Article

Multi-timescale reinforcement learning in the brain

https://doi.org/10.1038/s41586-025-08929-9

Received: 12 November 2023

Accepted: 21 March 2025

Published online: 4 June 2025

Check for updates

Paul Masset[1,2,3,4,11]✉, Pablo Tano[5,11], HyungGoo R. Kim[1,2,6,7], Athar N. Malik[1,2,8,9], Alexandre Pouget[5]✉ & Naoshige Uchida[1,2,10]✉

To thrive in complex environments, animals and artificial agents must learn to act adaptively to maximize fitness and rewards. Such adaptive behaviour can be learned through reinforcement learning[1], a class of algorithms that has been successful at training artificial agents[2–5] and at characterizing the firing of dopaminergic neurons in the midbrain[6–8]. In classical reinforcement learning, agents discount future rewards exponentially according to a single timescale, known as the discount factor. Here we explore the presence of multiple timescales in biological reinforcement learning. We first show that reinforcement agents learning at a multitude of timescales possess distinct computational benefits. Next, we report that dopaminergic neurons in mice performing two behavioural tasks encode reward prediction error with a diversity of discount time constants. Our model explains the heterogeneity of temporal discounting in both cue-evoked transient responses and slower timescale fluctuations known as dopamine ramps. Crucially, the measured discount factor of individual neurons is correlated across the two tasks, suggesting that it is a cell-specific property. Together, our results provide a new paradigm for understanding functional heterogeneity in dopaminergic neurons and a mechanistic basis for the empirical observation that humans and animals use non-exponential discounts in many situations[9–12], and open new avenues for the design of more-efficient reinforcement learning algorithms.

Many of the recent advances in artificial intelligence rely on temporal difference (TD) reinforcement learning (RL) in which the TD learning rule is used to learn predictive information[1] (equation (2)). By updating current estimates on the basis of future expected estimates, TD methods have been remarkably successful in solving tasks that require predicting future rewards and planning actions to obtain them[2,13].

The standard formulation of TD learning assumes a fixed discount factor (that is, a single-learning timescale), which, after convergence, results in exponential discounting: the value of a future reward is reduced by a fixed fraction per unit time (or timestep). Although this formulation is important for simplicity and self-consistency of the learning rule, it is well known that humans and other animals do not exhibit exponential discounting when faced with inter-temporal choices. Instead, they tend to show hyperbolic discounting: there is a fast drop in value followed by a slower rate for further delays[9,10]. Far from being irrational, non-exponential discounting can be optimal depending on the uncertainty in the environment, as has been documented in the behavioural economics and foraging literature[11,12]. Humans and animals can modulate their discounting function to adapt to the temporal statistics of the environment and maladaptive behaviour can be a signature of mental state or disease[14,15].

The TD rule can potentially be extended to learn more complex predictive representations than the mean discounted future reward of the traditional value function, in both artificial neural systems[16–19] and biological neural systems[20–24]. A growing body of evidence points to the rich nature of temporal representations in biological systems[25,26] and particularly in the basal ganglia[27–31]. Understanding how these rich temporal representations are learned remains a key question in neuroscience and psychology. An important component across most temporal-learning proposals is the presence of multiple timescales[22,31–35], which enables capturing temporal dependencies across a diverse range of durations: shorter timescales typically handle rapid changes and immediate dependencies, whereas longer timescales capture slow-changing features or long-term dependencies[34]. Furthermore, work in artificial intelligence has suggested that the performance of deep RL algorithms can be improved by incorporating learning at multiple timescales[19,36]. We therefore ask whether RL in the brain exhibits such multi-timescale properties.

We first investigate the computational implications of multi-timescale RL. We then show that dopaminergic neurons encode predictions at diverse timescales, providing a potential neural substrate for multi-timescale RL in the brain.

[1]Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, USA. [2]Center for Brain Science, Harvard University, Cambridge, MA, USA. [3]Department of Psychology, McGill University, Montréal, Québec, Canada. [4]Mila – Quebec Artificial Intelligence Institute, Montréal, Québec, Canada. [5]Department of Basic Neuroscience, Université de Genève, Geneva, Switzerland. [6]Department of Biomedical Engineering, Sungkyunkwan University, Suwon, Republic of Korea. [7]Center for Neuroscience Imaging Research, Institute for Basic Science (IBS), Suwon, Republic of Korea. [8]Department of Neurosurgery, Warren Alpert Medical School of Brown University, Providence, RI, USA. [9]Norman Prince Neurosciences Institute, Rhode Island Hospital, Providence, RI, USA. [10]Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University, Cambridge, MA, USA. [11]These authors contributed equally: Paul Masset, Pablo Tano. ✉e-mail: paul.masset@mcgill.ca; alexandre.pouget@unige.ch; uchida@mcb.harvard.edu

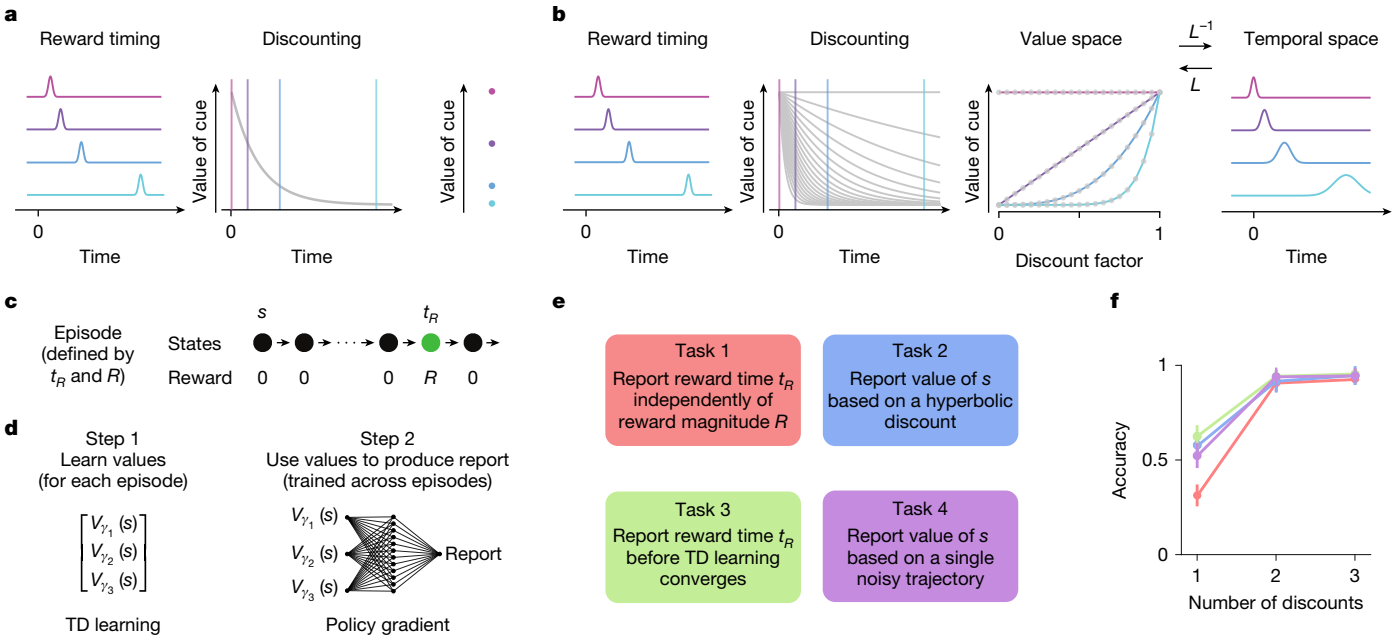


Fig. 1 | Computational advantages of multi-timescale RL. a, In single-timescale value learning, the value of a cue (at $t = 0$) predicting future rewards (left) is evaluated by discounting these rewards with a single exponential discounting function (middle). The expected reward size and timing are encoded, but confounded, in the value of the cue (right). **b**, In multi-timescale value learning, the same reward delays are evaluated with multiple discounting functions (middle left). The relative value of a cue as a function of the discount depends on the reward delay (middle right). A simple linear decoder based on the Laplace transform can thus reconstruct both the expected timing and the magnitude of rewards (right). **c**, Experiment to compare single-timescale versus multi-timescale learning. t_R and R are fixed within each episode and varied across episodes. **d**, In each episode (defined by a specific t_R and R), the value function is learned via tabular updates, using multiple discount factors (step 1). Given these values, step 2 consists of training a non-linear decoder to maximize the accuracy of a task-specific report. The decoder is trained across episodes using policy gradient. **e**, The architecture is trained across four tasks to highlight computational advantages of multi-timescale RL, including decoupling information about reward size and reward timing, the ability to learn with arbitrary discount functions, the ability to recover reward timing information before convergence and the ability to control the inductive bias (see main text and Methods). **f**, Mean accuracy is reported after 2,000 training episodes as the fraction of correct responses (see Methods). 'Three discounts' correspond to the γ-ensemble [0.6,0.9,0.99], 'one discount' to the top-performing ensemble across {[0.6,0.6,0.6], [0.9,0.9,0.9], [0.99,0.99,0.99]} and analogous for 'two discounts'. The error bars are the standard deviations (s.d.) across 100 test episodes and 3 trained policy gradient networks.

Advantages of multi-timescale RL

We first examined the computational advantages of RL agents using multiple timescales over those utilizing a single timescale. We start with a simple example environment in which a cue (s) predicts a future reward at a specific time (Fig. 1a; see Methods). In standard RL algorithms, the agent learns to predict future rewards, compressed into a single scalar value, that is, the sum of discounted future rewards expected from the current state[1]:

$$V(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{1}$$

where $V(s)$ is the value of the state s, r_t is reward at time t, γ is the discount factor ($0 < \gamma < 1$) and E denotes the expectation over stochasticity in the environment and actions. This exponentially functional form for the temporal discount (γ^t) is not arbitrary. It is naturally produced by the TD learning rule, a bootstrapping mechanism that updates the value estimate for state s after transitioning from s to s' and receiving reward r:

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)] \tag{2}$$

where α is the learning rate. This update process converges to the values defined above under very general conditions[1] and has been experimentally proven to be an extremely robust and efficient learning rule in training deep RL systems[13] and at characterizing the firing of dopaminergic neurons in the midbrain[6–8].

Now consider multi-timescale learning (Fig. 1b). Let V_i be the value learned using a discount γ_i. Moving the discount factor γ out of the expectation in equation (1), values can be rewritten (truncating at $t = T$) as

$$V_i = \begin{bmatrix} 1 & \gamma_i^{\Delta t} & \gamma_i^{2\Delta t} & \dots & \gamma_i^{T} \end{bmatrix} \begin{bmatrix} E(r|t=0) \\ E(r|t=\Delta t) \\ E(r|t=2\Delta t) \\ \vdots \\ E(r|T) \end{bmatrix} \tag{3}$$

Where we assumed that timestep transitions are discrete and of size Δt (see Methods). Thus, single-timescale learning projects all the timestep-specific expected rewards ($E(r|t)$) onto a single scalar (V_i) through exponential discounting (Fig. 1a) and therefore entangles reward timing and reward size. When learning with multiple timescales, instead of collapsing all future rewards onto a single scalar, there is a vector of value predictions, each computing value with its own discount factor γ_i [ref. 21]:

$$\begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} 1 & \gamma_1^{\Delta t} & \gamma_1^{2\Delta t} & \dots & \gamma_1^{T} \\ 1 & \gamma_2^{\Delta t} & \gamma_2^{2\Delta t} & \dots & \gamma_2^{T} \\ 1 & \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma_n^{\Delta t} & \gamma_n^{2\Delta t} & \dots & \gamma_n^{T} \end{bmatrix} \begin{bmatrix} E(r|t=0) \\ E(r|t=\Delta t) \\ E(r|t=2\Delta t) \\ \vdots \\ E(r|T) \end{bmatrix} = \mathbf{L}E(r|t) \tag{4}$$

The last equality shows that the array of values learned with multiple discounts (value space in Fig. 1b) corresponds to the z-transform

(that is, the discrete Laplace transform (**L**)) of the array that indicates the expected reward at all future timesteps (temporal space in Fig. 1b). As the z-transform is invertible, the agent using TD learning with multiple timescales implicitly encodes the expected temporal evolution of rewards, which can be recovered by applying a fixed, regularized linear decoder L^{-1} to the learned values[21,37] (Fig. 1b, right panel illustrates a situation with one reward per trajectory, but this approach also works for multiple reward; see Methods and ref. 21).

RL agents performing multi-timescale learning have been shown to produce performance superior to that of single-timescale agents across a wide range of complex problems[19,38]. To illustrate the computational advantages of multi-timescale representations, we considered several example tasks, including a simple linear maze (Fig. 1c–f and Extended Data Fig. 1a–o), branching mazes (Extended Data Figs. 1p–r, 2 and 3a–e), a navigation setting (Extended Data Fig. 2f–i) and a deep Q-network (DQN) setting (Extended Data Fig. 2j–l). In the linear maze, the agent navigates through a linear track (a sequence of 15 states), where it encounters a reward of a certain magnitude (R) at a specific time point (t_R) (see Fig. 1c). The value of R and t_R changes across episodes and remains constant within episodes. Each episode is initiated by a cue presented at the initial state (s). Within each episode, the agent first learns the expected future rewards predicted by the cue (that is, the value $V_\gamma(s)$) using a simple RL algorithm (tabular TD learning) using one or multiple discount factors (step 1 in Fig. 1d). Using the learned value (or values) associated with the cue, the agent then performs various tasks, which involve producing a task-specific report by transforming the learned values using a decoder network (step 2 in Fig. 1d). As some tasks involve complex, non-linear operations over the multi-timescale values, we trained a general, non-linear decoder for each task using policy gradient (see Methods). Our goal is to evaluate the advantages of the multi-timescale value representation over the single-timescale value representation, and the degree to which these advantages can be exploited by a simple, code-agnostic decoder. Therefore, in our model, multi-timescale values are not used directly to produce behaviour. Instead, they act as an enriched state representation from which task-specific behaviour can be subsequently decoded.

Task 1: disentangling reward timing and reward magnitude

In single-timescale systems, a high value at the cue could signify a small reward in the near future or a large reward in the distant future. By contrast, the pattern of values across discount factors (middle right panel in Fig. 1b) is invariant to reward magnitude. As a result, multi-timescale agents can disentangle reward timing from reward magnitude (task 1 in Fig. 1e,f) in which the agent reports reward timing independently of reward magnitude (Fig. 1f and Extended Data Fig. 1a–c; see Methods).

Task 2: learning values with non-exponential temporal discounts

The bootstrapping process of traditional TD value learning naturally converges to exponentially discounted values. Although several tasks can be optimally solved by knowing the exponentially discounted state values (that is, where the value of a reward at time t decreases as γ^t), the optimal discount in a specific task depends on its temporal contingencies, such as its hazard rate, its horizon, the cost of time and the uncertainty over time[19,38]. Indeed, human and animal judgements are generally more consistent with a hyperbolic discount (that is, decreasing as $1/(1+\gamma t)$) than an exponential discount[9,10]. Crucially, multi-timescale systems with a diversity of exponential discounts implicitly encode the expected reward magnitudes at all future times (Fig. 1b), so they can weigh the time-specific expected rewards with any chosen discount weights to retrieve the specific discount necessitated by the task. Our result shows that only multi-timescale systems can reliably report the hyperbolic value of the cue given a diversity of exponential values (task 2; Fig. 1e,f).

Task 3: inferring temporal information before convergence

In single-timescale systems, a high value of the cue could be due to a short delay (t_R) or simply because the value estimate in equation (1) has undergone more positive updates from an initial value of 0 (for example, if it has encountered the reward a larger number of times; see Extended Data Fig. 1s–w). In multi-timescale systems, the shape of value function across discount factors encodes the proximity to rewards (Fig. 1b, medium left panel), and this shape is invariant to the number of rewards encountered, to the extent that all value estimates depart from similar baselines and share similar learning parameters. As a result, multi-timescale agents can decode the time of reward (t_R) even in situations in which learning is incomplete (task 3; Fig. 1e,f and Extended Data Fig. 1; see Methods).

Task 4: state-dependent discount factor

Single-timescale systems are either myopic or far-sighted, whereas multi-timescale systems can adjust between myopic and far-sighted perspectives, leading to more accurate value estimates in incomplete learning scenarios. Consider a slight modification of the task in Fig. 1c, in which, in addition to the large deterministic reward ($R = 1$), small stochastic rewards sampled from a Gaussian distribution are perceived at every state (see Methods). If the agent experiences the trajectory many times, the noisy rewards average out, so they do not affect the learned value of the cue. In task 4, however, the agent experiences the trajectory only once, so the noisy rewards do affect the values learned with TD learning. Given the noisy values, the goal in this task was to report the true value of the cue that would arise after experiencing the trajectory an infinite number of times (this is, ignoring the noisy rewards). When t_R is small, far-sighted estimates not only integrate R but also all the noisy rewards farther in the future, in contrast to myopic estimates, which assign greater weight to R. However, when t_R is large, only far-sighted estimates can discern R from the noisy rewards. Thus, optimal accuracy is only achievable by multi-timescale agents that can estimate t_R and then adjust accordingly between myopic and far-sighted perspectives. Although in this task the uncertainty on the value of the cue arises due to receiving small noisy rewards at every state, a similar bias also improves the accuracy of value estimates in more realistic learning scenarios, in which uncertainty arises due to incomplete exploration of the full state space, as we have also shown in more realistic branching mazes, navigation scenarios (Extended Data Fig. 2f–i) and in the Lunar Lander environment using a DQN setting in which additional timescales act as auxiliary tasks (Extended Data Fig. 2j–l; see 'The myopic learning bias' in Methods).

To summarize, in multi-timescale value systems, the vectorized learning signal robustly contains temporal information independently of reward magnitude and learning conditions. This property empowers agents to flexibly adapt their behaviour to novel temporal contingencies and focus on either myopic or far-sighted estimates depending on the current situation.

Discounting across dopaminergic neurons

In the previous section, we demonstrated the computational advantages of learning with multiple discount factors for an RL agent. Building on these findings, we next investigated whether the brain uses such multi-timescale RL. Towards this goal, we examined the activity of dopaminergic neurons, which are believed to encode the TD error term in RL algorithms.

