calculate the AIC score for models with varying numbers of clusters and chose the model with the lowest AIC score. The AIC score served as a measure of model fit given the number of parameters calculated by taking the difference between two times the number of parameters and two times the log likelihood of the model. After taking the cluster with the lowest AIC scores, we ordered the relative contribution by those clusters.

Wall texture analysis

In Fig. 3, we identified a repeating wall texture pattern in the maze by analyzing video frames of a maze with the view angle fixed at 0° ($N = 5{,}000$ trials). We calculated the similarity matrix for the video frames; specifically, given the video frames, we flattened the video frame at each time point into vectors, mean corrected and normalized the vectors and measured similarity for all pairs of frames as the cosine of the angle between these vectors (concretely, the ith–jth entry of the similarity matrix gives the cosine of the angle between the video frames at times i and j). We also visualized the average, over positions, of each frame's similarity to those ahead of it and behind it as a function of distance. We repeated the same analyses on the feature-specific RPEs, calculating the similarity in the feature-specific RPEs at each time point. The feature-specific RPEs here were derived by running the agent with the trained weights from the normal maze described above but not allowing the agent to change its view angle in the stem of the maze (always fixed at 0°). For Extended Data Fig. 4, we repeated the same analysis but calculated the similarity with the position-lagged scalar RPE instead.

Confirmatory versus disconfirmatory cue responses

We defined confirmatory cues as cues that appeared on the side with more evidence so far and disconfirmatory cues as cues that appeared on the side with less evidence so far. If the agent or mouse had seen an equal number of cues on both sides, the next cue was defined as a neutral cue. For the neural data, we isolated cue kernels as in Engelhard et al.[9] with some modifications. Instead of using contralateral and

ipsilateral cues, we used predictors, including contralateral and ipsilateral cues with contralateral evidence, neutral evidence and ipsilateral evidence so far. For Fig. 4b, we selected those feature-specific RPE units that were modulated by cue onset and the ten time steps after cue onset, regardless of left or right cues. For Fig. 4c, we selected among the neurons modulated by cues from the encoding model ($N = 77/303$), plotting only the units modulated by cues ($N = 62/303$).

For Extended Data Fig. 5a,b, we applied the same regression for Fig. 4c to the original data split to recordings from the left and right hemispheres. For the feature-specific RPE model responses in Extended Data Fig. 5c,d, we split the units to left cue- versus right cue-responsive units based on the heat maps in Fig. 2g. The units were sorted by eye, choosing the units that responded to the left and right cues, respectively, at cue onset and throughout the ten time steps after cue onset and excluding units that responded at cue offset.

Outcome period responses

In Fig. 5a,d, the scalar RPE was calculated by summing the feature-specific RPE units. For the model responses at outcome time for Fig. 5b,e, we normalized each unit's activity five time steps before and five time steps after reward across 5,000 trials, so the minimum is 0 and the maximum is 1. We then averaged across trials and took the response at reward time. For the neural responses at outcome time for Fig. 5c,f, we matched the original empirical paper[9] and calculated the average activity in the first 2 s after the onset of the outcome period and baseline corrected by subtracting the average activity from the 1-s period preceding the outcome. For the histograms in Fig. 5b,c,e,f, a two-sided Wilcoxon signed rank test was performed to determine the P value for the median (yellow line).

SR modeling

We modeled a family of DA responses using a vector of PEs for an SR, based on Gardner et al.[31]. For comparability with the feature-specific RPE model, we used the same network-derived features $\vec{\phi}$ from the feature-specific RPE model (LSTM units from Fig. 2b) as targets for these successor predictions. Thus, we have a successor feature model whose weights we learned using Algorithm 3 of Barreto et al.[81]. Given the input state $\vec{\phi}(s_t)$, we trained a set of weights W to learn a vector of successor features $\vec{SF}$ with each SF_i trained with its own sensory PE δ_i (Fig. 6a). We z-scored the target features $\vec{\phi}(s_t)$ before training the successor feature model. To calculate the sensory PEs (to match with the feature-specific PEs in Fig. 1c), we trained a weight matrix W with rows $\vec{w}_i$, such that each successor feature

$$SF_{i,t} = \vec{\phi}_t \cdot \vec{w}_i. \tag{2}$$

Each $\vec{w}_i$ for each successor feature SF_i was then updated using the classic scalar RPE recursive step; that is, each has its own sensory PE δ_i

$$\delta_{i,t} = \phi_{i,t} + \gamma SF_{i,t+1} - SF_{i,t}. \tag{3}$$

We updated each $\vec{w}_i$ accordingly using

$$\vec{w}_i \leftarrow \vec{w}_i + \alpha\delta_{i,t}\vec{\phi}_t. \tag{4}$$

In total, 64 successor features were trained to match with the vector of 64 input features $\vec{\phi}_t$. We trained using this recursive update for 15,000 trials until the weights converged, using discount factor $\gamma = 0.99$ and a learning rate $\alpha = 0.001$.