To characterize the discounting properties of individual dopaminergic neurons, mice ($n = 8$; see Extended Data Fig. 10e) were trained in a cued delay task[27,39], in which on a given trial, one out of four distinct odour cues indicated its associated timing of a water reward (Fig. 2a). These odour cues were preceded by a trial start cue (green computer screen) by 1.25 s. The trial start cue reduced the timing uncertainty of

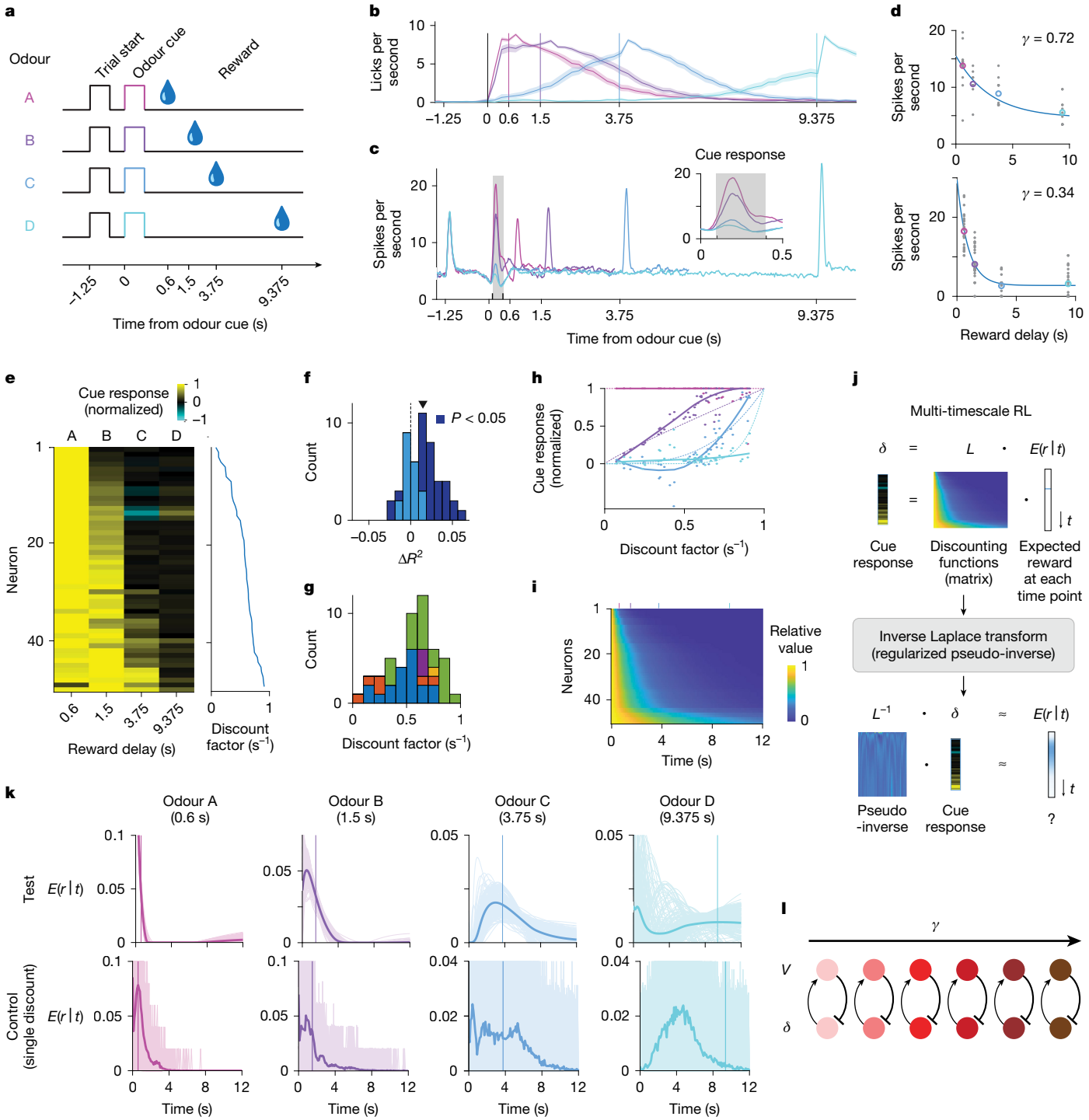


Fig. 2 | Dopaminergic neurons exhibit a diversity of discount factors that enables decoding of reward delays. a, Outline of the cued delay task structure. Image of a water droplet was created by googlefonts via SVG Repo under an Apache Licence. **b**, Anticipatory licking before reward delivery (mean across behaviour for all recorded neurons; the shaded error bar indicates 95% confidence interval using bootstrap). $n = 8$ mice. **c**, Average PSTH for the four trial types. The inset shows the firing rate in the 0.5 s following the cue predicting reward delay. The firing rate in the shaded grey box ($0.1\,s < t < 0.4\,s$) was used as the cue response in subsequent analysis. $n = 50$ dopaminergic neurons. **d**, Example cue response fits for two single neurons. **e**, Normalized cue responses across the population. For each neuron, the response was normalized to the highest response across the four possible delays. The inset shows the inferred discount factor for each neuron. **f**, Data are better fit by the exponential than the hyperbolic model (distance of mean R^2 to the unit line; the shading indicates significance for single neurons across bootstraps: $P < 0.05$ (dark blue) and $P > 0.05$ (light blue)). **g**, Distribution of inferred discount factors across neurons (mean discount factor across bootstraps). The colour indicates the animal. **h**, Shape of the normalized population response as a function of reward delay. The thick lines denote smoothed fit, the dotted lines indicate theory and the dots denote individual neurons. **i**, Discount matrix. Neurons are sorted as in panel **d**. **j**, Outline of the decoding procedure. **k**, The subjective expected timing of future reward $E(r|t)$ can be decoded from the population responses to the cue predicting reward delay. Decoding based on mean cue responses for test data (top row; see Methods) is better than using a model with a single discount factor (the mean discount factor across the population; bottom row; thin lines (light shade) indicate predictions for individual bootstraps, thick lines (dark shade within light shading) indicate mean prediction across bootstraps, and single dark vertical lines indicate reward timing; see Methods; Extended Data Fig. 4e). **l**, Model in which the RPE of each dopaminergic neuron contributes to a distinct value function (see Methods; Extended Data Fig. 7f–k).

the odour cue and ensured that the responses of dopaminergic neurons to the odour cues were mostly driven by a valuation signal rather than a saliency signal[40]. Mice showed anticipatory licking before reward delivery. The onset of the anticipatory licking was delayed for trials with cues predicting longer reward delays, indicating that the mice learned the delay contingencies (Fig. 2b). We recorded optogenetically identified single dopaminergic neurons in the ventral tegmental area ($n = 78$; see Methods). We focused our analysis on neurons ($n = 50$) that passed the selection criteria (including mean cue response firing rate above 2 spikes per second, positive goodness of fit on test data; see Methods). As expected from RL theory and the prediction error framework, the average responses to the reward cue decreased as the predicted reward timing increased[27,39] (Fig. 2c and Extended Data Fig. 3a,b). However, cue responses of individual neurons showed a great diversity of discounting across the reward delays, ranging from neurons responding strongly only to the cue indicating the shortest delay to neurons with a gradual decay of their response with cued reward delay (Fig. 2d,e).

To characterize the discount properties of individual neurons, we fit them individually using both an exponential discount model and a hyperbolic discount model. The exponential model provided a better fit to the responses of neurons than the hyperbolic model ($P = 2.2 \times 10^{-5}$, two-tailed Student's t-test, comparing the distribution across neurons of the mean (across bootstraps) difference in R^2 between the two fits; Fig. 2f and Extended Data Fig. 3c–e; see Methods) contrary to a previous observation[39]. Organism-level hyperbolic-like discounting can, therefore, arise from the diversity of exponential discounting in single neurons, as discussed above with artificial agents (Fig. 2d; see also refs. 12,19,33). This view is consistent with the wide distribution of inferred discount factors obtained across the population ($0.56 \pm 0.21\ s^{-1}$, mean ± s.d.; Fig. 2g). Fits to simulated data suggest that our estimate of inferred parameters is robust and primarily constrained by the number of trials (Extended Data Fig. 3f–j; see Methods). Furthermore, we measured behavioural discounting using the anticipatory lick rate and show that it is not correlated to the discounting measured from single dopaminergic neurons (Extended Data Fig. 5a–f; see Methods).

As we have shown above, artificial agents equipped with diverse discount factors exhibit various advantages. One key aspect contributing to these advantages is their unique ability to independently extract reward timing information, which is lacking in single-timescale agents. We next asked whether dopaminergic neurons provide a population code in which the structured heterogeneity across the population enables decoding of reward timing or the expected reward across time, $E(r|t)$. Mathematically, this transformation can be achieved by the inverse Laplace transform (or its discrete equivalent the z-transform)[21,34,37] (Fig. 2j). In our dataset, the dopaminergic cue responses for each reward delay exhibited unique shapes as a function of discount factors, suggesting that reward timing information is embedded in the dopaminergic population responses (Fig. 2h; compare with Fig. 1b). The temporal horizon across the population, which underlies these cue responses, can be visualized through the discount matrix, which indicates for each neuron the relative value of a future reward depending on the inferred discount factor (Fig. 2i).

If the dopaminergic population code is consistent with the Laplace code explored above (Fig. 1) and each dopaminergic neuron contributes to a distinct value estimate (Fig. 2l and Extended Data Fig. 7f–k), reward timing should be recoverable from the cue responses of dopaminergic neurons with a regularized discrete inverse Laplace transform of the neural activity (which does not require training a decoder). In our task, we can use the TD-error-driven cue responses (instead of the value in equation (4)) as they are driven by the discounted future value, that is, $\delta_{t_{\text{cue}}} = \gamma^{\Delta t} V_{t_{\text{cue}}+\Delta t} + C$, as $r_{t_{\text{cue}}} = V_{t_{\text{cue}}-\Delta t} = 0$; see Methods). This implies that the right-hand side of equation (4) can be approximated by the population dopamine responses. We used a pseudo-inverse of the discount matrix (computed using half of all trials) based on regularized singular value decomposition to approximate the inverse Laplace transform (Fig. 2j and Extended Data Fig. 4a–d; see Methods and ref. 21) and applied it to the cue responses of a dopaminergic neuron (computed on the held-out half of the trials). The decoder was able to predict reward timing, closely matching the true reward delay (Fig. 2k, top row). This prediction was lost if we shuffled the neuron identities, indicating that it is not a generic property of the discount matrix (Extended Data Fig. 4f). We quantified this decoding by computing a distance metric (using 1-Wasserstein distance) between the true and predicted reward delay across conditions (compared with shuffle control: $P = 1.2 \times 10^{-4}$ for 0.6-s reward delay and $P < 1.0 \times 10^{-20}$ for the other delays, one-tailed Wilcoxon signed-rank test; Extended Data Fig. 4g; see Methods). Predictions from the model were more accurate than an alternative model with a single discount factor in which the response of each neuron is interpreted as a sample from the reward timing distribution ($P_{t=0.6\,s} = 1$, $P_{t=1.5\,s} < 1.0 \times 10^{-31}$, $P_{t=3.75\,s} = 0.0135$ and $P_{t=9.375\,s} < 1.0 \times 10^{-14}$, one-tailed Wilcoxon signed-rank test; Fig. 2k, bottom row, and Extended Data Fig. 4e; see Methods). Consistent with the above observation that cue responses were fit better with exponential over hyperbolic discounting models, the accuracy of reward timing decoding was typically higher when using the discount matrix from the exponential model than the discount matrix from the hyperbolic model ($P_{t=0.6\,s} = 1$, $P_{t=1.5\,s} < 1.0 \times 10^{-31}$, $P_{t=3.75\,s} < 1.0 \times 10^{-33}$ and $P_{t=9.375\,s} < 1.0 \times 10^{-3}$, one-tailed Wilcoxon signed-rank test; Extended Data Fig. 6a–e). Furthermore, the decoding performance was comparable with simulated data with matched trial numbers, indicating that the remaining uncertainty in decoded reward timing is primarily driven by limited sample size in the data (for example, the number of neurons and the number of trials per condition; Extended Data Fig. 6f,g; see Methods). We performed the decoding analysis at the single-animal level for two of the animals for which we had a sufficient number of neurons and observed decoding of subjective reward timing (Extended Data Figs. 5g and 10e; Methods).

Together, these results establish that dopaminergic neurons compute prediction errors with a heterogeneity of discount factors and show that the structure in this heterogeneity can be exploited by downstream circuits to decode reward timing.

Ramping heterogeneity and discounting

In the task above (Fig. 2), prediction errors in dopaminergic neurons were measured through discrete transitions in the value functions at the time of cue. In more naturalistic environments, value might change more smoothly, for example, when an animal approaches a goal[41]. In these tasks, ramps in dopaminergic signalling have been initially interpreted as quantifying value functions[41,42], but have recently been shown to conform to the predictions of the TD learning model. Specifically, these ramps can be understood as moment-by-moment changes in values or as TD error along an increasingly convex value function in which the derivative is also increasing[43–45]. Here we show that some of the diversity in ramping activity across neurons can be understood as evidence for multi-timescale RL across dopaminergic neurons.

We analysed the activity of optogenetically identified dopaminergic neurons ($n = 90$ from $n = 13$ mice; Extended Data Fig. 10e; see Methods and ref. 44) while mice traversed along a linear track in virtual reality. Although mice were free to locomote, their movements did not affect the dynamics of the scene (see Methods and ref. 44 for details). At trial onset, a linear track appeared, the scene moved at continuous speed, and reward was delivered 7.35 s after motion onset (Fig. 3a). The slope of ramping across neurons was on average positive (Fig. 3b), but single neurons exhibited a diversity of ramping activity (Fig. 3b, inset, and 3e,f) ranging from monotonic upwards and downwards ramps to non-monotonic ramps.

We hypothesized that this seemingly puzzling heterogeneity can be understood as a signature of multi-timescale RL. Considering that the value function is set by the limits on the precision of internal timing

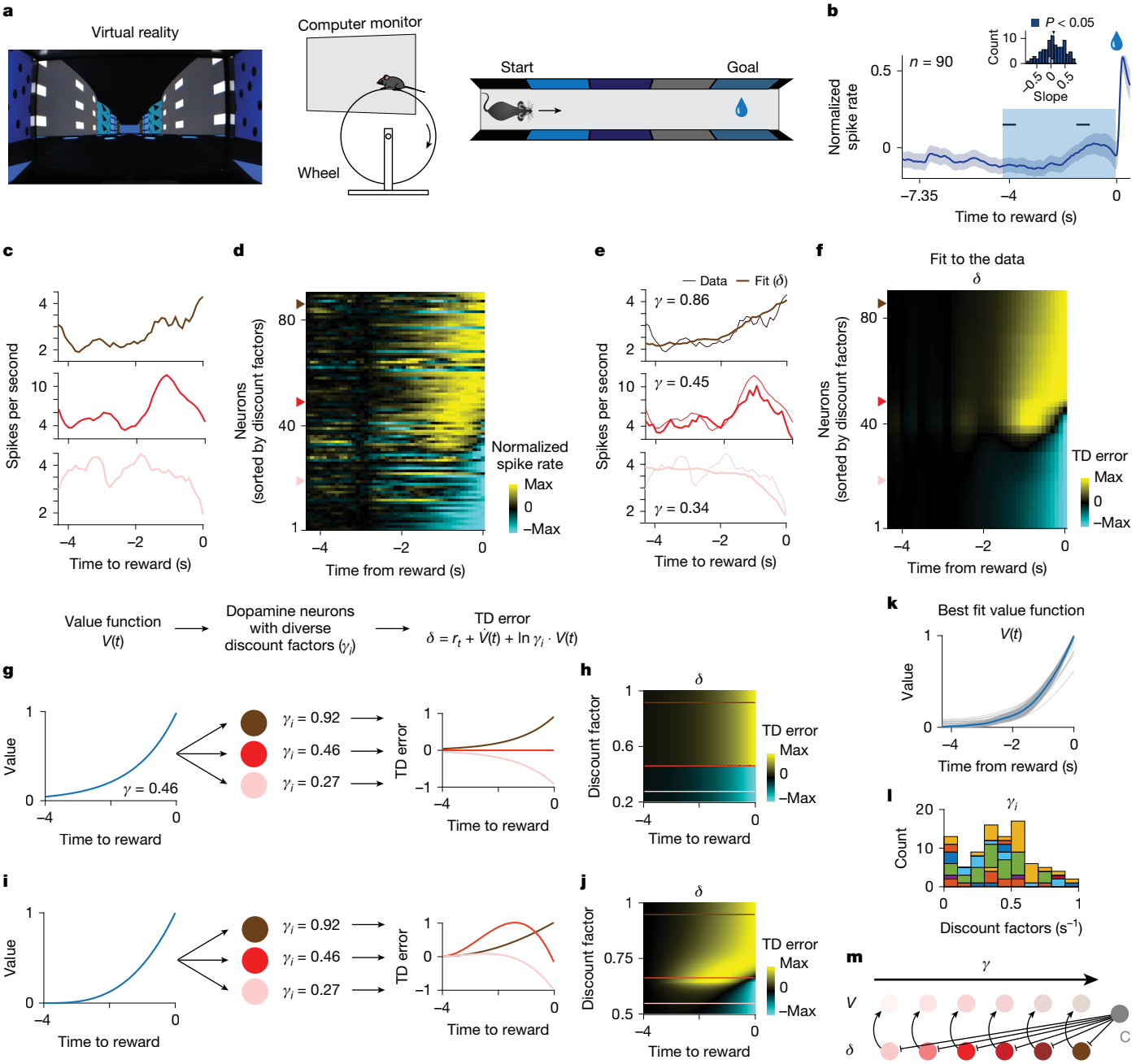


Fig. 3 | The diversity of discount factors across dopaminergic neurons explains qualitatively different ramping activity. a, Experimental setup. View of the virtual reality corridor at movement initiation (left). Schematics of the experimental setup (middle and right). The mouse image in the diagram of the experimental setup in the right panel was created by Gil Costa under a Creative Commons licence CC BY 4.0. **b**, Average activity of single dopaminergic neurons ($n = 90$) exhibits an upwards ramp in the last few seconds of the track before reward delivery. The error bars represent s.e.m. across neurons. The inset shows that the slope of the activity ramp (computed between the two black horizontal ticks in main panel) is positive on average but varies across neurons (for population, mean slope = 0.097; $P = 0.0175$. For single neurons, positive and $P < 0.05$: $n = 53$; negative and $P < 0.05$: $n = 32$; $P > 0.05$: $n = 5$, two-tailed Student's t-test). Image of a water droplet in panels **a**,**b** was created by googlefonts via SVG Repo under an Apache Licence. **c**, Example single neurons showing diverse ramping activity in the final approach to reward, including monotonic upwards (dark red), non-monotonic (red) and monotonic downwards (light red) ramps. **d**, Individual neurons across the population exhibit a spectrum of diversity in their ramping activity. Neurons are sorted according to the inferred discount factor from the common value function model (panel **k**). **e**, Example model fits for the single neurons shown in panel **c**. **f**, The model captures the diversity of ramping activity across the population. Neurons are ordered by the inferred discount factor as in panel **d**. **g**,**h**, Diversity of ramping as a function of discount factor for an exponential value function. **i**,**j**, Diversity of ramping as a function of discount factor for a cubic value function. **k**, Inferred value function. The thin grey lines denote the inferred value function for each bootstrap. The thick blue line indicates the mean over bootstraps. **l**, Histogram of inferred discount factors. Mean ± s.d. of 0.42 ± 0.23. **m**, Model in which the value function used for the RPE computation is shared across dopaminergic neurons (see Methods and Extended Data Fig. 7f–k).

mechanisms and the reduction in uncertainty due to visual feedback[45,46], we first assumed that heterogeneous dopaminergic neurons contribute to learning a common model of the value of the states in the environment and therefore share a common value function (Fig. 3m; see Methods). Depending on the shape of this value function, governed by the statistics of the environment being learned, the TD error from

neurons with different discount factors will exhibit different types of activity ramps. At a given time, the sign of the TD error will depend on the relative scale of the upcoming increase in value and the reduction of this future value due to discounting. Given an increase in value $1/\gamma_o$ (with $\gamma_o < 1$), a neuron with a discount factor smaller, equal or larger than γ_o will experience a negative, zero or positive TD error, respectively (Extended Data Fig. 7a; see Methods). For an exponential value function (Fig. 3g, left panel), in which the value increases by a fixed factor $\frac{1}{\gamma_o}$ at every timestep, a neuron with discount factor γ_o will have no TD error during the entire visual scene (red line, Fig. 3g,h). A neuron with a higher (or lower) discount factor than γ_o will experience an upwards (or downwards) monotonic ramp in its activity (darker and lighter red line in Fig. 3g,h, respectively). However, if the value function is non-exponential (for example, cubic as a function of distance to reward (Fig. 3i, left panel) or hyperbolic as a function of distance to reward (Extended Data Fig. 7b, left panel)), there will not be a neuron whose discount factor is able to match the increases in value function at all timesteps. Neurons with high or low discount factors will still ramp upwards or downwards (darker and lighter red line in Fig. 3i,j and Extended Data Fig. 7b, respectively), but neurons with intermediate discount factors will exhibit non-monotonic ramping (red line, Fig. 3i,j and Extended Data Fig. 7b) as observed in the neural data.

To fit this model to the dopaminergic neurons, we used a bootstrapped constrained optimization procedure on a continuous formulation of the TD error[45,47] ($\delta_i(t) = b_i + \alpha_i(\gamma_i^{\mathrm{d}t}\mathrm{d}V(t)/\mathrm{d}t - \gamma_i^{\mathrm{d}t}\ln(\gamma_i)V(t))$; see Methods) by fitting a non-parametric common value function and neuron-specific gains, baselines and discount factors. Although the gain and baseline activity scale the range of activity, only the interaction between the value function and the discount factor affects the shape of the TD error across time (see Methods). The heterogeneity of ramping activity across single neurons is explained (Fig. 3e,f) by a common convex value function (Fig. 3k) and a diversity of discount factors across single neurons (Fig. 3l). We did not observe a significant correlation neither between inferred parameters and the mediolateral position of the implanted electrodes (Extended Data Fig. 7c–e; although we did not sample extensively lateral positions) nor with licking behaviour before reward delivery (a measure of behavioural discounting; Extended Data Fig. 8a–d; see Methods). Furthermore, the model fit was robust when applied at the single-animal level for the two animals with sufficient numbers of neurons (Extended Data Fig. 8e–j; see Methods). So far, we proposed a descriptive model with a common value function across neurons, suggesting that the prediction errors of single neurons are pooled to estimate a single value function. Recent models for distributed prediction errors across dopaminergic neurons have instead used parallel loops in which individual neurons contribute to estimating separate value functions[20,21,23,48–50]. We obtained similar results in such a model in which neurons estimate separate value functions and instead share a common expectation of reward timing (see Methods; Extended Data Fig. 9). We can reconcile these two models as being two edge cases of a model in which, across independent value estimators, there is a relative amount of mixing between independent estimates and a common value signal (see the section 'Mixing in distributed RL models' in Methods; Extended Data Fig. 7f–k).

Together, these results show that diversity in slow changes in activity across a single neuron (known as dopamine ramps) in environments with gradual changes in value can be explained by a diversity of discount factors and is a signature of multi-timescale RL.

Correlated discount factors across tasks

Distributional RL and other distributed RL formulations provide agents with greater flexibility as they allow agents to adapt risk sensitivity and discounting to the statistics of the environment[17,19,21,23]. However, they leave open the question of the biological implementation of this adaptivity. Specifically, the tuning of single dopaminergic neurons, controlled by the sensitivity to reward size or the discount factor, could be either a circuit property and therefore task and context specific or a cell-specific property, with the contribution of different neurons recruited according to task demands. However, measurements of tuning diversity at the single-neuron level are usually done in a single behavioural task[20,51,52], leaving open the question of this implementation across contexts.

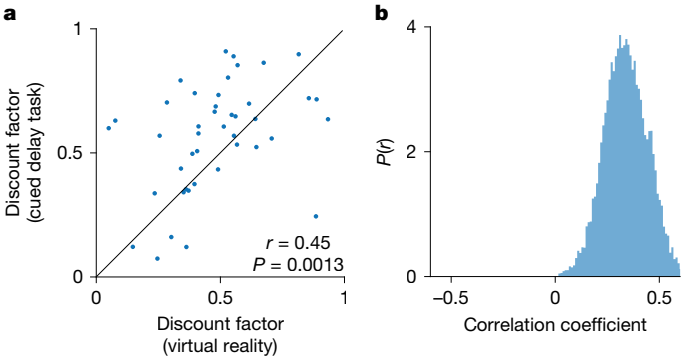


Fig. 4 | Discount factors of single dopaminergic neurons are correlated across behavioural contexts. a, Correlation across neurons between the discount factors inferred in the virtual reality task and the discount factors inferred in the cued delay task (Spearman's rank correlation, one-tailed Student's *t*-test). **b**, Distribution of Spearman's rank correlations between the discount factors across the two tasks for randomly sampled pairs of bootstrap estimates (0.34 ± 0.104, mean ± s.d.; $P < 1.0 \times 10^{-30}$, one-tailed Student's *t*-test).

Here we characterized discount factors across two behavioural tasks, and a subset ($n = 43$) of the single neurons analysed above (Figs. 2 and 3) was recorded on the same day in both behavioural tasks. Using this dataset, we found that the discount factors inferred independently across the two behavioural tasks are correlated (Fig. 4a,b). Furthermore, in the cued delay task, we were able to decode subjective reward timing from population cue responses using the discount matrix built from the discount factors inferred in the virtual reality task ($P_{t=0.6\,\mathrm{s}} = 1$, $P_{t=1.5\,\mathrm{s}} < 1.1 \times 10^{-20}$, $P_{t=3.75\,\mathrm{s}} < 3.8 \times 10^{-20}$ and $P_{t=9.375\,\mathrm{s}} < 2.9 \times 10^{-5}$, compared with shuffled data; Extended Data Fig. 10a–d; see Methods). These results suggest that the discount factor (or its ranking) is a cell-specific property and strongly constrains the biological implementation of multi-timescale RL in the brain.

Discussion

In this work, we have analysed the unique computational benefits of multi-timescale RL agents and shown that we can explain multiple aspects of the activity of dopaminergic neurons through that lens.

The understanding of dopaminergic neurons as computing a reward prediction error from TD RL algorithms has transformed our understanding of their function. However, recent experimental work expanding the anatomical locations of recordings and the task designs has shown heterogeneity in dopamine responses that is not readily explained within the canonical TD framework[42,53,54]. However, a number of these seemingly anomalous findings can be reconciled and integrated within extensions of the RL framework, further reinforcing the power and versatility of the TD theory in capturing the intricacies of brain learning mechanisms[23,24,48,55,56]. In this work, we have revealed an additional source of heterogeneity across dopaminergic neurons: they encode prediction errors across multiple timescales. Together, these results indicate that at least some of the heterogeneity observed in dopamine responses reflects variations in key parameters within the RL framework. Thus, these results indicate that the dopamine system uses 'parameterized vector prediction errors', including a discrete Laplace

transform of the future temporal evolution of the reward function, allowing for the learning and representation of richer information than what can be achieved with scalar prediction errors in the traditional RL framework.

The constraint on the anatomical implementation of multi-timescale RL suggested by the alignment of discount factors between the two tasks could also inform algorithm design. Adapting the discount factor has been used to improve performance in several algorithms, with proposed methods ranging from meta-learning an optimal discount factor[57], learning state-dependent discount factors[58] or combining parallel exponentially discounting agents[19,33,36]. Our results provide evidence supporting the third model, but the recruitment mechanisms of the neurons to adapt the global discounting function with task or context and the link between anatomical location and discounting[30] and the contribution of other neuromodulators, such as serotonin[59,60], to this adaptation remain open questions. Similarly, the contribution of this vectorized error signal on the downstream temporal representations[26,28] remains to be explored.

Understanding how this recruitment occurs will be a key step towards a mechanistic understanding of the contribution of this timescale diversity to calibration and miscalibration in intertemporal choices. There has been a conundrum that RL theories use exponential discounting, whereas humans and animals often exhibit hyperbolic discounting. A previous study, which examined discounting in dopaminergic neurons, has argued that single dopaminergic neurons exhibit hyperbolic discounting[39]. However, they used uncued reward responses for zero reward delay, probably biasing the estimate towards hyperbolic (as responses to unpredicted rewards are typically large and potentially contaminated by salience signals). By contrast, our data are consistent with exponential discounting at the level of single neurons, suggesting that RL machinery defined by each dopaminergic neuron conforms to the rules of a simple RL algorithm. Hyperbolic-like discounting can occur when these diverse exponential discounting are combined at the organism level[12,14,33]. More generally, the relative contribution of multiple timescales to the global computation governs the discount function at the organism level and should be calibrated to the uncertainty in the hazard rate of the environment[12].

Appropriately recruiting the heterogeneity of discount factors is therefore important to adapt to the temporal uncertainty of the environment. This view draws parallels with the distributional RL hypothesis that naturally fits with current work on anhedonia, as a miscalibration of optimism and pessimism can lead to biases in the learned value[20]. Miscalibration of the discounting spectrum can lead to excessive patience or impulsivity. A bias in this distribution due to genetical, developmental or transcriptional factors could bias the learning at the level of the organism towards short-term or long-term goals. Behaviourally such bias would manifest itself as an apparent impulsivity or lack of motivation, leading to a potential mechanistic interpretation of these maladaptive behaviours. Similarly, this view could guide the design of algorithms that recruit and leverage these adaptive temporal predictions.

Our study has established a new paradigm to understand the functional role of prediction error computation in dopaminergic neurons, and opens new avenues to develop mechanistic explanations for deficits in intertemporal choice in disease and inspire the design of new algorithms.

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/s41586-025-08929-9.

1. Sutton, R. S. & Barto, A. G. *Reinforcement Learning* 2nd edn (MIT Press, 2018).
2. Tesauro, G. Temporal difference learning and TD-Gammon. *Commun. ACM* **38**, 58–68 (1995).
3. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
4. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
5. Wurman, P. R. et al. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature* **602**, 223–228 (2022).
6. Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. *Science* **275**, 1593–1599 (1997).
7. Schultz, W. Neuronal reward and decision signals: from theories to data. *Physiol. Rev.* **95**, 853–951 (2015).
8. 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).
9. Ainslie, G. Specious reward: a behavioral theory of impulsiveness and impulse control. *Psychol. Bull.* **82**, 463–496 (1975).
10. Frederick, S., Loewenstein, G. & O'Donoghue, T. Time discounting and time preference: a critical review. *J. Econ. Lit.* **40**, 351–401 (2002).
11. Laibson, D. Golden eggs and hyperbolic discounting. *Q. J. Econ.* **112**, 443–478 (1997).
12. Sozou, P. D. On hyperbolic discounting and uncertain hazard rates. *Proc. R. Soc. London. B* **265**, 2015–2020 (1998).
13. Botvinick, M. et al. Reinforcement learning, fast and slow. *Trends Cogn. Sci.* **23**, 408–422 (2019).
14. Redish, A. D. Addiction as a computational process gone awry. *Science* **306**, 1944–1947 (2004).
15. Lempert, K. M., Steinglass, J. E., Pinto, A., Kable, J. W. & Simpson, H. B. Can delay discounting deliver on the promise of RDoC? *Psychol. Med.* **49**, 190–199 (2019).
16. Sutton, R. S. et al. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems* Vol. 2 761–768 (International Foundation for Autonomous Agents and Multiagent Systems, 2011); https://dl.acm.org/doi/10.5555/2031678.2031726
17. Bellemare, M. G., Dabney, W. & Rowland, M. *Distributional Reinforcement Learning* (MIT Press, 2023).
18. Jaderberg, M. et al. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations* (ICLR, 2017).
19. Fedus, W., Gelada, C., Bengio, Y., Bellemare, M. G. & Larochelle, H. Hyperbolic discounting and learning over multiple horizons. Preprint at https://arxiv.org/abs/1902.06865 (2019).
20. Dabney, W. et al. A distributional code for value in dopamine-based reinforcement learning. *Nature* **577**, 671–675 (2020).
21. Tano, P., Dayan, P. & Pouget, A. A local temporal difference code for distributional reinforcement learning. *Adv. Neural Inf. Process. Syst.* **1146**, 13662–13673 (2020).
22. Brunec, I. K. & Momennejad, I. Predictive representations in hippocampal and prefrontal hierarchies. *J. Neurosci.* **42**, 299–312 (2022).
23. Lowet, A. S., Zheng, Q., Matias, S., Drugowitsch, J. & Uchida, N. Distributional reinforcement learning in the brain. *Trends Neurosci.* **43**, 980–997 (2020).
24. Masset, P. & Gershman, S. J. in *The Handbook of Dopamine (Handbook of Behavioral Neuroscience)* Vol. 32 (eds Cragg, S. J. & Walton, M.) Ch. 24 (Academic Press, 2025).
25. Buhusi, C. V. & Meck, W. H. What makes us tick? Functional and neural mechanisms of interval timing. *Nat. Rev. Neurosci.* **6**, 755–765 (2005).
26. Tsao, A., Yousefzadeh, S. A., Meck, W. H., Moser, M.-B. & Moser, E. I. The neural bases for timing of durations. *Nat. Rev. Neurosci.* **23**, 646–665 (2022).
27. Fiorillo, C. D., Newsome, W. T. & Schultz, W. The temporal precision of reward prediction in dopamine neurons. *Nat. Neurosci.* **11**, 966–973 (2008).
28. Mello, G. B. M., Soares, S. & Paton, J. J. A scalable population code for time in the striatum. *Curr. Biol.* **25**, 1113–1122 (2015).
29. Soares, S., Atallah, B. V. & Paton, J. J. Midbrain dopamine neurons control judgment of time. *Science* **354**, 1273–1277 (2016).
30. Enomoto, K., Matsumoto, N., Inokawa, H., Kimura, M. & Yamada, H. Topographic distinction in long-term value signals between presumed dopamine neurons and presumed striatal projection neurons in behaving monkeys. *Sci. Rep.* **10**, 8912 (2020).
31. 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).
32. Kiebel, S. J., Daunizeau, J. & Friston, K. J. A hierarchy of time-scales and the brain. *PLoS Comput. Biol.* **4**, e1000209 (2008).
33. Kurth-Nelson, Z. & Redish, A. D. Temporal-difference reinforcement learning with distributed representations. *PLoS ONE* **4**, 7362 (2009).
34. Shankar, K. H. & Howard, M. W. A scale-invariant internal representation of time. *Neural Comput.* **24**, 134–193 (2012).
35. Tanaka, C. S. et al. Prediction of immediate and future rewards differentially recruits cortico-basal ganglia loops. *Nat. Neurosci.* **7**, 887–893 (2004).
36. Sherstan, C., Dohare, S., MacGlashan, J., Günther, J. & Pilarski, P. M. Gamma-Nets: generalizing value estimation over timescale. In *Proceedings of the AAAI Conference on Artificial Intelligence* **34**, 5717–5725 (2020).
37. Momennejad, I. & Howard, M. W. Predicting the future with multi-scale successor representations. Preprint at *bioRxiv* https://doi.org/10.1101/449470 (2018).
38. Reinke, C., Uchibe, E. & Doya, K. Average reward optimization with multiple discounting reinforcement learners. In *Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science* (eds Liu, D. et al.) 789–800 (Springer, 2017).
39. Kobayashi, S. & Schultz, W. Influence of reward delays on responses of dopamine neurons. *J. Neurosci.* **28**, 7837–7846 (2008).
40. Schultz, W. Dopamine reward prediction-error signalling: a two-component response. *Nat. Rev. Neurosci.* **17**, 183–195 (2016).
41. Howe, M. W., Tierney, P. L., Sandberg, S. G., Phillips, P. E. M. & Graybiel, A. M. Prolonged dopamine signalling in striatum signals proximity and value of distant rewards. *Nature* **500**, 575–579 (2013).
42. Berke, J. D. What does dopamine mean? *Nat. Neurosci.* **21**, 787–793 (2018).

43. Gershman, S. J. Dopamine ramps are a consequence of reward prediction errors. *Neural Comput.* **26**, 467–471 (2014).
44. Kim, H. G. R. et al. A unified framework for dopamine signals across timescales. *Cell* **183**, 1600–1616.e25 (2020).
45. 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).
46. Guru, A. et al. Ramping activity in midbrain dopamine neurons signifies the use of a cognitive map. Preprint at *bioRxiv* https://doi.org/10.1101/2020.05.21.108886 (2020).
47. Doya, K. Reinforcement learning in continuous time and space. *Neural Comput.* **12**, 219–245 (2000).
48. Lee, R. S., Sagiv, Y., Engelhard, B., Witten, I. B. & Daw, N. D. A feature-specific prediction error model explains dopaminergic heterogeneity. *Nat. Neurosci.* **27**, 1574–1586 (2024).
49. Cruz, B. F. et al. Action suppression reveals opponent parallel control via striatal circuits. *Nature* **607**, 521–526 (2022).
50. Millidge, B., Song, Y., Lak, A., Walton, M. E. & Bogacz, R. Reward bases: a simple mechanism for adaptive acquisition of multiple reward types. *PLoS Comput. Biol.* **20**, e1012580 (2024).
51. Engelhard, B. et al. Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons. *Nature* **570**, 509–513 (2019).
52. Eshel, N., Tian, J., Bukwich, M. & Uchida, N. Dopamine neurons share common response function for reward prediction error. *Nat. Neurosci.* **19**, 479–486 (2016).
53. Cox, J. & Witten, I. B. Striatal circuits for reward learning and decision-making. *Nat. Rev. Neurosci.* **20**, 482–494 (2019).
54. Collins, A. L. & Saunders, B. T. Heterogeneity in striatal dopamine circuits: form and function in dynamic reward seeking. *J. Neurosci. Res.* https://doi.org/10.1002/jnr.24587 (2020).
55. Gershman, S. J. et al. Explaining dopamine through prediction errors and beyond. *Nat. Neurosci.* **27**, 1645–1655 (2024).
56. Watabe-Uchida, M. & Uchida, N. Multiple dopamine systems: weal and woe of dopamine. *Cold Spring Harb. Symp. Quant. Biol.* **83**, 83–95 (2018).
57. Xu, Z., van Hasselt, H. P. & Silver, D. Meta-gradient reinforcement learning. In *Advances in Neural Information Processing Systems* Vol. 31 (Curran Associates, 2018).
58. Yoshida, N., Uchibe, E. & Doya, K. Reinforcement learning with state-dependent discount factor. In *2013 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)* https://ieeexplore.ieee.org/document/6652533 (IEEE, 2013).
59. Doya, K. Metalearning and neuromodulation. *Neural Netw.* **15**, 495–506 (2002).
60. Tanaka, S. C. et al. Serotonin differentially regulates short- and long-term prediction of rewards in the ventral and dorsal striatum. *PLoS ONE* https://doi.org/10.1371/journal.pone.0001333 (2007).

Methods

Animal care and surgical procedures

The mouse behavioural and electrophysiological data presented here were collected as part of a previous study in which all experimental procedures are described in detail[44]. As described in this study, all procedures were performed in accordance with the US National Institutes of Health Guide for the Care and Use of Laboratory Animals and approved by the Harvard Animal Care and Use Committee.

We used a total of 13 adult C57BL6/J DAT-Cre male mice. Mice were backcrossed for over five generations with C57BL6/J mice, Animals were singly housed after surgery on a reverse 12-h dark–12-h light cycle (dark from 7:00 to 19:00). Single dopaminergic neurons were optogenetically identified using custom-built micro-drives with eight tetrodes and an optical fibre as described in our previous study[44]. Significance was assessed using the stimulus-associated spike latency test[61].