After training our weight matrix W, we reran the algorithm with the learned weights frozen to calculate the 64 sensory PEs for the 64 features. In Fig. 6b–d, we generated the heat maps with the same methods as with the feature-specific RPE model's results in Fig. 2e–g. In our analyses in Fig. 6e, we compare the feature-specific RPE model's scalar RPE with the averaged sensory PE, which is the average over the 64 sensory PEs. The outcome period plots in Fig. 6f–i were generated the same way as in Fig. 5a,b,d,e.

Distributional reinforcement learning modeling

To model the features with a distributional reinforcement learning model, we adapted a TD-1 algorithm to train weights for 31 distribution channels based on algorithms specified in refs. 27,82 and briefly summarized below. Each channel has two learning rates, α_i^+ for positive PEs and α_i^- for negative PEs, chosen so that the expectiles $\tau_i = \frac{\alpha_i^+}{\alpha_i^+ + \alpha_i^-}$ evenly tile the range between 0 and 1 (exclusive) with $\alpha_i^+ + \alpha_i^- = 0.001$.

The input features for the model were (again, for comparability with the feature-specific RPE) the z-scored features $\vec{\phi}$ from the deep RL network. We trained weights that mapped the features to distributional value channels, including a bias term. We trained a set of linear feature weights for each channel by, at each time step, directly predicting the ($\gamma = 0.99$ discounted) final outcome of the trial (effectively, expectile regression or TD-1; this avoids distributional imputation in a bootstrapped TD-0 update).

After training for 15,000 trials, we froze the converged weights to calculate the 31 expectile PEs for the distribution channels. We then computed TD-0 Bellman errors at each step with an imputation algorithm[82] to sample the one-step bootstrap backup value for each of the 31 channels from the ensemble. Each unit's per trial positive and negative PEs were weighted by its α_i^+ and α_i^-, respectively.

To generate plots for Fig. 7b–i, we repeated the same methods as in Fig. 6b–i, with one difference for the heat maps in Fig. 7b–d; specifically, in the heat maps, we used the original ordering of the distribution channels because the channels themselves already have a natural order from pessimistic to optimistic channels.

APE models

The APE framework may be formally thought of as the standard RL setup, but the reward team is replaced with the execution of actions. Although previous work[28,29] implemented APE in the manner of the Rescorla–Wagner rule (that is, without a future-predictive term), we extended these models to a full TD formulation analogous to Schultz et al.[1]. It is worth noting that Rescorla–Wagner is a special case of this with maximal temporal discounting.

The scalar APE model used to describe the Parker et al.[6] results correspondingly used a variant of the typical Q-learning approach. For a hypothetical DA population with some preferred action a (for example, left lever press), that is,

$$\delta_t{}^a = I_{a_t = a} + \gamma V^a(s_{t+1}) - V^a(s_t) \tag{5}$$

$$V^a(s_t) \leftarrow V^a(s_t) + \eta\delta_t{}^a. \tag{6}$$

Here, $\delta_t{}^a$ is the PE at time t and corresponds to the DAergic response. $I_{a_t = a}$ is an indicator for the population's preferred action and is equal to 1 if $a_t = a$ and 0 otherwise. $\gamma \in [0,1]$ is a temporal discounting factor. $V^a(s)$ denotes the value of state s, which is equal to the expected cumulative discounted number of times a will be executed during the remainder of the trial. η is a learning rate parameter.

The feature-based APE model used to describe Jin and Costa[26] is similar. Indeed, the approach used to derive it was identical; we treated the actions as being the source of the rewards,

$$\delta^a{}_{i,t} = \frac{I_{a_t = a}}{N} + \gamma V^a{}_{i,t+1} - V^a{}_{i,t} = \frac{I_{a_t = a}}{N} + w^a{}_{i,t}(\gamma\phi^a{}_{i,t+1} - \phi^a{}_{i,t}) \tag{7}$$

$$w^a{}_t \leftarrow w^a{}_t + \eta\delta^a{}_t\phi^a{}_t. \tag{8}$$

Here, $\delta^a{}_{i,t}$ is the PE at time t for feature channel i and corresponds to the DAergic response of that DA neuron. N is the total number of

channels. $w^a_{i,t}$ is the ith entry in the value weight vector at time t. $\phi^a_{i,t}$ is the ith entry in the state vector at time t.