All mice ($n = 13$) were used in the virtual reality task and 8 of those were also used in the cued delay task. The targeted mediolateral location varied from 320 μm to 1,048 μm for neurons recorded in the virtuality task and for neurons recorded in the cued delay task. Neurons recorded at mediolateral position of more than 900 μm were excluded from the analysis as they were considered to be in the substantia nigra pars compacta. For experimental reasons, experimenters were not blinded to the identity of the mice. Sample size was maximized given experimental constrains.

RL at multi-timescales

In standard RL, the value of a state s under a given policy π is defined as the expected sum of discounted future rewards:

$$V(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | s, \pi\right] \tag{5}$$

The discount factor γ (whose value is between 0 and 1) is a fixed factor at each timestep devaluating future rewards. This exponentially functional form for the temporal discount is not arbitrary. This temporal discount is naturally produced by the TD learning rule, a bootstrapping mechanism that updates the value estimates using the experienced transition from s to s' with reward r:

$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)] \tag{6}$$

where α is the learning rate. This update process converges to the values defined above under very general conditions[62].

After convergence, the value $V(s)$ can be rewritten by taking the sum and the discount factor outside of the expectation:

$$V_\gamma(s) = \sum_{t=0}^{\infty} \gamma^t E[r_t | s] \tag{7}$$

Where we have added a γ subscript to the value to indicate that the value is computed for that particular discount, and we have omitted the dependence of the expectation on π for simplicity. This last expression reveals a very useful property: $V_\gamma(s)$, as a function of the discount $\gamma \in (0,1)$, is the unilateral z-transform of $E[r_t|s]$ as a function of future time $t \in (0, \infty)$, of with real-valued parameter γ^{-1} (that is, the discrete-time equivalent of the Laplace transform[63]). As the z-transform is invertible, in the limit of computing values with an infinite amount of γ, the agent can recover the expected rewards at all future times $\{E[r_t|s]\}_{t=0}^{\infty}$ from the set of learned values $\{V_\gamma(s)\}_{\gamma \in (0,1)}$:

$$Z^{-1}\{V_\gamma(s)\}_{\gamma \in (0,1)} = \{E[r_t|s]\}_{t=0}^{\infty} \tag{8}$$

Thus, if the agent performs TD learning with an infinite amount of discounts, the converging points of the TD backups would encode not only the expected sum of discounted rewards, as in traditional RL, but also the expected reward at all future timesteps, although the latter lies in a different space, analogous to the frequency and temporal spaces of the Fourier transform.

Decoding tasks

The four tasks in Fig. 1e were designed with a similar structure. In the four tasks, the agent first performs N backups of tabular TD learning (equation (4) in the previous section) on the experimental states (Fig. 1c). Then, the learned values for the cue s are input into a policy gradient network with one hidden layer of 32 units, and a ReLU non-lineary (Fig. 1d, step 2). The policy gradient network receives in its input the values learned by TD learning and reports in its output the corresponding estimate for each task. The policy gradient network was trained across 2,000 episodes, after which we evaluated the accuracy of its report.

The precise structure of each episode depends on the task (see details below). In general, in each episode, the agent learned values from scratch using TD learning for a specific experimental condition (that is, a Markov decision process (MDP)), and the policy gradient network maximized its reporting performance across episodes. Thus, for each episode i, the policy (π_θ) was a map from the learned multi-timescale values (V_γ^i) to actions (a_i). The parameters (θ) of the policy gradient network were optimized to maximize reporting accuracy across episodes (the specific measure to report depends on the experimental condition). The parameters were learned by optimizing the traditional policy gradient loss, using an Adam optimizer with a learning rate of 0.001 to maximize the task-specific expected return $J(\pi_\theta)$ of the policy π_θ:

$$\nabla_\theta J(\pi_\theta) = E_{B \sim \pi_\theta}\left[\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(a_i \mid V_\gamma^i) C_i\right] \tag{9}$$

where B is a batch of $n = 100$ episodes and C_i is a RL binary signal indicating whether the report (a_i; the output of the network) was correct or incorrect for episode i, given the learned multi-timescale values V_γ^i. To tackle the exploration–exploitation problem, we extended the policy using ε-greedy, with $\varepsilon = 0.3$ (performance is reported with $\varepsilon = 0$). We used a decoder trained with RL methods instead of supervised learning as it does not require an oracle that knows the correct responses, and is therefore a more realistic model of biological learning.

In task 1 (Fig. 1e,f and Extended Data Fig. 1a–c), in each episode, a discrete reward time t_R is sampled between 1 and 15 and a discrete reward magnitude R sampled between 1 and 10. This defines a MDP shown in Extended Data Fig. 1a. For this MDP, TD learning was used to learn the value of the first state of the MDP s, which we refer to as the 'cue'. In all tasks, the value of the cue was learned using one, two or three discount factors (γ) from the set {0.6,0.9,0.99}, depending on the experimental condition. The results indicated as 'three discounts' corresponds to the discount factors [0.6,0.9,0.99]. As there is noise in the simulation (see below), the results indicated as 'one discount' corresponds to the top performer over three identical discount factors ([0.6,0.6,0.6], [0.9,0.9,0.9], [0.99,0.99,0.99]) and analogously for the results indicated as 'two discounts'. After performing TD learning, the values were fed as input into the policy gradient network whose output was the guessed reward time (the network has 15 discrete actions, corresponding to reporting reward times from 1 to 15). Performance was evaluated as the fraction of correct responses across test episodes (1 for estimating the correct reward time, 0 otherwise). We have shown the performance of the policy gradient network as it is trained in Extended Data Fig. 1c. In Extended Data Fig. 1m–o, we have shown a similar experiment but using two reward times and reward magnitudes in the MDP. In this complex case, a more accurate decoding is obtained using five discounts instead of three.

In task 2 (Fig. 1e,f and Extended Data Fig. 1d–f), the structure of each episode was as in task 1 but with a discrete reward time t_R sampled between 1 and 8 and a discrete reward magnitude R sampled between

1 and 4. The learned values were input into a policy gradient network with 32 possible discrete outputs, representing the 32 possible hyperbolic values obtained in all the possible experiment (4 possible reward magnitudes × 8 possible reward times):

$$V(s) = \frac{R}{1 + 0.9t_R} \quad (10)$$

Performance was evaluated as the fraction of correct responses across episodes.

In task 3 (Fig. 1e,f and Extended Data Fig. 1g–i), we used the MDP shown in Extended Data Fig. 1g while keeping R fixed at 1 but varying t_R and the number of times (N) that the full MDP had been experienced by the agents. As TD backups were performed online after every transition, N is proportional to the total number of TD backups. The (possibly incomplete) learned values at s from these N experiences were fed into the policy gradient network (Extended Data Fig. 1h), which was trained across episodes to optimize the reporting performance of t_R.

We also evaluated learning in incomplete-information situations using the MDP shown in Extended Data Fig. 1p–r. In each episode, the length of the two branches was uniformly sampled from 5 to 15 (if they are the same, they were resampled until being different). Thus, in each episode, there is a shorter branch and a longer branch. Each branch was experienced a random number of times (N) sampled from a uniform distribution with the range of 1–99 (denoted by uniform(1,99)). Thus, the number of TD backups performed for the two branches could be highly asymmetric. The learned values (with one or multiple discounts) were fed as input into the policy gradient network with a binary output indicating which path was the shortest one; performance was evaluated as the fraction of correct responses (Extended Data Fig. 1o). Single-timescale agents can incorrectly believe that one branch is shorter than the other branch if it has been experienced more often, but multi-timescale agents can determine the distance to the reward independently of the asymmetric experience. In the next section, we present a theoretical proof showing that at any time during TD learning (that is, before learning converges), multi-timescale systems can perform the z-transform and decode the timing of non-zero rewards (in the absence of timing stochasticity). In addition to the theoretical proof, we present an intuitive explanation (supported by Extended Data Fig. 1s–w).

In task 4 (Fig. 1e,f and Extended Data Fig. 1j–l), we keep the reward R fixed at 1 and t_R varies between 1 and 4. Crucially, small random rewards sampled from normal(0,0.25) were added to every state (fixed within episodes). If the agent experienced the trajectory an infinite number of times, the noisy rewards would be averaged out, so they would not affect the value estimates of the cue. We note these 'true' value estimates as $V_\gamma^{\text{true}}(s)$, to distinguish them from $V_\gamma(s)$, which are the values learned with (incomplete) TD learning. In task 4, the agent experienced the trajectory only once (that is, a single backup of TD learning along the trajectory), so the small random rewards do affect the values $V_\gamma(s)$ learned with TD learning. These noisy values are input into the policy gradient network, the goal of which is to report the true value of the cue $V_{\gamma=0.9}^{\text{true}}(s)$, with a discount of 0.9, that would arise after experiencing the trajectory of an infinite number of times (this is, ignoring the noisy rewards). Although in task 4 we have illustrated the advantage of the myopic learning bias in a task in which the uncertainty on the value estimates arises due to stochastic rewards received at every state, the myopic bias is beneficial independently of the origin of the uncertainty. For example, in more realistic state spaces, uncertainty usually arises due to incomplete exploration of the state space. In Extended Data Fig. 2a, we illustrate that myopic estimates are generally more accurate when the near future is more certain than the far future, and far-sighted estimates are more accurate when the far future is more certain than the near future. We have shown the benefits of the myopic learning bias in more-realistic scenarios in which uncertainty arises due to noise in a branching task, as well as due to incomplete exploration of the state space in a grid-world, and in a deep RL environment (see Methods below; Extended Data Fig. 2).

In tasks 1–4, the TD-learning process was corrupted by noise. In each episode, the learning rate was sampled from a normal distribution with mean of 0.1 and variance of 0.001 (denoted by $N(0.1, 0.001)$), the number of TD backups was sampled from uniform(59,99) (except in the tasks with incomplete learning: tasks 3 and 4). This variability was included to make sure that the decoder learns robust decoding strategies instead of just memorizing the exact values of each experimental condition. For example, in task 1, with one discount, the value of a temporally close small reward was similar to the value of a temporally far high reward, so reward time cannot be disentangled from reward magnitude. However, although these two values were similar, they were not identical, so a decoder with enough precision could learn to memorize them to report reward time. Introducing a small amount of random noise in the learning process assures robustness in the evaluation of the reporting performance.

Finally, note that we used a non-linear decoder for the policy gradient network instead of a linear one. As we showed in the previous section, in principle, reward time can be decoded from value estimates with the L^{-1} linear decoder. We used a non-linear decoder for two reasons. First, optimal performance in tasks 2 and 4 requires non-linear operations over the learned values. Task 2 requires computing the hyperbolic value, and task 4 requires biasing towards myopic or far-sighted estimates based on the estimated reward time. Second, even in tasks in which the goal is only to report reward time (for example, tasks 1 and 3), the linear decoder L^{-1} is only guaranteed to work well in optimal learning conditions (unnoisy value estimates with an infinite number of discounts). In incomplete learning conditions, the L^{-1} decoder has been shown to be very sensible to noise[21], which leads to poor performance in the tasks studied here. In general, our goal in these simulations was to illustrate the power of the multi-timescale representations over the single-timescale representations, as recoverable by a simple non-linear decoder.

Recovering temporal information before TD learning converges

In Extended Data Fig. 1s–w, we illustrate intuitively why the temporal information is available before TD learning converges for multi-timescale agents (experiment in Fig. 1e). Consider the two experiments in Extended Data Fig. 1s, one with a short wait between the cue and reward (pink) and one with a longer wait (cyan). For a single-timescale agent (Extended Data Fig. 1t), the value of the cue depends not only on the experiment length but also on the number of times that each experiment has been experienced (N, the number of TD backups). Thus, for a given set of learning parameters (learning rate, discount factor, timestep length and reward magnitude), the single-timescale agent can incorrectly believe that the cyan cue indicates the shorter trajectory, if it has been experienced more often (left part of the plot). However, as we show theoretically in this section, as temporal information is encoded across discount factors for a multi-timescale agent, multi-timescale agents can determine reward timing independently of N. In Extended Data Fig. 1u, the patterns of three dots highlighted with rectangles are indicative of the reward time and are only affected by the learning parameters by a multiplicative factor. Indeed, when we plot the multi-timescale values as a function of the number of times that the experiments are experienced (N; Extended Data Fig. 1v–w), we saw that the pattern across discounts is maintained, enabling a downstream system to robustly decode reward timing.

The following is a theoretical proof of this advantage. Consider a multi-timescale agent performing TD learning on the trajectory $s \rightarrow \cdots \rightarrow s_T$ in which there is no variability in outcome timing (that is, non-zero outcomes always happen at the same states, but their magnitude can be stochastic) and all rewards are positive. Under these

assumptions, the agent is able to decode reward timing if it has access to $\{\delta_{(r_\tau,0)}\}_{\tau=0}^{T}$, the future times at which outcomes r_τ are non-zero given the current state, where $\delta_{(r_\tau,0)}$ is a Kronecker delta function that is equal to 1 if r_τ is zero and equal to 0 otherwise. At any time during TD learning, the value estimate for s computed with TD learning can be written with the following general expression (note the absence of the expectation):

$$V_\gamma(s) = \sum_{\tau=0}^{T} \gamma^\tau f_\tau(\alpha, N, R_{0:\tau})(1-\delta_{(r_\tau,0)}) \tag{11}$$

where $f_\tau(\alpha, N, R_{0:\tau})$ is a non-zero scalar that depends on τ, on the learning rate α, on the number of times the trajectory has been experienced N and on the history of outcome magnitudes experienced in the past $R_{0:\tau}$. This decoupling shares similarity with the successor representation[37,64,65]. Crucially, $f_\tau(\alpha, N, R_{0:\tau})$ does not depend on γ, so, at all times during learning, it holds that:

$$Z^{-1}\{V_\gamma(s)\}_{\gamma\in(0,1)} = \{f_\tau(\alpha, N, R_{0:\tau})(1-\delta_{(r_\tau,0)})\}_{\tau=0}^{T} \tag{12}$$

As $f_\tau(\alpha, N, R_{0:\tau})$ is non-zero for all τ and $\{1-\delta_{(r_\tau,0)}\}_{\tau=0}^{T}$ is only non-zero at τ in which a reward happens, the non-zero values of the right-hand side expression indicates the future reward timings. In other words, applying the inverse transform at any time during learning to the multi-timescale estimate $\{V_\gamma(s)\}_{\gamma\in(0,1)}$ gives an expression whose non-zero values are the future outcome timings. In summary, in the absence of timing stochasticity, the multi-timescale agent can recover future outcome timing before TD converges, a capability that is not present in single-timescale agents.

The myopic learning bias

For the value estimate of a state s to converge to the expression shown in equation (1), learning needs to be 'complete', which requires that (1) all possible paths from s are explored; and (2) if there are stochastic rewards or transitions, all possible paths are explored a sufficiently large number of times such that the stochasticity is averaged out. Although these two conditions are usually true in artificial laboratory experiments, they are rarely true in natural environments. When these two conditions are not met, value estimates must be computed on the basis of incomplete, uncertain information. The myopic learning bias states that, when learning from incomplete and uncertain information, the accuracy of myopic versus far-sighted value estimates depends on the uncertainty structure of the future. Myopic estimates are more accurate if the near future is more certain than the far future, and far-sighted estimates are more accurate if the far future is more certain than the near future.

We illustrate the key idea of the myopic learning bias in Extended Data Fig. 2a,b. In Extended Data Fig. 2a, we show two states: s and s'. In both cases, the animal must decide whether to take the upwards or downwards branch. The key difference between s and s' is that, in s, the near future after the decision is more certain than the far future; however, in s', the far future after the decision is more certain than the near future. These states represent frequent situations encountered in natural environments. For example, the two paths that leave from state s represent paths in which the far-away consequences are known with certainty, but there are multiple possible paths (with different and unexplored outcomes) that eventually lead to the more certain states. Conversely, state s' represents a common exploratory situation, in which the branching-tree structure of the MDP causes the number of possible outcomes to increase exponentially with the distance from s'. In incomplete learning scenarios, the structure of future uncertainty from s and s' is opposite. If the agent has not experienced all possible trajectories from s and s', in state s, the near future will be more uncertain than the far future, and in state s', the far future will be more uncertain than the near future.

Consider, for example, what happens if the agent has experienced the upwards and downwards trajectories (from s and s') only once, as we show in Extended Data Fig. 2b. In this case, there are four possible scenarios depending on which specific trajectories the agent visits. As some of the possible paths are left unexplored, the agent must learn from incomplete information, and its value estimates can differ from the those of an agent that has experienced all the paths. A perfect agent that has experienced all trajectories will choose the upwards trajectory from s and s', and therefore the upwards trajectory is the 'correct' decision to make at s and s'. In Extended Data Fig. 2b, we show that, when learning from incomplete information, the probability of making the correct decision by following myopic estimates (low γ) versus far-sighted estimates (high γ) depends on the uncertainty structure of the future. In s, in which the far future is more certain than the near future, it is beneficial to follow far-sighted estimates. In s′, in which the near future is more certain than the far future, it is beneficial to follow myopic estimates. In summary, in s, the myopic value estimates only integrate the certain near future, without being contaminated by the uncertain far future, leading to more accurate value estimation. By contrast, in s', the myopic estimates only see the noisy near future, without being able to improve on this noise by the certain information that happens in the far future, leading to less accurate value estimation.

The myopic learning bias is independent of the source of the uncertainty. In Extended Data Fig. 2a,b, the uncertainty comes due to incomplete state exploration (and also in the grid-world shown in Extended Data Fig. 2f–i; see Methods below). Conversely, in task 4 in Fig. 1 and in the MDP shown in Extended Data Fig. 2c–e, the uncertainty comes due to stochasticity in the rewards. Multi-timescale agents that have myopic and far-sighted estimates at every state can, in principle, leverage this representation advantage to improve performance in specific tasks. For example, in task 4 and Extended Data Fig. 2c–e, the multi-timescale agent can determine the temporal distance to large deterministic rewards (t_R), and adapt accordingly between myopic or far-sighted perspectives, leading to superior performance.

Note that, in general, exploiting the myopic learning bias requires two components: (1) having available myopic and far-sighted value estimates and every state, and (2) knowing if the near future is more certain or uncertain than the far future at every state. The second component can be sometimes inferred by the multi-timescale values (such as in task 4), but this is not necessarily the case in general. For example, in Extended Data Fig. 2a,b, the second component might require the agent to rely on separate uncertainty estimates, such as counting the fraction of paths left unexplored at different moments along the paths. Another useful proxy to estimate the uncertainty structure of the future is the estimated distance to important environmental events (such as the distance to the landing zone in the Lunar Lander environment or to the rewarded zone in the grid world, see below). In summary, only multi-timescale learning systems satisfy component 1, which provides a representational advantage that is absent in single-timescale systems. This representational advantage can be exploited if the uncertainty structure of the future is known. In some scenarios, the uncertainty structure of the future can be inferred by looking at the multi-timescale value array, but it could require separate uncertainty estimates in other scenarios.

Myopic learning bias: branching task. In Extended Data Fig. 2c, we present a simple MDP to highlight the advantages of the myopic learning bias. In this maze, each state is associated with a random reward drawn from [−0.5,0.5], except for two states (s and s'; orange circles), which result in a deterministic reward of 1. The optimal strategy in this scenario is to move upwards at both states s and s', which is the policy that an optimal agent would implement after experiencing the trajectories a sufficiently large number of times, after randomness is averaged out.