Our models of the Parker et al.[6] task and the Jin and Costa[26] task were essentially the same with some slight differences in their parameters, so we will first describe their shared abstract structure and then elaborate separately on the parameters. Both tasks began at a 'start' state in which levers were presented. To randomize the timing of subsequent actions, the start state had a probabilistic self-transition. After exiting the start state, the agent needed to press a lever a fixed number times to progress to the outcome states; they chose which lever to press at each time point according to a softmax policy over the true reward values for each lever. For each press, they first progressed to a 'premotor' state that preceded the chosen lever press state (that is, led to it with probability 1 on the next time step) and then to the 'press' state itself (where the agent that preferred the corresponding action earned a reward of 1). The final choice state similarly had a probabilistic self-transition, after which the agent entered a reward state or an omission state depending on the levers it had pulled. An abstract state diagram may be found in Extended Data Fig. 7.

In the Parker et al.[6] task, the agent only needed to press a lever once to reach the outcome states. The softmax policy has an inverse temperature $\tau = 0.25$, the start state self-transition has probability 0.8, the preoutcome self-transition has probability 0.95, the temporal discount factor for the reward model is $\gamma_{RPE} = 0.95$, and the temporal discount factor for the APE model is $\gamma_{APE} = 0.5$. Discounting is harsher for the APE model to match the value used in modeling the Jin and Costa[26] task (described below), but we note that the specific values used do not change the qualitative pattern of the results but only affect the scale of the responses in Fig. 8d (left).

In our simplification of the Jin and Costa[26] task, the agent needed to press a lever three times to reach the outcome states. To randomize the within-sequence press timing, each of the choice states also has a self-transition with probability 0.1. The start state self-transition has probability 0.8. The softmax policy has an inverse temperature $\tau = 0.25$, and the temporal discount factor for both the scalar and feature-specific models is $\gamma = 0.5$. In general, we believe that harsher discounting is appropriate for the APE models because the computational goal of action prediction is presumably preparing a single upcoming response rather than computing a long-run average. However, we note, as described above, that the specific value of γ does not affect the qualitative pattern of the results (there are DA neurons with specific preferred presses that emit a biphasic response and those neurons will change their tuning in response to changes in tuning upstream), only the relative magnitudes of the peaks and troughs in the DA response.

Statistics and reproducibility

We verified reproducibility of the modeling by replicating our analyses on a second independent network run starting from the publicly posted code. Results from a retrained network are shown in Extended Data Fig. 1, which used the same hyperparameters as the original run and took 162 million time steps to learn. One minor difference was the scheduling of the hyperparameter n_{steps}, which we lowered to 20 early in training and increased to 50 for the last 28 million time steps. The lower n_{steps} helped improve performance in part because it meant that the A2C algorithm had fewer trajectories than it needed to sample and explore and therefore could more easily learn the task. After performance was above chance, a higher n_{steps} value was needed to match the original performance to allow the agent to integrate the cues across the entire trial to maintain the correct tower difference count. The plots for Extended Data Fig. 1 were generated the same as their counterparts in Figs. 2c–g, 4b and 5a,b,d,e. Most of our statistical tests were nonparametric and do not make data distributional assumptions; for some tests, we assumed the data distribution to be normal, but this was not formally tested.

We also report reanalysis of a previously reported neural dataset[9]. Because the data were previously obtained, we did not use statistical methods to determine the sample size ahead of time, but the number of recorded neurons ($N = 303$) was at the time of original publication the largest ever reported number of identified DA neurons in a single study. Because the original data contained only a single experimental group, randomization and blinding were not used.

Reporting summary

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

Data availability

Both the model and neural data that support the findings of this study are available on Figshare at https://doi.org/10.6084/m9.figshare.25752450 (ref. 83). A description of the data can be found with the code at https://github.com/ndawlab/vectorRPE/. Source data are provided with this paper.

Code availability

Code used for the deep RL model, VR environment and analysis of the data to reproduce the figures can be found at https://github.com/ndawlab/vectorRPE/.

References

77. Pinto, L. et al. An accumulation-of-evidence task using visual pulses for mice navigating in virtual reality. *Front. Behav. Neurosci.* **12**, 36 (2018).
78. Aronov, D. & Tank, D. W. Engagement of neural circuits underlying 2D spatial navigation in a rodent virtual reality system. *Neuron* **84**, 442–456 (2014).
79. Brockman, G. et al. OpenAI Gym. Preprint at https://arxiv.org/abs/1606.01540 (2016).
80. Hill, A. et al. Stable baselines. *GitHub* https://github.com/hill-a/stable-baselines (2018).
81. Barreto, A. et al. Successor features for transfer in reinforcement learning. In *Proc. 31st Conference on Neural Information Processing Systems* (eds. Guyon, I. et al.) 4055–4065 (Curran Associates, Inc., 2017).
82. Rowland, M. et al. Statistics and samples in distributional reinforcement learning. In *Proc. 36th International Conference on Machine Learning*, Vol. 97 (eds. Chaudhuri, K. & Salakhutdinov, R.) 5528–5536 (PMLR, 2019).
83. Lee, R. S., Sagiv, Y., Engelhard, B., Witten, I. B. & Daw, N. D. A feature-specific prediction error model explains dopaminergic heterogeneity. *Figshare* https://doi.org/10.6084/m9.figshare.25752450 (2024).