However, in our simulation, the agent only learns from three trajectories: (1) a trajectory that moves down at *s*, (2) another that moves up at *s* and up at state *s*′, and (3) a trajectory that moves up at *s* and down at *s*′. As rewards are stochastic, the information that the agent gets on each episode is incomplete. When learning from a limited number of experiences, the smaller stochastic rewards can overpower the larger deterministic rewards, making it challenging to achieve optimal performance. At state *s*, only far-sighted agents can discern the significance of the large deterministic rewards, thereby causing myopic agents to perform near chance at *s* (Extended Data Fig. 2d, red). At state *s*′, the situation is reversed. Far-sighted agents not only integrate the close-by large reward but also all the stochastic rewards farther in the future. Myopic agents, in contrast, assign greater weight to the deterministic reward than to the future stochastic rewards, thus enabling optimal performance at *s*′ (Extended Data Fig. 2d, blue). Therefore, only agents that could dynamically adapt between being far-sighted at *s* and myopic at *s*′ can attain optimal performance when learning from limited experiences (Extended Data Fig. 2e).

To evaluate how well the agent acts given limited information, we averaged performance over the following procedure: (1) sampled rewards along the three trajectories mentioned before, (2) learned the *Q*-values (until convergence) for *s* and *s*′ using the rewards from the sampled trajectories, and (3) chose the actions that maximize the *Q*-values. In Extended Data Fig. 2d, we evaluated performance as the fraction of episodes in which the *Q*-value of the branch with the deterministic reward was higher than the *Q*-value of the branch without the deterministic rewards. Performance was measured as the proportion of correct decisions across 10,000 iterations of this procedure.

To evaluate the multi-timescale agent of Fig. 1d on this task (Extended Data Fig. 2e), we followed a similar procedure. In each episode, we randomized the identity of the top and bottom branches after the bifurcation, which defines an episode-specific MDP. For each episode-specific MDP, the agent performed *Q*-learning until near convergence using the three trajectories mentioned in the previous paragraph. The *Q*-values at the current state (*s* or *s*′) were fed into the policy learning architecture of Fig. 1d, which outputs the decision to move up or down in the episode-specific MDP. The policy-learning network was trained across episodes to produce actions that maximize overall task performance. For the single-discount agent, we have reported the maximum performance over the agents with discounts [0.6,0.6] and [0.99,0.99], which achieved a performance of 77 ± 2% and 83 ± 1%, respectively. For the multi-discount agent, we use the discounts [0.6,0.99], which achieved a performance of 94 ± 1%. The error bars correspond to the s.e.m. across 500 episodes in a validation set.

Myopic learning bias: grid world. Previous theoretical work showed that a myopic discount in RL can serve as a regularizer when approximating the value function from a limited number of trajectories[66]. In Extended Data Fig. 2f–i, we highlight the fact that the benefit of the myopic discount is contingent on the distance between the current state and significant environmental events. Consider the simple navigation scenario depicted in Extended Data Fig. 2f. The motion of the agent is random and isotropic, garnering a minor random reward from a normal distribution with mean 0 and s.d. 0.01 in each step and three more substantial rewards upon reaching the areas denoted by fire ($r = -4$) and water ($r = 2$) symbols. We evaluated how well the agent could determine the true value function (under a discount factor $\gamma = 0.99$) under the aforementioned stochastic policy. Crucially, the agent performed this task after experiencing only a limited number of trajectories. The grey arrows show an example trajectory, with the actual and estimated values for these trajectories shown in Extended Data Fig. 2g.

Consider the trajectory shown in Extended Data Fig. 2g. For this trajectory, the myopic estimate (using a discount factor $\gamma = 0.6$; green) clearly provides a better estimate of the true value function (grey) than using the discount factor $\gamma = 0.99$ (brown), which is the discount under which we computed the true value function. We quantified that the myopic estimate is a better approximation of the true value function by evaluating the agreement between pairs of states along the estimated and true curves. We evaluated accuracy using the Kendall rank correlation coefficient between the true value function in the entire maze and the value estimates. The Kendall coefficient measures the fraction of concordant pairs between the two value functions (across all pairs of states in the maze). For every pair of states, it computes whether the two value functions agree on which element of the pair is the larger one. Note that this measure of accuracy is behaviourally more relevant than alternative accuracy measures that compare the absolute magnitude of values across states. In other words, for an agent navigating the maze, it is more important to be accurate on the relative values of alternative goal states than on their absolute values.

In Extended Data Fig. 2h,i, the agent learned from *N* randomly sampled trajectories starting either in the lower half (blue) or upper half (red) of the maze. The values for the states in the *N* sampled trajectories were learned until convergence using the rewards and transitions in the sampled trajectories. After convergence, we computed the Kendall rank correlation between the estimates and the true value function, and reported performance as the average correlation across 10,000 sets of *N* sampled trajectories. Extended Data Fig. 2h shows that when learning from two randomly sampled trajectories, the estimates of the value function using a myopic discount factor are more accurate than far-sighted discounts when the trajectories start in the lower half of the maze (blue curve in Extended Data Fig. 2h). This result agrees with the intuition built in Extended Data Fig. 2g when learning from a single trajectory. However, if the agent is distant from important events (that is, trajectories starting in the upper half of the maze, red curve), the myopic estimates approach the noise level, whereas estimates with larger discount factors are more accurate. With the accumulation of more data from the environment, that is, more trajectories, the far-sighted estimate progressively aligns with the true value computed with $\gamma = 0.99$ in the entire maze (Extended Data Fig. 2i).

Myopic learning bias: networks with discount factors as auxiliary tasks. An alternative way to leverage multi-timescale learning benefits, in contrast to the architecture presented in Fig. 1d, is to use them as auxiliary tasks (Extended Data Fig. 2j). In this framework, the deep network acts according to the *Q*-value computed with a single behavioural timescale, but concurrently learns about multiple other timescales as auxiliary tasks to enhance the representation in the hidden layers, which allows them to obtain superior performance in complex RL environments[19,38,67,68]. This approach is similar to distributional RL networks that learn the quantiles of the value distribution but act according to the expectation of that distribution[17]. Of note, we showed that the auxiliary learning timescales display the myopic learning bias highlighted so far. In the Lunar Lander task (Extended Data Fig. 2j, left) in which the agent must land a spacecraft, *Q*-values computed using a myopic discount provide a more accurate representation of the future when the agent is close to the landing site (blue in Extended Data Fig. 2k), whereas the opposite holds when the agent is far from the landing site (red in Extended Data Fig. 2k), as shown in Extended Data Fig. 2l.

In the Lunar Lander environment, the state space consists of eight elements, including the position and velocity of the lander, its angular position and angular velocity, as well as an additional input related to the contact with the ground. The action space is composed of four actions: doing nothing and activating one of three different engines. The agent is a DQN[3] with two hidden layers of 512 units each, separated by ReLU activation functions. In addition to the *Q*-values that control the agent, the network has *Q*-values for 25 additional discounts factors equally spaced between 0.6 and 0.99. Thus, if there are |a| actions in the environment, for each discount the network has |a| additional output units. All sets of |a| units (one for each discount) use the Huber (that is, smooth L1, $\beta = 1$) *Q*-learning loss function with its corresponding

discount. All the auxiliary Q-learning losses update the action that was actually chosen in the environment by the behavioural Q-value units, and thus all of them learn the consequences of the behavioural policy, but using different discount factors. The total loss function uset to train the network averages the Q-learning losses of all the discount factors. To train the DQN, we used a learning buffer of 20,000 samples, a learning rate of 10^{-3} and a batch size of 32. As in traditional DQNs, we used a target network to compute the TD target, which is updated every 1,000 samples with the weights from the policy network to stabilize the learning process. For exploration, the agent uses a linearly decreasing ε-greedy policy that goes from $\varepsilon = 1.0$ at the first sample to a minimum value of $\varepsilon = 0.01$ after 40,000 samples.

Our goal was to compute the degree to which Q-values computed with alternative discounts can capture the true Q-value of the behavioural policy. The multi-timescale DQN uses a behavioural discount $\gamma_{\text{beh}} = 0.99$, and its policy is produced by choosing actions that maximize the Q-values with that discount factor. As in the navigation scenario presented in the previous section, our hypothesis was that, when important events lie in the proximal future (here, close to the landing site), the Q-values learned using myopic discounts capture the true behavioural Q-value more accurately, whereas far-sighted discounts are more accurate when important events lie in the distant future (far from the landing site).

Under the policy of the DQN (π_{DQN}), the true value of state s is:

$$V^{\text{true}}_{\gamma_{\text{beh}}}(s) = E_{\pi_{\text{DQN}}}\left[\sum_{t=0}^{T} \gamma^{t}_{\text{beh}} r_t\right] \tag{13}$$

If the DQN has perfectly learned the Q-value of state s, then the estimate $Q_{\gamma}(s, a_{\text{beh}})$ of the DQN should be equal to $V^{\text{true}}_{\gamma_{\text{beh}}}(s)$, where a_{beh} is the action produced by the DQN at s. For the analysis, we computed the true value of state s by simulating the policy a large number of times. We evaluated accuracy as the degree to which the estimated $Q_{\gamma}(s, a_{\text{beh}})$ captures the true $V^{\text{true}}_{\gamma_{\text{beh}}}(s)$, and compared accuracy across the auxiliary discount factors.

After training the network for 50,000 samples (and achieving close-to-optimal performance), we computed $V^{\text{true}}_{\gamma_{\text{beh}}}(s)$ empirically across states by recording the actual discounted sum of rewards obtained by the agent when departing from state s. We calculated $V^{\text{true}}_{\gamma_{\text{beh}}}(s)$ empirically for 25,000 states. Then, we compared, across states, the empirically calculated $V^{\text{true}}_{\gamma_{\text{beh}}}(s)$ with the Q-values produced by the DQN at those states.

To measure accuracy, we used the Kendall rank correlation as in the previous section. The Kendall correlation measures the fraction of concordant pairs between $V^{\text{true}}_{\gamma_{\text{beh}}}$ and the estimated Q_{γ}, across sampled pairs of states. As in the navigation scenario presented in the previous section, for an agent deciding which state to navigate to, it is more important to be accurate on the relative values between pairs of states than on the absolute value of individual states. Therefore, the Kendall correlation is behaviourally more relevant than other accuracy metrics that compare the absolute magnitude (for example, $(V^{\text{true}}_{\gamma\text{beh}}(s) - Q_{\gamma}(s, a_{\text{beh}}))^2$).

Given that the environment and the training process are stochastic, we reported the accuracy by averaging over 10 randomly initialized networks.

Cued delay task

All the data in the experiments with mice were collected in the previous study[44]. The experimental details, including the surgical procedures, behavioural setup and the behavioural tasks, have been described there[44]. Here we focus on the task description as our analysis includes task conditions that were not analysed in the previous study.

Mice were head-fixed on a wheel in front of three computer monitors and an odour port. At trial onset, the screens flashed green to indicate the beginning of the trial. After $t = 1.25$ s, an odour cue was delivered. This reward delay cue was one of four possible odours, and each cue was associated with a unique reward delay chosen from 0.6, 1.5, 3.75 or 9.375 s. The association between odour and reward delay was randomized across mice. The inter-trial interval was adjusted depending on the reward delays such that the trial start cues were spaced by 17–20 s. Mice performed 81.4 ± 12.5 trials (mean ± s.d.) per session across the 36 sessions in which neurons were recorded in the task.

Approach-to-target virtual reality task

We refer the reader to the previous study for details on the experimental procedures[44]. Mice were also trained in additional conditions, which we did not analyse in the present study, including teleport and speed modulation in the virtual reality scene.

Here we analysed single-neuron recordings in the sessions with no teleport or speed manipulation and in the open-loop condition. Mice were free to locomote but their motion did not affect the dynamics of the visual scene. After scene motion onset, the visual scene progressed at constant speed until reward was delivered after 7.35 s.

Mice performed 58.8 ± 21.7 trials (mean ± s.d.) per session across the 60 sessions in which neurons were recorded in the task. Spiking activity was convolved with a box filter of length 10 ms. When plotting neural activity, we further convolved the responses by a causal exponential filter ($e^{-0.05\text{dt}}$). Spiking-rate traces across neurons were normalized using a modified z-score. The mean was taken as the average firing activity cross the first 1.5 s and the standard deviation across the entire 4.35 s.

Fitting neural activity in the cued delay task

For the cued delay task, we fit the responses of single neurons to the delay cue (calculated as the firing rate in the time interval $0.1\text{ s} < t < 0.4\text{ s}$ after the cue onset; see shaded area in Fig. 2c) using two discounting models as in ref. 63, the classic exponential model and a hyperbolic model. For the exponential model, we fit the responses to a cue predicting a reward in τ seconds by:

$$FR_{\text{exp}} = b + \alpha\gamma^{\tau} = b + \alpha e^{-\lambda\tau} \tag{14}$$

The discount factor γ can also be expressed as a discount rate λ and vice versa: $\lambda = -\ln\gamma$ or $\gamma = e^{-\lambda}$. The discount factors fitted to data are always expressed in units of seconds, that is, the discount factor is the devaluation 1 s into the future.

For the hyperbolic model, we used a standard model for hyperbolic discounting in which the parameter k controls discounting:

$$FR_{\text{hyp}} = b + \alpha\frac{1}{1 + k\tau} \tag{15}$$

We fit both models by minimizing the mean squared error (the fit function in MATLAB). For both models, we constrained the baseline and gain parameters such that $0 < b < 40$ and $0 < \alpha < 40$. For the exponential model, the discount rate was constrained such that $0.0001 < \lambda < 20$, and for the hyperbolic model, the discount parameter was constrained such that $0 < k < 20$. Note that all the parameters were fitted independently for each single neuron.

To characterize the robustness and significance of our estimated parameters, we used a bootstrap procedure. For each run, we split the trials in half and fit the models independently on each half. We computed for each split the explained variance using the other half of the data (Extended Data Fig. 3c,d) and correlated the inferred parameter values for each neuron across both splits (Extended Data Fig. 3f–h).

We restricted our subsequent analysis to neurons that had a positive explained variance on the test set ($n = 17$ neurons excluded), an average firing rate in the cue period over the 4 delays above 2 spikes per second ($n = 11$ neurons excluded) and with mediolateral distance above 900 μm ($n = 4$ neurons excluded). Non-selected neurons are shown in Extended Data Fig. 3b. Poorly fit neurons often were non-canonical

dopaminergic neurons that also did not exhibit a strong reward response.

Decoding expected reward timing from population responses

The vectorized prediction error allows us to directly decode the expected timing of reward given the cue responses[21]. The value at time t is given by:

$$V_t = E\left[\sum_t^T \gamma^t r_t\right] = \gamma^{\Delta t} E(r|\Delta t) + \gamma^{2\Delta t} E(r|2\Delta t) + \ldots + \gamma^T E(r|T) \quad (16)$$

In the cued delay task, at the time of the cue indicating reward delay, the response of dopaminergic neurons is driven by the discounted future reward. The reward prediction error $\delta_t = r_t + \gamma^{\Delta t} V_{t+1} - V_t$ becomes simply $\delta_t = \gamma^{\Delta t} V_{t+1} + cst$ as there is no reward delivered at the time of the cue ($r_{t_{\text{cue}}} = 0$) and the reward expectation before the reward cue delivery is identical across conditions ($V_{t_{\text{cue}}} = C$; where C is a constant). Thus, the TD error at the time of reward delay cue ($\delta_{t_{\text{cue}}} = r_{t_{\text{cue}}} + \gamma^{\Delta t} V_{t_{\text{cue}}+\Delta t} - V_{t_{\text{cue}}}$) becomes $\delta_{t_{\text{cue}}} = \gamma^{\Delta t} V_{t_{\text{cue}}+\Delta t} + C$, and if we assume the constant is 0 or the TD error is baseline subtracted, at convergence the prediction error is given by:

$$\delta_i = \begin{bmatrix} \gamma_i^{\Delta t} & \gamma_i^{2\Delta t} & \ldots & \gamma_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} E(r|\Delta t) \\ E(r|2\Delta t) \\ \vdots \\ E(r|T) \end{bmatrix} \quad (17)$$

In single-timescale RL, the temporal information is collapsed, and it is not possible for the system receiving the learning signal (the striatum in this case) to untangle the signal. However, in a distributed system learning at multiple timescales, the reward expectation $E(r|t)$ is encoded with multiple discount factors γ_i:

$$\begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix} = \begin{bmatrix} \gamma_1^{\Delta t} & \gamma_1^{2\Delta t} & \cdots & \gamma_1^{\mathrm{T}} \\ \gamma_2^{\Delta t} & \gamma_2^{2\Delta t} & \cdots & \gamma_2^{\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_n^{\Delta t} & \gamma_n^{2\Delta t} & \cdots & \gamma_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} E(r|\Delta t) \\ E(r|2\Delta t) \\ \vdots \\ E(r|T) \end{bmatrix} = \mathbf{L} p(r|t) \quad (18)$$

The temporal information about reward timing is now distributed across neurons, and if the tuning of individual neurons is sufficiently diverse, we can write:

$$\begin{bmatrix} E(r|\Delta t) \\ E(r|2\Delta t) \\ \vdots \\ E(r|T) \end{bmatrix} \approx \mathbf{L}^{-1} \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix} \quad (19)$$

Where $\mathbf{L}^{-1}$ is the approximate pseudo-inverse of $\mathbf{L}$ such that $\mathbf{L}^{-1}\mathbf{L} \approx I$. In practice, the matrix $\mathbf{L}$ is not very well conditioned as the rows of the matrix are exponentially decaying functions, so the right side (further in the future) is sparsely populated (Fig. 2i and Extended Data Fig. 4a–d). We therefore need to use a regularized pseudo-inverse.

To invert the discount matrix $\mathbf{L}$, we used the regularized singular value decomposition (SVD) approach similar to the one proposed in ref. 21. We then normalized the resulting prediction to constrain it to be a probability distribution ($p(r|t) > 0$, for all t and $\sum_t p(r|t) = 1$). More specifically, the regularized SVD approach corresponds to optimizing:

$$||\mathbf{L}p(r|t) - \Delta_d||^2 + \alpha^2 ||E(r|t)||^2 \quad (20)$$

The standard SVD of the discount matrix can be written as:

$$\mathbf{L} = \sum_{s=1}^{L} \sigma_s u_s v_s^T = USV^T \quad (21)$$

$$E(r|t) \approx \sum_{s=1}^{L} \left(\frac{\sigma_s^2}{\alpha^2 + \sigma_s^2}\right) \frac{u_s^T \Delta_d v_s}{\sigma_s} \equiv \mathbf{L}^{-1}\Delta_d \quad (22)$$

where $\Delta_d = [\delta_1 \ldots \delta_N]^T$. The smooth regularization introduced by the Tikhonov regularization through the parameter α (which we can choose by inspection of the distribution of singular values σ_s, see below) is more robust than a strict truncated SVD in which we only take a number of factors and set the remaining factors to zero. An alternative approximation to this inverse problem is Post's approximation[34,37]. It relies on evaluating higher-order derivatives and lacks robustness if the Laplace space is not sampled with enough precision (that is, not enough neurons tiling the γ space).