Acknowledgements

We thank A. Luna and J. Lopez for their help with the VR software system; W. Dabney for discussion on the distributional RL model and providing the imputation function for the distributional RL model; M. Lee and E. Grant for help with training the deep RL network; P. Dayan, A. Kahn and L. Brown for comments on this work; and the BRAIN CoGS team and the laboratories of N.D.D. and I.B.W. for their help. This work was supported by an NSF GRFP (R.S.L.), 1K99MH122657 (B.E.), National Institutes of Health R01 DA047869 (I.B.W.), U19 NS104648-01 (I.B.W.), ARO W911NF-16-1-0474 (N.D.D.), ARO W911NF1710554 (I.B.W.), Brain Research Foundation (I.B.W.), Simons Collaboration on the Global Brain (I.B.W.) and the New York Stem Cell Foundation (I.B.W.).

Author contributions

R.S.L., I.B.W. and N.D.D. conceived the project. B.E. and I.B.W. conducted the original neural recording experiments, and B.E.

provided software, interpretation and analysis. R.S.L. and N.D.D. developed the model, and Y.S. extended it. R.S.L. wrote software, trained the deep network and conducted data analyses. R.S.L. and Y.S. conducted model simulations. R.S.L., Y.S., I.B.W. and N.D.D. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

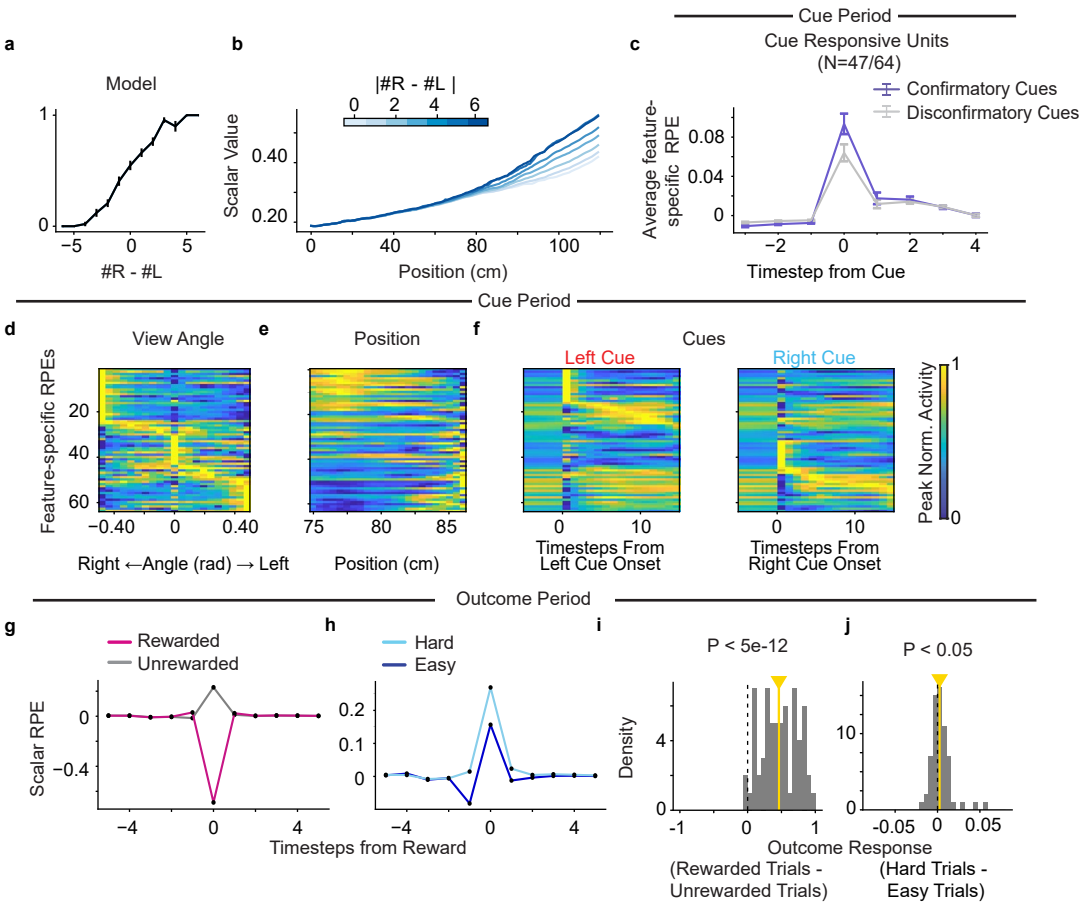
Extended data is available for this paper at https://doi.org/10.1038/s41593-024-01689-1.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41593-024-01689-1.

Correspondence and requests for materials should be addressed to Ilana B. Witten or Nathaniel D. Daw.

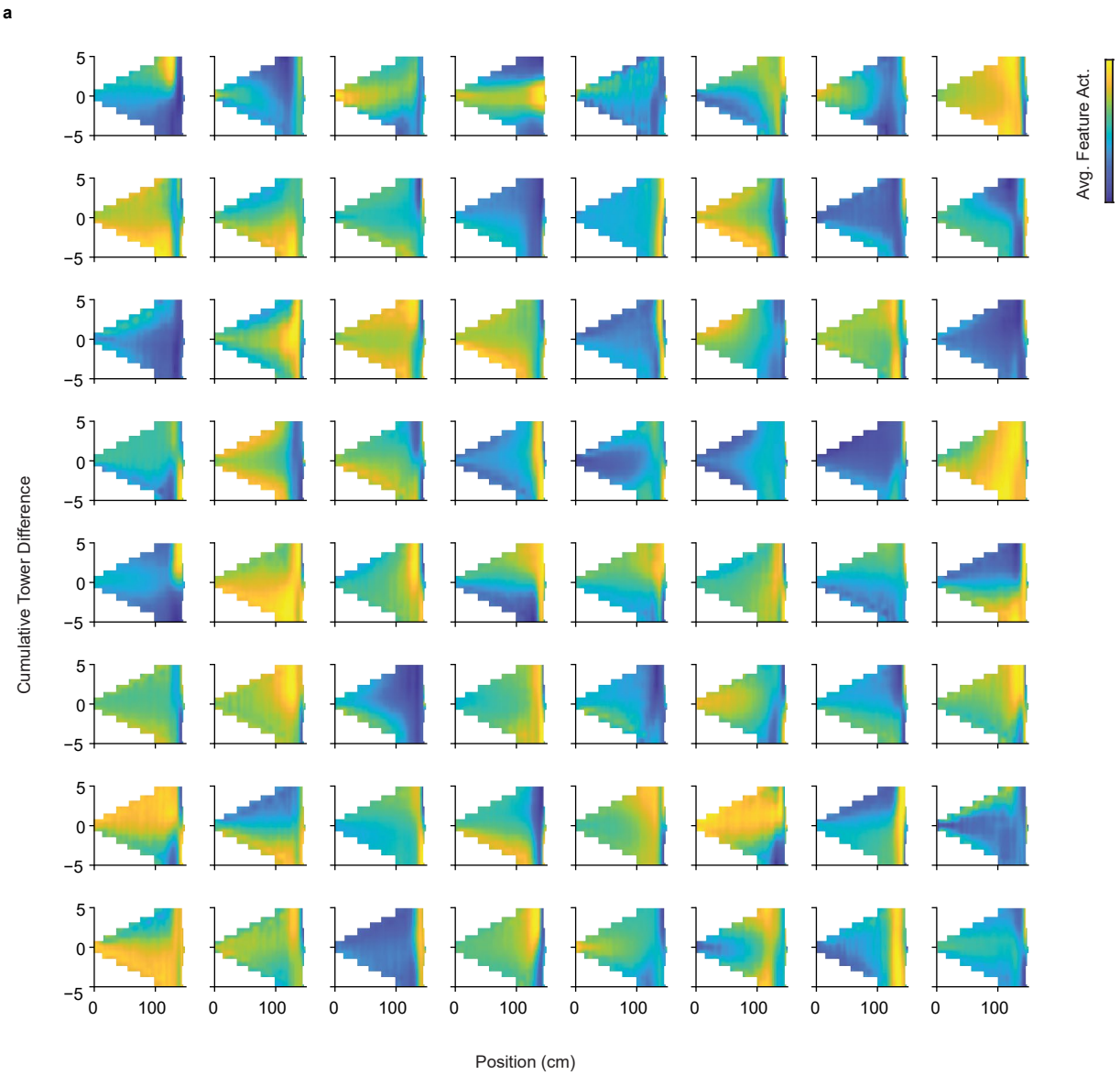
Peer review information *Nature Neuroscience* thanks Talia Lerner and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