The procedure in the previous section allows us to estimate the discount factor independently for each neuron. We then choose a discretization step $\Delta t = 100$ ms and a temporal horizon $T = 12$ s over which to make the prediction. This allowed us to construct the discount matrix $\mathbf{L}$ shown in Fig. 2i for the exponential model and Extended Data Fig. 6c for the hyperbolic model. To choose a suitable value for the regularization parameter α, we performed the regular SVD on the discount matrix $\mathbf{L}$ and assessed the values at which the singular values become negligible. We chose a value of α that corresponded to the transition between large singular values and negligible values (Extended Data Fig. 4b). Using this approach, we used $\alpha = 2$ in our decoding analysis.

For each delay, we constructed a pseudo-population response Δ_d across the recorded neurons. For each bootstrap, we took the mean activity for each cue, subtracted the inferred baseline parameter b and normalized the maximum response to 1. To assess the robustness of the predictions, we used the mean responses and baseline from half the trials to construct Δ_d, used the estimated discount factors from the other half of the trials to estimate $\mathbf{L}^{-1}$ and we repeated this approach for each bootstrap ($n_{\text{predictions}} = 200$). In the figures (Fig. 2k and Extended Data Figs. 5g, 6d,f and 10c), the thin lines correspond to the predictions from individual bootstraps and the thicker line to the average of these predictions. For shuffle control, we randomized the identity of the neurons in the pseudo-population response Δ_d. This means that in the shuffle control, a given neuron is not decoded with its corresponding weights but by a random row of the decoding matrix $\mathbf{L}^{-1}$.

To ensure that the prediction corresponded to a probability distribution, we normalized the resulting prediction of reward timing. We first set the probability of obtaining a reward to zero for all times in which the prediction was negative, then we normalized the distribution to be a valid probability distribution (such that the probability mass over $t \in [0,12]$ summed to 1).

For the time decoding using a single average discount factor, we used a different approach. The inversion procedure would not work as the discount matrix would be of rank 1. Instead, if we assume a fixed known reward size and a single discount factor, the response of individual neurons would correspond to different estimates of the reward timing. For each bootstrap, we estimated the expected reward timing for each neuron. For a given firing rate FR for the held-out data, we estimated the reward timing using the parameter estimates from the trained data. The baseline b_i and gain α_i parameters are specific to each neuron, whereas the discount factor γ is the average discount factor across all the neurons. The expected reward timing for neuron i is given by the following equation:

$$E_i(t) = \frac{\log \max\left(\frac{\mathrm{FR}_i - b_i}{\alpha_i}, 0.0001\right)}{\log \gamma} \quad (23)$$

Together, the neurons provide a distribution of expected reward timing with each neuron predicting a sample of the distribution of expected reward times. The average distribution is obtained by averaging the distributions across all the bootstraps, excluding predicted reward times beyond 12 s and normalizing the distribution to be a probability

distribution. Similarly to the SVD-based decoding, in Extended Data Fig. 4f, the thin lines correspond to the predictions from individual bootstraps and the thicker line to the average of these predictions.

Quantifying reward timing decoding accuracy

To quantify the reward timing decoding accuracy, we used the 1-Wasserstein distance (or earth mover's distance) between distributions as our metric. We used the 1-Wasserstein distance as the difference in support between the predicted reward timing distribution (probability mass as most locations) and the single true reward timing (probability mass at a single location) is not conducive to using the Kullback–Leibler divergence.

For each bootstrap, we generated $n = 100{,}000$ samples from the predicted reward timing distributions and computed the 1-Wassertstein distance between the predicted reward timing and the true corresponding reward delay (using the MATLAB function ws_distance from https://github.com/nklb/wasserstein-distance). For each condition (exponential fit, hyperbolic fit, average discount factor, simulation fit and their associated shuffled predictions), we obtained a distribution of 1-Wasserstein distances across the bootstraps ($n = 200$). To assess the significance of the differences in reward timing predictions across conditions, we used the one-tailed Wilcoxon's signed-rank test (using the MATLAB function signrank).

Analysing behavioural discounting through the lick rate in the cued delay task

To quantify the influence of behaviour on the discount factors inferred at the single-neuron level, we analysed the relationship between the behavioural discounting and the neural discounting. To quantify behavioural discounting, we used the anticipatory lick rate in the 0.6 s following the delay cue. For each neuron, we computed the average lick rate for each of the four delays. We then fitted an exponential model (as for the neurons) to the four average lick rates for the four delays (shown in Extended Data Fig. 5a,d, left panels). For each neuron, we therefore obtained a behavioural discount factor. To quantify the relationship between behavioural and neural discount factor, we used the Spearman rank correlation.

Comparing the neural discount factor as a function of lick rate

To investigate the effect of the of the lick rate on the discount factor measured in single neurons, we also compared the modulation of the inferred discount factor with the lick rate. For each neuron and each reward delay, we split the trials into low and high lick rate trials depending on whether the lick rate in the entire anticipation period was strictly below or above (or equal) to the median lick rate for this reward delay. We then fit the exponential discounting model separately for low lick rate trials and high lick rate trials. We compared the difference in the inferred discount factor for each neuron across the two conditions. We performed this analysis at the single-animal level for the two animals with sufficient number of neurons (Extended Data Fig. 5h,i).

Analysing behavioural discounting through the lick rate in the virtual reality task

To quantify the relationship between behaviour and neural activity in the virtual reality task as mice approach the reward location, we computed a measure of ramping in the lick rate and compared it with our measure of neural discounting. For each neuron, we computed the average lick rate during the reward anticipation period as the mice are moving along the linear track. We computed the average lick rate in three windows: early (−3.45 s to −2.95 s before reward delivery), middle (−1.95 s to −1.45 s before reward delivery) and late (−0.75 s to −0.25 s before reward delivery). Using these windows, we computed three modulation indices using the following equation:

$$MI_{2-1} = \frac{\text{Lick rate}_2 - \text{Lick rate}_1}{\text{Lick rate}_2 + \text{Lick rate}_1} \tag{24}$$

We compared these modulation indices to the inferred discount factors, showing no significant correlations, whereas the three measures of licking are strongly correlated among themselves (Extended Data Fig. 8).

Fitting neural activity in the virtual reality task

To quantify the heterogeneity of discount factors in the virtual reality task, we fit the neural activity in the last 4.30 s ($t = 3.05$ s after scene motion onset) of the approach to reward period in which the ramping activity was most pronounced. To assess the robustness of the fit, we used a bootstrap procedure in which for each bootstrap ($n_{\text{bootstrap}} = 100$), we partitioned the trials into two halves and computed the two average PSTHs using $dt = 0.1$ s as our discretization step. We then computed the mean value of the parameters across all bootstraps. We limit our analysis to neurons whose firing rate over the analysis period is larger than 2 spikes per second. We fit the two models (common value function and common reward timing expectation) to this data.

In the virtual reality task, the expectations vary smoothly as a function of time and distance and we therefore use the discretized formulation of the TD error for continuous time in our fits[45,47]:

$$\delta_i(t) = b_i + \alpha_i\left(\gamma_i^{dt}\frac{dV(t)}{dt} - \gamma_i^{dt}\ln(\gamma_i)V(t)\right) \tag{25}$$

Although this formulation is also discretized as the standard formulation of the TD error, the presence of the derivative $\frac{dV_i(t)}{dt}$ (which is computed numerically) improves the stability of the fitting procedure. The two models differ in whether value function is estimated directly (and shared across neurons) or indirectly (and distinct across neurons). The discount factor is also in units of seconds, allowing comparison with the values estimated in the cued delay task.

Common value function model

In the common value function model, $V(t)$ is common across neurons and is directly fitted by the optimization procedure which minimizes:

$$\min_{\alpha,b,\gamma,V} ||\text{FR} - \Delta||^2 \tag{26}$$

With,

$$\Delta = \begin{bmatrix} \delta_1(t_0) & \cdots & \delta_1(T) \\ \vdots & \ddots & \vdots \\ \delta_n(t_0) & \cdots & \delta_N(T) \end{bmatrix} \tag{27}$$

We fit the gains, baseline, and discount factors of individual neurons (α_i, b_i and γ_i respectively) and the join value function V using a constrained optimization procedure (*fmincon* in MATLAB, $\alpha_i \in [0.05,50]$, $b_i \in [0.05,12]$, $\gamma_i \in [0.05,0.999999]$, and $V \in [0.05,5]$).

We performed this analysis both on the full population of neurons that passed the inclusion criteria (Fig. 3) as we well as independently for the subsets of neurons belonging to m3044 (29 neurons) and m3054 (24 neurons; Extended Data Fig. 8e–j).

Common reward expectation model

In the common reward expectation model, the reduction in uncertainty in reward timing due to sensory feedback as the mice approach the reward leads to an upwards ramp in the average TD error signal across dopaminergic neurons[44–46]. In a task such as the cued delay task shown in Fig. 2, once the cue has been presented, the time estimation until the reward is based on the internal clock of the mice that experiences scalar timing (that is, the standard deviation of the noise in the estimation grows linearly with the estimation time)[26]. In the virtual reality task, there is visual feedback, and as the mice approach the reward, the uncertainty is instead reduced (Extended Data Fig. 9a). We also showed that this alternative model also provides a similar

explanation of ramping diversity as originating from a heterogeneity of discount factors.

We use a joint fitting procedure in which we simultaneously fit the discount factors across neurons and the expected timing of reward as a function of position in the virtual track. Similarly to ref. 45, we interpret the ramping in single neurons as originating from the reduction in uncertainty due to the visual feedback as the mice approach the reward. Although each neuron has a distinct discount factor and its own value function, the world model, which parametrizes the changes in reward expectation with visual feedback, is shared across dopaminergic neurons. This arises as this shared model is the product of the integration of the diverse dopamine signals, as well as of other neural computations that control reward expectations[69].

Individual neurons therefore act as independent agents estimating value given a shared expectation of reward timing. Each neuron has a distinct discount factor γ_i with which it computes value given the expected reward timing. We assumed that inference has converged and therefore we have the value V_i associated with neuron i:

$$V_i = \sum_{\tau=t}^{T} \gamma_i^{\tau-t} E(r|\tau, t, T) \tag{28}$$

Here we assumed that $E(r|\tau, t, T)$ takes the form a folded normal distribution with parameters $\mu = T - t$ and (fitted) standard deviation σ. The folded normal distribution reflects the weight of the negative component of a normal distribution back onto positive values[70]. The folded normal distribution formulation leads to the following distribution for the expected reward timing for $\tau > 0$:

$$E(r|\tau, t, T) = \sqrt{\frac{2}{\pi\sigma^2}} e^{-\frac{(\tau^2+(T-t)^2)}{2\sigma^2}} \cosh\left(\frac{(T-t)\tau}{\sigma^2}\right) \tag{29}$$

In our analysis, the mean, $\mu = T - t$, is given by the current position in the virtual reality track and the only fitted parameter is the standard deviation σ. At each timestep, we fit a different value of the standard deviation. As observed through the fitting procedure, the standard deviation was initially high and reduced as the mice approached the reward location. This is an indication that similarly than proposed in ref. 45, the ramping in activity in the dopaminergic neuron arises from the reduction in uncertainty due to the visual feedback as the mice approach the reward. We used a slightly different formulation than in ref. 45 as we required additional flexibility to fit data and specifically needed to go beyond the assumptions of Gaussian state uncertainty. Note also that here we assumed that the uncertainty is in the timing of the reward rather than in the state.

To normalize the contributions of the different neurons, we used a normalized firing rate and therefore only fit the discount factor γ and standard deviation σ of the reward expectation.

$$\min_{\gamma,\sigma} ||\mathrm{FR} - \Delta||^2 \tag{30}$$

With,

$$\Delta = \begin{bmatrix} \delta_1(t_0) & \cdots & \delta_1(T) \\ \vdots & \ddots & \vdots \\ \delta_n(t_0) & \cdots & \delta_N(T) \end{bmatrix} \tag{31}$$

We performed the constrained optimization with the MATLAB function fmincon and constrained the parameters such that $\gamma \in [0.001, 0.99]$ and $\sigma \in [0.1, 12]$.

Mixing in distributed RL models

When explaining how multi-timescale RL can explain the diversity of ramping activity, we proposed two possible interpretations: one with a common value function across all neurons and another with a common reward timing estimation; whereas in the cued delay task, each dopaminergic neuron contributed to learning an independent value function. Here we reconcile these approaches as shown in Extended Data Fig. 7f–k. We can understand these different models as a spectrum in which the value functions used for the computations for each discount factor is more or less shared across them.

In 'classic' multi-timescale TD learning[19], the values (V_i) and RPEs (δ_i) for the different discounts (γ_i) are updated independently, which guarantees its convergence. Now consider a situation in which the value–RPE circuits of the multiple discounts share a common value function, to a degree λ between 0 and 1. In this case, the next value estimate in the timescale-independent TD backup is corrected by:

$$V_i(s_t) \leftarrow V_i(s_t) + \alpha(r_t + \gamma_i \widetilde{V}_i(s_{t+1}) - V_i(s_t)) \tag{32}$$

$$\widetilde{V}_i = \lambda \widetilde{V} + (1 - \lambda) V_i \tag{33}$$

Where $\widetilde{V}$ is the mean value function across all discounts. The main motivation for this modification of the traditional TD backup is neuroanatomical, as it is plausible to consider a degree of common shared activity across nearby value units. Unlike the model with a fully shared common value function (that is, $\lambda = 1$), multi-timescale learning with small values of the sharing parameter (for example, $\lambda = 0.1$) preserve, to a large degree, all the computational advantages of multi-timescale learning, while being more biologically realistic than fully separated circuits (that is, $\lambda = 0$). Using $\lambda = 0.1$ in the four tasks of Fig. 1e (while keeping the same simulation parameters as in Fig. 1; see 'Decoding tasks' in Methods), the accuracy of the report of the agent with three discounts {0.6,0.9,0.99} is 82 ± 5% in task 1, 94 ± 2% in task 2, 92 ± 3% in task 3 and 93 ± 3% in task 4. Therefore, regularizing the independent value functions with a small degree of shared activity preserves all the multi-timescale advantages highlighted so far.

With this modification to the learning rule, V_i does not converge to a pure exponential form anymore (compare dashed lines with solid lines in Extended Data Fig. 7f–h value panels, middle row), even with a small sharing parameter ($\lambda = 0.1$ in Extended Data Fig. 7g). As a result, the RPE does not converge to 0 across the trajectory (Extended Data Fig. 7g,h, δ, bottom row), so TD learning does not fully converge at the level of individual value estimators. However, we found empirically that learning stabilizes completely after 1,000 TD backups (using $\alpha = 0.1$). Crucially, owing to the non-exponential form of the learned value function, we observed that $\gamma < \widetilde{\gamma}$ ($\gamma > \widetilde{\gamma}$) have RPEs that mostly ramp down (or up), so our characterization of cell-specific discounts based on ramping patterns is mostly independent of which of the ramping interpretations we adopt. These ramping patterns across timescales are robust when varying the magnitude of the sharing parameter λ ($\lambda \in [0,1]$), the learning rate α and the number of TD backups (we used a learning rate α of 0.01 and 2,000 TD backups for Extended Data Fig. 7f–h). For the simulations in Extended Data Fig. 7f–h, we used the discounts fitted experimentally in Fig. 3l (90 units), and plot only three discounts in Extended Data Fig. 7f–h (renormalized to lie between −1 and 1) corresponding to the 20th, 70th and 90th percentiles, corresponding to discounts 0.25, 0.56 and 0.88.

The model in the cued delay task corresponds to $\lambda = 0$. The common value function model that we propose would be similar to a value of $\lambda = 1$, but note that the common value model used for fitting in equation (25) is slightly different than in equation (32), as in equation (32) the shared value is only used for estimating future value. The model from equation (32) corresponds more closely to the common reward timing estimation model (Extended Data Fig. 9) in which the reduction in uncertainty with visual feedback affects the estimate of future value as outlined below. In this model, the TD error for neuron i can be written as follow: $\delta_i = \gamma_i V'_i - V_i$.

$V_i = \sum_{\tau=t}^{T} \gamma_i^{\tau-t} E(r|\tau, t, T) = \sum_{\tau=t}^{T} \gamma_i^{\tau-t} E_\tau$ is the value before the sensory feedback and $V'_i = \sum_{\tau=t+1}^{T} \gamma_i^{\tau-t-1} E'(r|\tau, t+1, T) = \sum_{\tau=t+1}^{T} \gamma_i^{\tau-t-1} E'_\tau$ is the value of the next state, including the sensory feedback and the reduction in uncertainty in reward timing. To highlight the contribution of the sensory feedback, we also introduced $V_i^{t+1} = \sum_{\tau=t+1}^{T} \gamma_i^{\tau-t-1} E(r|\tau, t+1, T) = \sum_{\tau=t+1}^{T} \gamma_i^{\tau-t-1} E_\tau$, which would be the value at the next step in the absence of sensory feedback (and therefore no reduction in uncertainty about reward timing).

We can rewrite the TD error as follows:

$$\begin{aligned} \delta_i &= \gamma_i V'_i - V_i = \gamma_i V'_i - \gamma_i V^{t+1}_i + \gamma_i V^{t+1}_i - V_i \\ \delta_i &= \gamma_i V^{t+1}_i - V_i + \gamma_i \Delta V_i \end{aligned} \tag{34}$$

Here the correction due to the sensory feedback appears as ΔV_i, which we can also write as $\sum_{\tau=t+1}^{T} \gamma_i^{\tau-t-1} (E'_\tau - E_\tau)$. Similarly than in ref. 45, the sensory feedback acts as a correction term in the prediction error computation. Here the shared correction term is the reduction in uncertainty, so it takes a slightly different form than in the general formulation with shared value $\widetilde{V}$ and would correspond to a case in which the sharing parameter depends on the discount factor. This source of 'regularization' could occur through different pathways. Here it is the reduction in uncertainty due to the structure of the virtual reality task that leads to a reduction in the uncertainty about reward timing as the mice approach the reward. This contribution of a shared signal or estimate from a parallel value estimation has also been used to explain non-canonical prediction errors in motor tasks[71]. In the cued delay task, there is no feedback about reward timing and therefore we have $\Delta V_i = 0$ in equation (34), and this would correspond to a situation in which the loops are entirely decoupled ($\lambda = 0$). In practice, we might expect some low level of coupling given the anatomical considerations outlined above. As long as the loops are not completely coupled ($\lambda \neq 1$), there is enough information to leverage the computational advantages shown in Fig. 1 and perform the decoding shown in Fig. 2 (Extended Data Fig. 7i–k).

Comparing parameters across tasks

We used two methods to assess the relationship between the inferred discount factors in the approach-to-reward virtual reality task and the cued delay task. First, we used the mean parameters across bootstraps and computed the Spearman rank correlation. Next, we computed, for $n = 10{,}000$ randomly selected (with replacement) pairs of bootstraps, the Spearman rank correlation between the parameters across the two tasks and plotted the distribution of these correlation.

For the decoding of reward timing using parameters inferred in the virtual reality task, we also used a bootstrap approach. We computed the discount matrix and the decoding matrix for each bootstrap estimate of the discount factors in the virtual reality task.

Simulations to assess limits on parameter estimation

To assess the contribution of the limits imposed by the number of trials and the stochasticity in firing rates to the accuracy of the reward timing prediction and the similarity of inferred parameters across tasks, we ran a series of simulations with parameters chosen to match those inferred from the data. For the simulation parameters, we used the mean inferred value for the parameters across all the bootstraps for the respective task.