https://doi.org/10.1038/s41593-024-01689-1



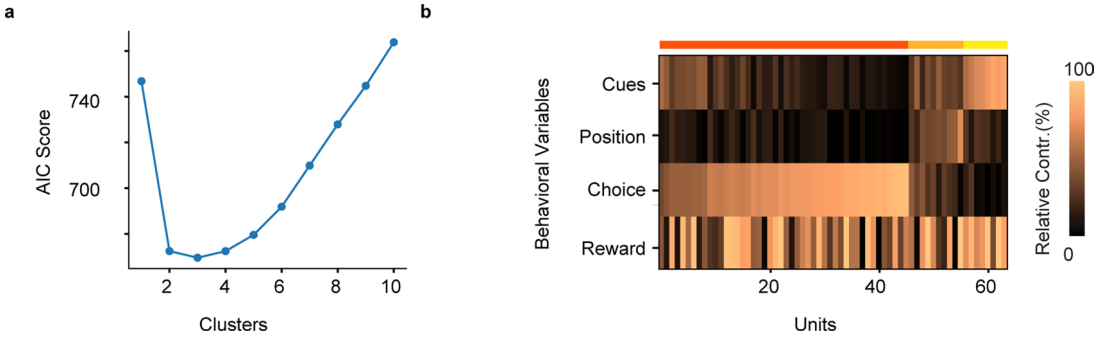
Extended Data Fig. 1 | Deep RL network retrained with the same task and a different seed. (a) Psychometric curve showing the retrained agent's performance, with error bars showing ±1 s.e.m. (N = 1000 trials) **(b)** Retrained agent's scalar value during the cue period decreased as a function of trial difficulty (defined as the absolute tower difference, blue gradient). **(c)** Retrained agent's feature-specific RPE units' response to confirmatory (purple) and disconfirmatory (gray) cues. Response is averaged across cue-sensitive units only (N = 47/64), with error bars indicating ±1 s.e.m. **(d)** Retrained agent's feature-specific RPE units averaged across trials plotted with respect to view angle. Each row represents one unit's peak normalized response to the view angle. **(e-f)** Same as **(d)** but for the agent's **(e)** position and **(f)** left (red) and right (blue) cues. **(g)** Retrained agent's scalar RPE time-locked at reward time for rewarded (magenta) and unrewarded trials (gray). **(h)** Same as **(g)** but for rewarded trials with different trial difficulties, with hard trials (light blue) and easy trials (dark blue) defined like in Fig. 5d. **(i)** Histogram of retrained agent's feature-specific RPE units' response to reward minus omission at reward time (P = 4.06e-12 for two sided Wilcoxon signed rank test, N = 64). Yellow line indicates median. **(j)** Same as **(i)** but for rewarded trials plotted against trial difficulty (P = 0.028 for two sided Wilcoxon signed rank test, N = 64). In **(j)**, there is an outlier data point at 0.16 for a feature-specific RPE unit showing strong reward expectation modulation.

https://doi.org/10.1038/s41593-024-01689-1



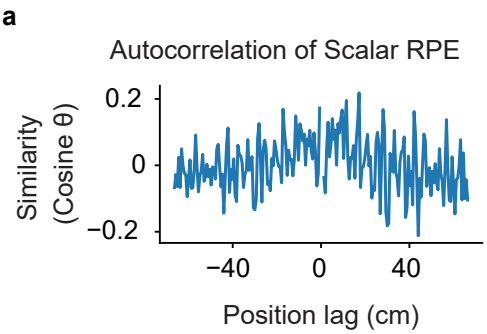
Extended Data Fig. 2 | Tuning of 64 LSTM feature units to position and evidence. (a) Each panel shows an individual feature unit and how it is tuned to the agent's position in the maze and the cumulative tower difference at that position.

https://doi.org/10.1038/s41593-024-01689-1



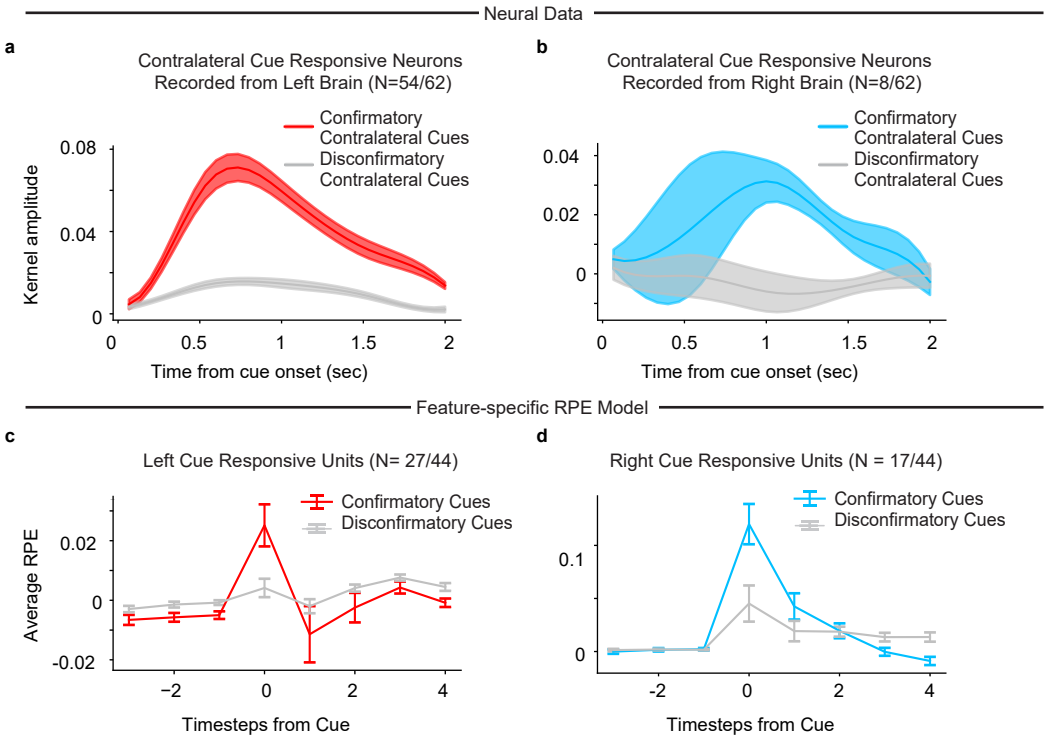
Extended Data Fig. 3 | Feature-specific RPEs clustered to behavioral features of the task to match Engelhard et al[4] clustering analysis. (a) Optimal number of clusters was 3, selected by minimizing the AIC scores for models with different numbers of clusters. **(b)** Relative contribution of the behavioral features including cues, position, choice and reward response for the 64 units sorted based on the highest probability belonging to the cluster, with colored lines on top indicating the cluster identity. Relative contribution is defined as the percentage of the explained variance for the partial model not including the variable versus the full model.

https://doi.org/10.1038/s41593-024-01689-1



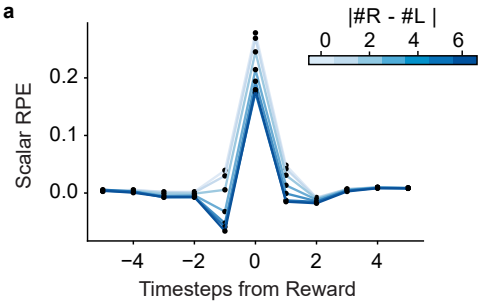
Extended Data Fig. 4 | Scalar RPE signal does not reflect the incidental high-dimensional features. (a) Scalar RPE signal autocorrelated across time (similarity defined as the cosine of position-lagged scalar RPE responses) does not show peaks at position lag 43 cm for the wall-pattern repetition location.

https://doi.org/10.1038/s41593-024-01689-1



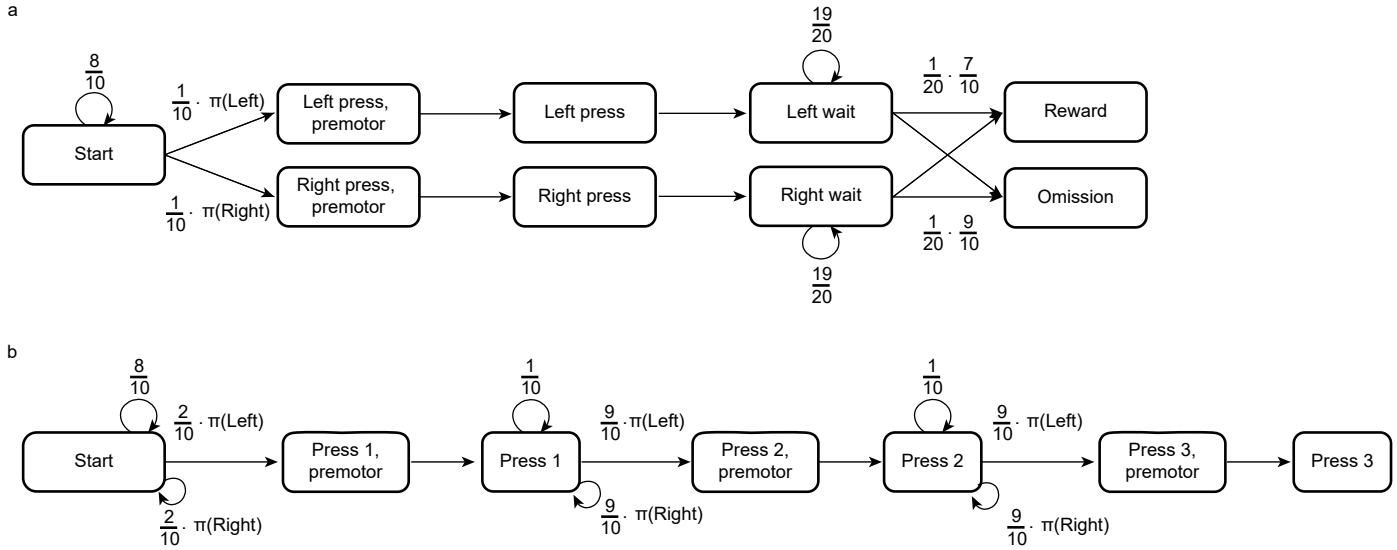
Extended Data Fig. 5 | Cue responses in model and DAergic neurons represent a vector code of lateralized response and RPE. (a) Average responses of the contralateral cue responsive DA neurons[9] only recorded from the left hemisphere (N = 54/62 neurons, subset of contralateral cue responsive neurons from Fig. 4c) for confirmatory (red) and disconfirmatory (gray) contralateral cues. Colored fringes represent ±1 s.e.m. for kernel amplitudes. **(b)** Same as **(a)** but for DA neurons recorded on the right hemisphere (N = 8/62) responding to confirmatory (blue) and disconfirmatory (gray) confirmatory cues. **(c-d)** Same as **(a-b)**, but for feature-specific RPE model units responding to **(c)** left cues specifically (N = 27/44, subset of the cue responsive neurons from Fig. 4b that were modulated by left cues only) and **(d)** right cues specifically (N = 17/44). Error bars indicate ±1 s.e.m.