For the cued delay task, for each neuron, we generated $n = 80$ trials ($n = 20$ per delay), comparable with behavioural sessions in the task, simulated cue responses by taking samples from a Poisson distribution with a rate parameter corresponding to the value predicted by the exponential discount model for the corresponding reward delay. We used the same procedure as for analysing the recorded data by performing $n = 100$ bootstraps and fitting the simulated data on random partitions of the data.

For the virtual reality task, for each neuron, we generated $n = 80$ trials, comparable with behavioural sessions in the task, by taking samples from a Poisson distribution with a rate parameter corresponding to the predicted activity given equation (22). We then performed the fitting procedures similarly than for the experimental data.

Reporting summary

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

Data availability

The raw electrophysiological data can be found on DANDI Archive (https://dandiarchive.org/dandiset/000251). The curated electrophysiological data can be found at https://doi.org/10.17632/tc43t3s7c5.1 (see ref. 72).

Code availability

The code used for simulations can be found on GitHub (https://github.com/pablotano8/multi_timescale_RL). The data analysis code for the electrophysiological experiments can be found at https://doi.org/10.17632/tc43t3s7c5.1 (see ref. 72).

61. Kvitsiani, D. et al. Distinct behavioural and network correlates of two interneuron types in prefrontal cortex. *Nature* **498**, 363–366 (2013).
62. Sutton, R. S. Learning to predict by the methods of temporal differences. *Mach. Learn.* **3**, 9–44 (1988).
63. Oppenheim, A., Willsky, A. & Hamid, W. *Signals and Systems* (Pearson, 1996).
64. Dayan, P. Improving generalisation for temporal difference learning: the successor representation. *Neural Comput.* **5**, 613–624 (1993).
65. Gershman, S. J. The successor representation: its computational logic and neural substrates. *J. Neurosci.* https://doi.org/10.1523/JNEUROSCI.0151-18.2018 (2018).
66. Amit, R., Meir, R. & Ciosek, K. Discount factor as a regularizer in reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning* 269–278 (PMLR, 2020).
67. Badia, A. P. et al. Agent57: outperforming the Atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning* 507–517 (PMLR, 2020).
68. Reinke, C. Time adaptive reinforcement learning. ICLR 2020 work. Beyond tabula rasa RL. Preprint at https://doi.org/10.48550/arXiv.2004.08600 (2020).
69. Gershman, S. J. & Uchida, N. Believing in dopamine. *Nat. Rev. Neurosci.* **20**, 703–714 (2019).
70. Leone, F. C., Nelson, L. S. & Nottingham, R. B. The folded normal distribution. *Technometrics* **3**, 543–550 (1961).
71. Lindsey, J. & Litwin-Kumar, A. Action-modulated midbrain dopamine activity arises from distributed control policies. *Adv. Neural Inf. Process. Syst.* **35**, 5535–5548 (2022).
72. Masset, P. et al. Data and code for 'Multi-timescale reinforcement learning in the brain', V1. *Mendeley Data* https://doi.org/10.17632/tc43t3s7c5.1 (2025).

Acknowledgements We thank S. J. Gershman and J. Mikhael for their contributions to the preceding studies; M. Watabe-Uchida for advice on task design; members of the Uchida and Pouget laboratories, including A. Lowet and M. Burrell, for discussions and comments; and W. Carvalho, G. Reddy and T. Ott for their comments on the manuscript. This work is supported by US National Institutes of Health (NIH) BRAIN Initiative grants (R01NS226753 and U19NS113201), NIH grant 5R01DC017311 to N.U., and a grant from the Swiss National Science Foundation (315230_197296) to A.P. This research was carried out in part thanks to funding from the Canada First Research Excellence Fund, awarded to P.M. through the Healthy Brains, Healthy Lives initiative at McGill University.

Author contributions P.M., P.T., A.P. and N.U. conceived the project. P.M., H.R.K., A.N.M. and N.U. designed the electrophysiology experiments. A.N.M. and H.R.K. performed the electrophysiology experiments and curated the data. P.T. performed the simulations with artificial agents. P.M. performed the analysis of electrophysiological data. P.M., P.T., A.P. and N.U. wrote the paper with input from H.R.K.

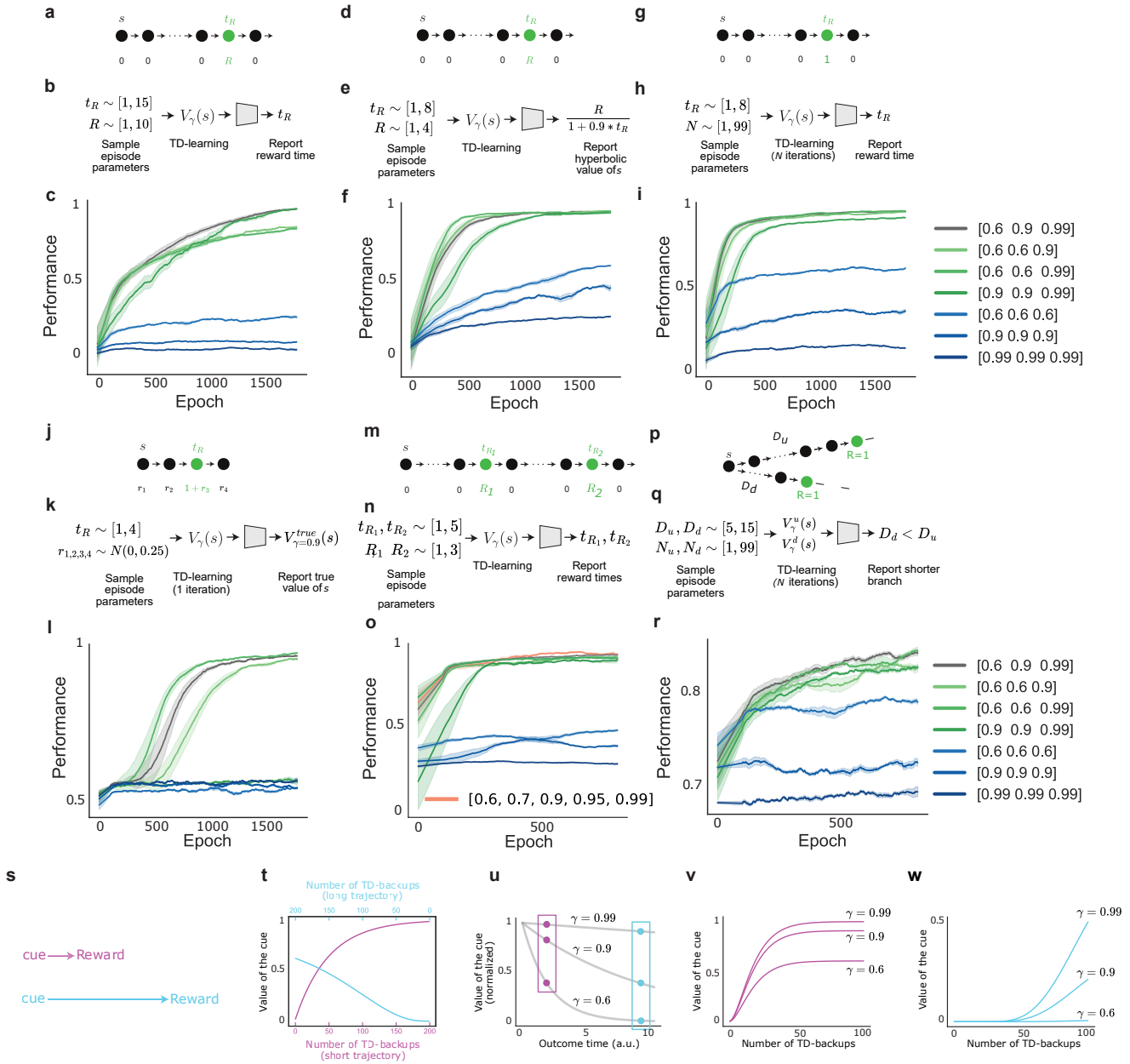
Competing interests The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-025-08929-9.

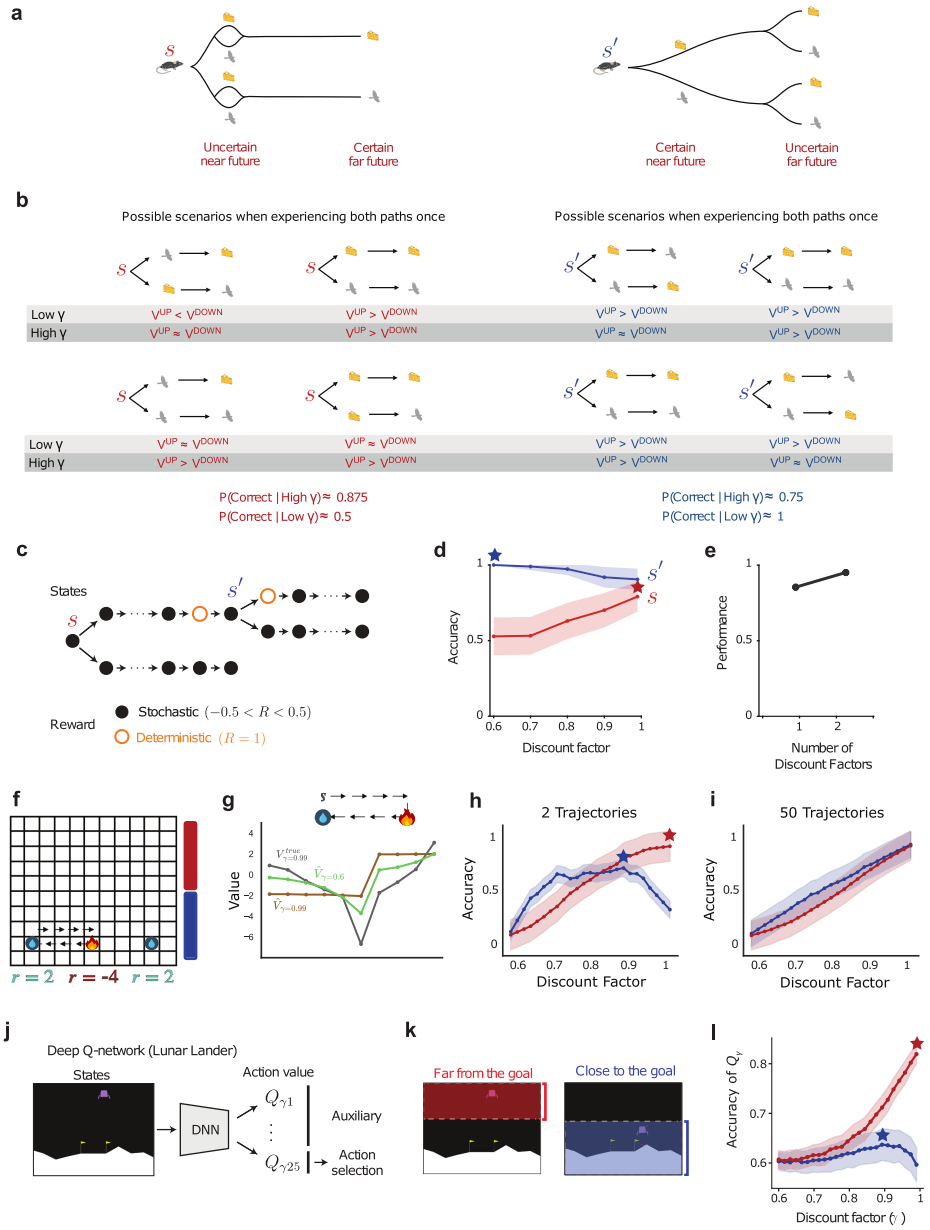
Correspondence and requests for materials should be addressed to Paul Masset, Alexandre Pouget or Naoshige Uchida.

Peer review information *Nature* thanks Kenji Doya and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.



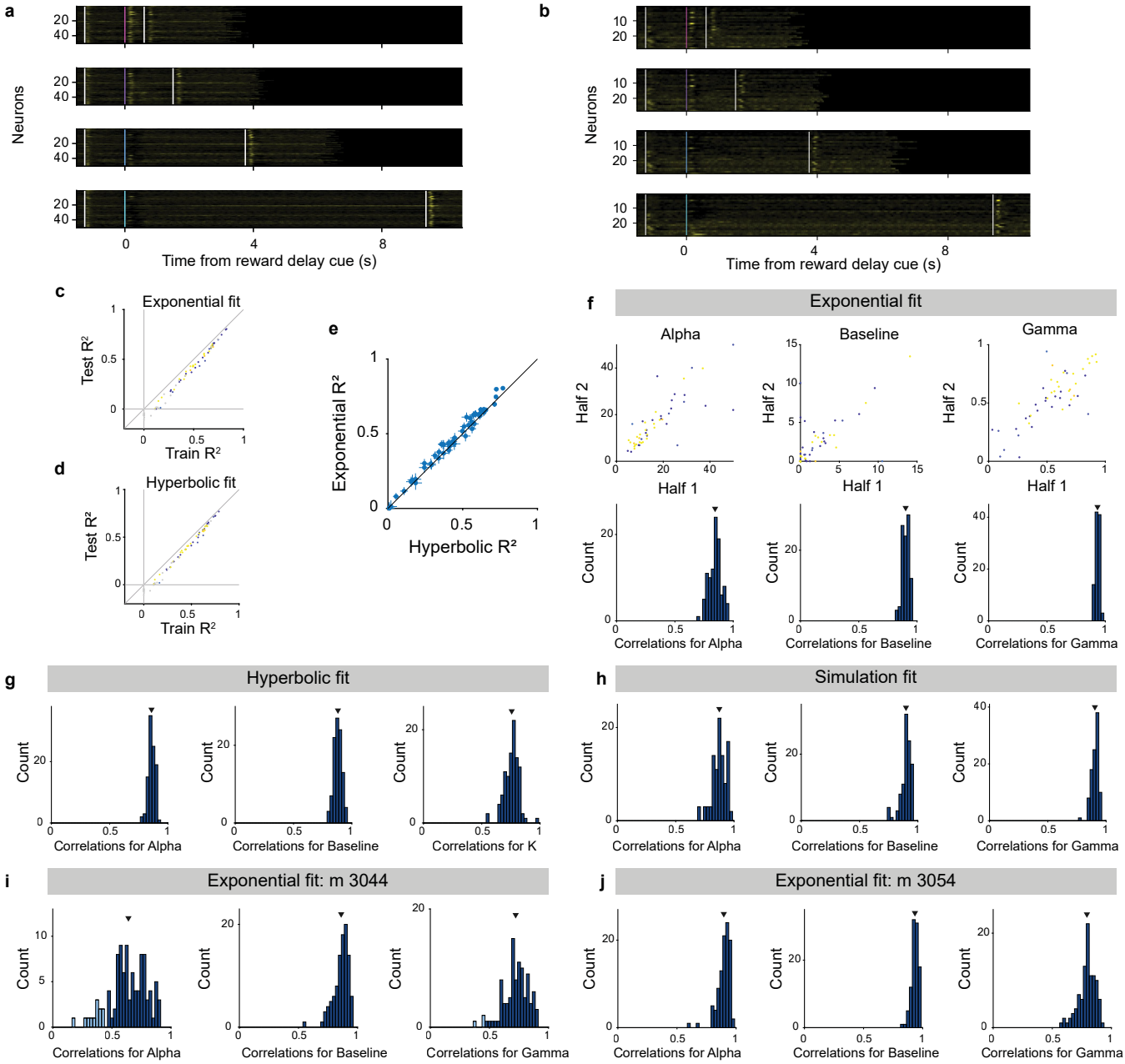
Extended Data Fig. 1 | See next page for caption.

Extended Data Fig. 1 | Decoding simulations for multi-timescale vs. single-timescale agents. (a-c). Experiment corresponding to **Task 1 in** Fig. 1. **a**, MDP with reward R at time t_R. **b**, Diagram of the decoding experiment. In each episode, the reward magnitude and time are randomly sampled from discrete uniform distributions, which defines the MDP in **a**. Values are learned until near convergence using TD-learning. Values with different discount factors are learned independently. The learned values for the cue (s) are fed into a non-linear decoder which learns, across MDPs, to report the reward time. **c**, Decoding performance as the decoder is trained. Different colors indicate the discount factors used in TD-learning. **(d-f)**. Experiment corresponding to **Task 2 in** Fig. 1. **d**, MDP with reward R at time t_R. **e**, Diagram of the decoding experiment. In each episode, the reward magnitude and time are randomly sampled from discrete uniform distributions, which defines the MDP in **a**. Values are learned until near convergence using TD-learning. Values with different discount factors are learned independently. The learned values for the cue (s) are fed into a non-linear decoder which learns, across MDPs, to report the hyperbolic value of the cue. **f**, Decoding performance as the decoder is trained. Different colors indicate the discount factors used in TD-learning. **(g-i)**. Experiment corresponding to Task 3 in Fig. 1. **g**, MDP with reward equal to 1 at time t_R. **h**, Diagram of the decoding experiment. In each episode, the reward time and the number of TD iterations (N) are sampled from discrete uniform distributions. Values are learned by performing N TD-learning backups on the MDP. Values with different discount factors are learned independently. The learned values for the cue (s) are fed into a non-linear decoder which learns, across MDPs, to report the reward time. **i**, Decoding performance as the decoder is trained. Different colors indicate the discount factors used in TD-learning. **(j-l)**. Experiment corresponding to Task 4 in Fig. 1. **j**, MDP with reward equal to 1 at time t_R, and a noisy reward added to every state. **k**, Diagram of the decoding experiment. In each episode, the reward time is sampled from discrete uniform distributions. Values are learned by performing a single iteration of TD-learning backwards through the MDP. Values with different discount factors are learned independently. The learned values for the cue (s) are fed into a non-linear decoder which learns, across MDPs, to report the true value of the cue after experiencing the trajectory an infinite number of times (this is, ignoring the random rewards). **l**, Decoding performance as the decoder is trained. Different colors indicate the discount factors used in TD-learning. **(m-o)** Experiment with two rewards. **m**, MDP with two rewards of magnitude $R1$ and $R2$ at times $t_{R1} < t_{R2}$. Value estimates $V_\gamma(s)$ are fed into a non-linear decoder which learns, across MDPs, to report both reward times. **o**, Decoding performance as the decoder is trained. Different colors indicate the discount factors used in TD-learning. **(p-r)** Experiment to determine the shortest branch when learning from incomplete information. **p**, MDP with two possible trajectories. In this example, the upwards trajectory is longer than the downwards trajectory. **q**, Diagram of the decoding experiment. In each episode, the length of the two branches D and the number of times that TD-backups are performed for each branch are randomly sampled from uniform discrete distributions. Then, TD-backups are perform ed for the two branches the corresponding number of times. After this, they are fed into a decoder which is trained, across episodes, to report the shorter branch. **r**, Decoding performance as the decoder is trained. Different colors indicate the discount factors used in TD-learning. **s-w**: Temporal estimates are available before convergence for multi-timescale agents. **s**, Two experiments, one with a short wait between the cue and reward (pink), and one with a longer wait (cyan). **t**, The identity of the cue with the higher value for a single-timescale agent (here $\gamma = 0.9$) depends on the number of times that the experiments have been experienced. When the longer trajectory has been experienced significantly more often than the short one, the single-timescale agent can incorrectly believe that it has a larger value. **u**, For a multi-timescale agent, the pattern of values learned across discount factors is only affected by a multiplicative factor that depends on the learning rate, the prior values and the asymmetric learning experience. The pattern therefore contains unique information about outcome time. **v,w**, When plotted as a function of the number of times that trajectories are experienced, the pattern of values across discount factors is only affected by a multiplicative factor. In other words, for the pink cue, the larger discount factors are closer together than they are to the smaller discount factor, and the opposite for the cyan cue. This pattern is maintained at every point along the x-axis, and therefore is independent of the asymmetric experience, and it enables a downstream system to decode reward timing. Error bars are the standard deviations (s.d.) across 100 test episodes and 3 trained policy gradient (PG) networks.