https://doi.org/10.1038/s41593-024-01689-1



Extended Data Fig. 6 | Scalar RPE shows fine-grained reward expectation modulation. (a) Scalar RPE's response modulated by reward expectation given by the difficulty of the task, defined as the absolute value of the final tower difference (blue gradient) of the trial.

https://doi.org/10.1038/s41593-024-01689-1



Extended Data Fig. 7 | State diagrams for simulations of the Parker et al[6] and Jin and Costa[26] tasks in Fig. 8. (a) State diagram for the Parker et al[6] task simulation. **(b)** State diagram for the Jin and Costa[26] task simulation. In both panels, arrows indicate probabilistic transitions between states, with probabilities described by the arrow labels. Unlabeled arrows denote transitions with the remaining probability to make the total sum to 1. $\pi(x)$ refers to the probability of executing action x under the agent's behavioral policy π.

nature portfolio

Corresponding author(s): Nathaniel Daw, Ilana Witten

Last updated by author(s): 05-12-2024

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	Model simulation data was collected using custom Matlab code (Matlab 2019a) based on the ViRMEn package (https://pni.princeton.edu/pni-software-tools/virmen-virtual-reality-matlab-engine, commit d4cd673 from github.com/ndawlab/vectorRPE), custom Python code (Python 3.7), Stable Baselines (version 2.10.0, Advantage Actor Critic (A2C) algorithm used within the Stable Baselines package), Open AI Gym (version 0.14.0), and Tensorflow (version 1.14.0). Neural data was collected using Matlab code (Matlab 2015b, Mathworks Inc) based on the ViRMEn package (https://pni.princeton.edu/pni-software-tools/virmen-virtual-reality-matlab-engine).
Data analysis	Analyses were done using custom Matlab code (Matlab 2019a, Mathworks Inc) and custom Python code (Python 3.7). The code is available in a public github repository: https:// github.com/ndawlab/vectorRPE

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

Both the model and neural data that support the findings of this study is available on Figshare at https://doi.org/10.6084/m9.figshare.25752450. Description of the data can be found with the code at https://github.com/ndawlab/vectorRPE/.

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	n/a
Reporting on race, ethnicity, or other socially relevant groupings	n/a
Population characteristics	n/a
Recruitment	n/a
Ethics oversight	n/a

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

Life sciences study design

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

Sample size	All neural data used in this paper were previously published data from Engelhard et al. 2019 Nature, and thus we did not determine the sample size ahead of time. In the original paper, statistical calculations were not used to choose the sample size.
Data exclusions	In the original paper, neurons were excluded from the study using pre-established criteria. Neurons were selected for analysis based on visual inspection of recording stability, using both the images as well as ΔF/F traces. Only neurons that were stable for at least 50 trials were included in the dataset.
Replication	In the original paper, statistical methods were used to ensure that the neuronal findings were not due to inter-mice variability.
Randomization	In the original paper, there was only one experimental group, thus no randomization was needed.
Blinding	In the original paper, there was only one group thus no group allocation blinding was necessary.

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

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals	All neural data used in this study was previously published data from Engelhard et al Nature 2019. 2 strains of mice were used: either male DAT::IRES-Cre mice (n=14, The Jackson Laboratory strain 006660) or male mice resulting from the cross of DAT::IREScre mice and the GCaMP6f reporter line Ai148 mice (n=17, Ai148xDAT::cre, The Jackson Laboratory strain 030328). All mice were 2-6 months old during their use in the study.
Wild animals	The original study did not involve wild animals.
Reporting on sex	The original study only considered male mice.
Field-collected samples	The original study did not involve samples collected from the field.
Ethics oversight	All experimental procedures were conducted in accordance with the National Institutes of Health (NIH) guidelines and were reviewed by the Princeton University Institutional Animal Care and Use Committee (IACUC).

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

Plants

Seed stocks	n/a
Novel plant genotypes	n/a
Authentication	n/a