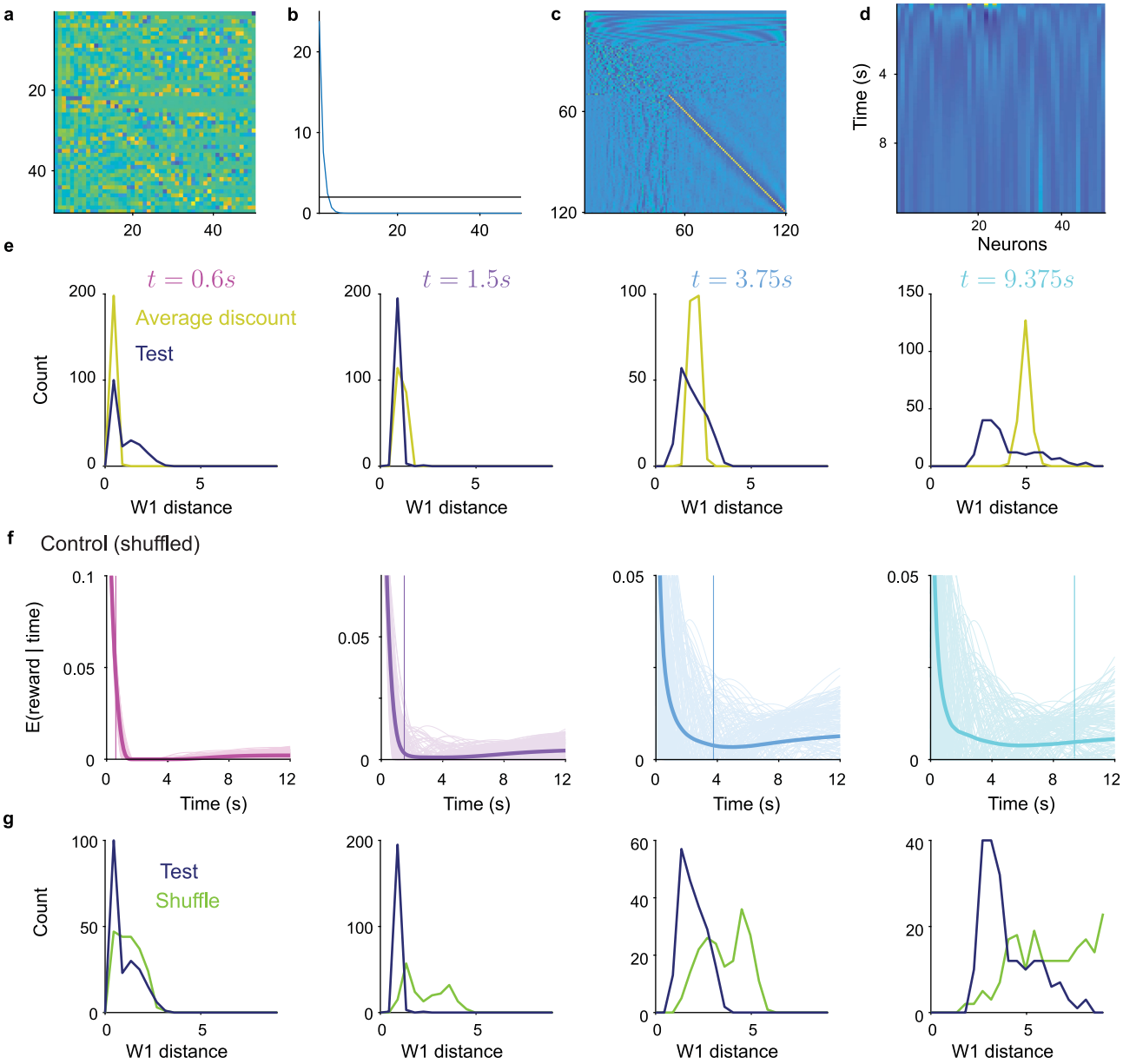
Extended Data Fig. 2 | See next page for caption.

Extended Data Fig. 2 | The myopic learning bias. a, Illustration of the myopic learning bias. Consider a scenario in which the upwards and downwards paths from s and s' are experienced only once, such that at least one path in the small bifurcations is left unexplored. In state s' (blue) the far future is more uncertain than the near future, and in state s (red) the near future is more uncertain than the far future. **b**, When experiencing the upwards and downwards paths only once, there are 4 possible scenarios depending on which path of the corresponding bifurcations is visited. When learning from limited information, the myopic (low γ) and farsighted (high γ) estimates would make different decisions depending on whether the V^{UP} estimate is larger, smaller or approximately equal to V^{DOWN} ($V^{\mathrm{UP}} \approx V^{\mathrm{DOWN}}$ occurs when both estimates have similar magnitudes and some small variations in the precise magnitude of rewards, the prior values or the learning parameters could change whether $V^{\mathrm{UP}} < V^{\mathrm{DOWN}}$ or $V^{\mathrm{UP}} > V^{\mathrm{DOWN}}$). In both s and s', the correct decision is to follow the upwards path (the 'correct' decision is the decision made by a hypothetical RL agent that experiences all possible trajectories an infinite number of times). Below we show the approximate probability that the agent chooses the correct path, if it follows the myopic estimates (low γ) or far-sighted estimates (high γ). Illustration of a mouse in panel **a** and silhouette of a raptor in panels **a**,**b** were adapted from the NIAID NIH BIOART Source. Illustration of a block of cheese in panels **a**,**b**, was adapted from SVG Repo under a CC0 1.0 licence. **c**, Task structure to evaluate the myopic learning bias when uncertainty arises due to stochastic rewards. The three dots collapse 5 transitions between black states. Black states give a small stochastic reward and orange states give a large deterministic reward. **d**, Accuracy at selecting the branch with the large deterministic reward under incomplete learning conditions. At state s (orange), agents with larger discount factors (far-sighted) are more accurate. At state s' (blue), agents with a small discount factor (myopic) are more accurate. Error bars are half s.d. across 10,000 episodes, maximums are highlighted with stars. **e**, Mean performance in this task by the agent in Fig. 1d (see main text and Methods). **f**, Maze to highlight the myopic learning bias in cases where uncertainty arises due to incomplete exploration of the state space. Rewards are indicated with water and fire. An example trajectory is shown with transparent arrows. The red and blue bars to the right denote the states in the Lower and Upper half. **g**, True (grey) and estimated (green and brown) values for the example trajectory on top and shown in panel a. In the x-axis we highlight the starting timestep with s, the timestep when the fire is reached and the timestep when the water is reached. Image of fire in panels **f**,**g** was created by dstore via SVG Repo under a CC0 1.0 licence. Images of water droplet in panels **f**,**g** were adapted from SVG Repo under a CC0 1.0 licence. **h**, Accuracy (y-axis) is measured as the Kendall tau coefficient between the estimate with a specific gamma (x-axis) and the true value function $V_\gamma = 0.99$. Error bars are deviations across 300 sets of sampled trajectories. The red (blue) curve shows average accuracy for the states on the upper (lower) half of the maze, indicated with color lines on panel **a**. **i**, As the sampled number of trajectories increases, the myopic learning bias disappears. **j**, Architecture that learns about multiple timescales as auxiliary tasks. **k**, States are separated according to the agent being close to the goal (blue) or far from the goal (orange). Images in panels **j**,**k** were adapted from Farama Foundation under an MIT licence. **l**, Accuracy of the Q-values in the Lunar Lander environment as a function of their discount factor, estimated as the fraction of concordant state pairs between the empirical value function and the discount specific Q-value estimated by the network. Error bars are s.e.m across 10 trained networks, maximums are highlighted with stars. See Methods for details.

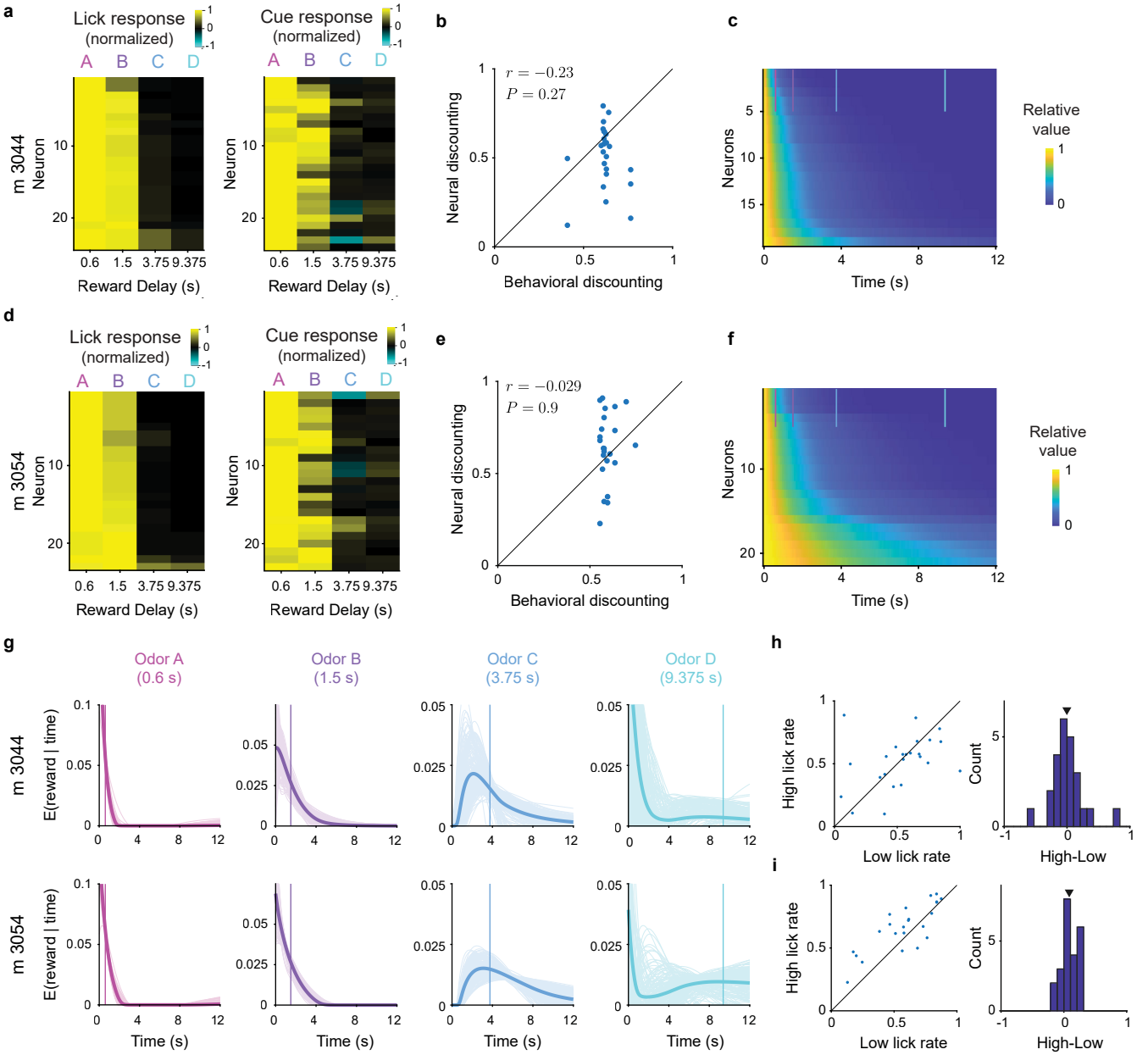


Extended Data Fig. 3 | See next page for caption.

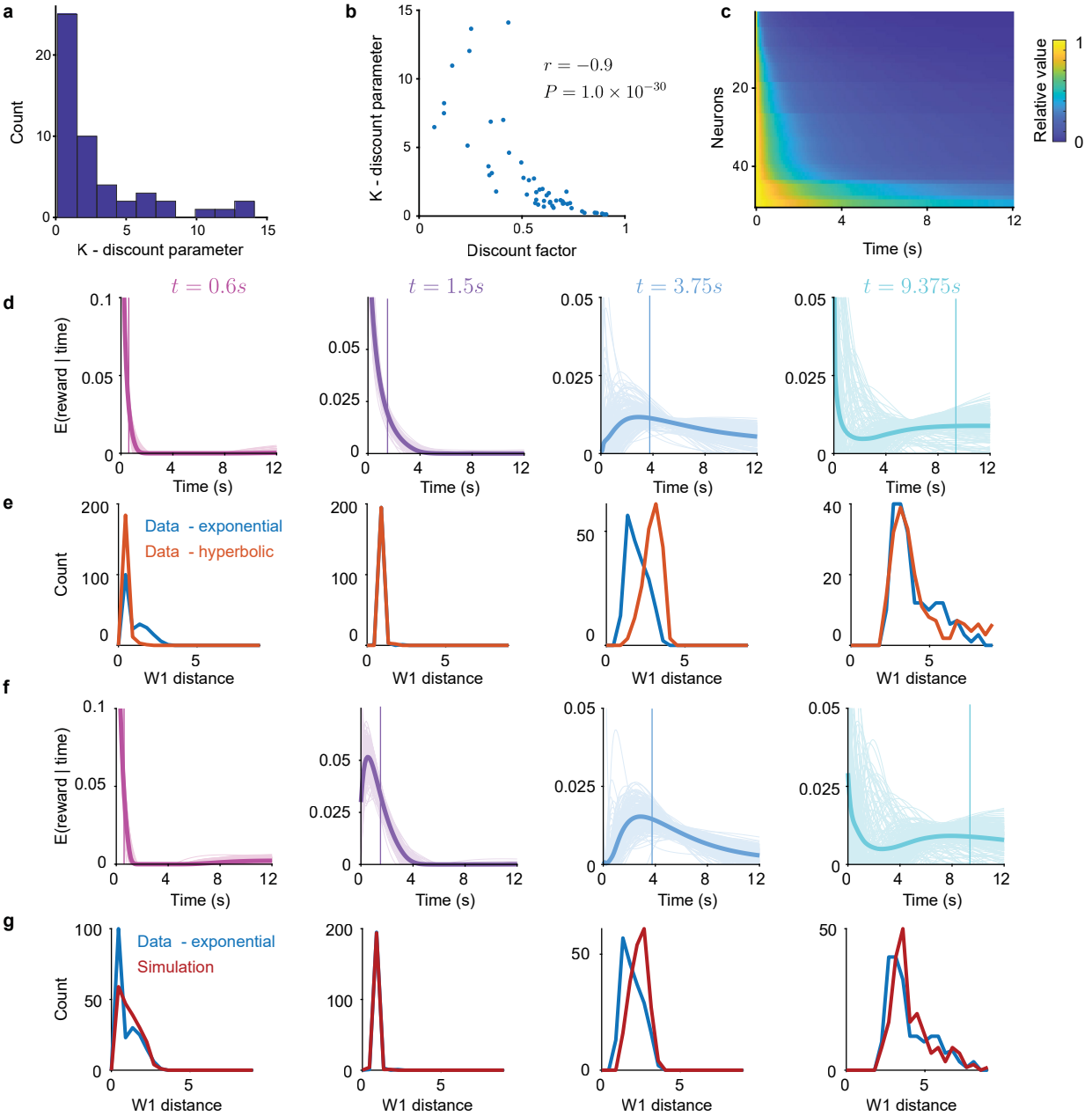
Extended Data Fig. 3 | Single neuron responses and robustness of fit in the cued delay task. a, PSTHs of single selected neurons ($n = 50$) responses to the cues predicting a reward delay of 0.6 s, 1.5 s, 3.75 s, and 9.375 s (from top to bottom). Neurons are sorted by the inferred value of the discount factor γ. Neural responses are normalized by z-scoring each neuron across its activity to all 4 conditions. **b**, PSTHs of single non-selected neurons ($n = 23$) responses to the cues predicting a reward delay of (from top to bottom). Neurons are sorted by the inferred value of the discount factor γ. Neural responses are normalized by z-scoring each neuron across its activity to all 4 conditions. **c**, Variance explained for training vs testing data for the exponential model. For each bootstrap, the variance explained was computed on both the half of the trials used for fitting (train) and the other half of the trials (test). Neurons ($n = 13$) with a negative variance explained on the test data are excluded from the decoding analysis (grey dots). **d**, Same as panel **c** but for the fits for the hyperbolic model. **e**, Mean goodness of fit on held-out data across 100 bootstraps for each selected neuron for the exponential and hyperbolic models. The data lies above the diagonal line suggesting a better fit from the exponential model as shown in Fig. 2f. Error bars indicate 95% confidence interval using bootstrap, see Methods. **f**, The values of the inferred parameters in the exponential model are robust across bootstraps. top row, Inferred value of the parameters across two halves of the trials (single bootstrap) for the gain α, baseline b and discount factor γ, respectively. Bottom row, Distribution across $n = 100$ bootstraps of the Pearson correlations across neurons between the inferred parameter values in the two halves of the trials. Reported mean is the mean correlation across bootstraps and reported p-value is the highest p-value for all the bootstraps for a given parameters assessed via Student's t-test. Distribution of correlations for the gain α (mean = 0.84, $P < 1 \times 10^{-20}$), baseline b (v, mean = 0.9, $P < 1.0 \times 10^{-32}$) and discount factor γ (vi, mean = 0.93, $P < 1.0 \times 10^{-46}$). **g**, Same as panel **f** (lower row) but for the hyperbolic model with distribution of correlations for the gain α (mean = 0.86, $P < 1 \times 10^{-26}$), baseline b (v, mean = 0.88, $P < 1.0 \times 10^{-28}$) and shape parameter k (vi, mean = 0.76, $P < 1.0 \times 10^{-11}$). **h**, Same as panel **f** (lower row) but for the exponential model simulated responses with distribution of correlations for the gain α (mean = 0.86, $P < 1.0 \times 10^{-10}$), baseline b (v, mean = 0.88, $P < 1.0 \times 10^{-24}$) and discount factor γ (vi, mean = 0.76, $P < 1.0 \times 10^{-26}$). Note that the distributions of inferred parameters are in a similar range than the fits to the data suggesting that trial numbers constrain the accuracy of parameter estimation. **i**, similar to the panel **f** (lower row) for the neurons recorded in mouse m3044, showing that across bootstraps, the estimates are consistent for the gain α (mean = 0.64, $P < 0.05$ for 88/100 bootstraps, light blue $P > 0.05$, dark blue $P < 0.05$), baseline b (mean = 0.86, $P < 0.012$) and discount factor γ (mean = 0.72, $P < 0.05$ for 97/100 bootstraps, light blue $P > 0.05$, dark blue $P < 0.05$). **j**, same as panel **f** (lower row) for the neurons recorded in mouse m3054. The estimates are consistent for the gain α (mean = 0.90, $P < 0.0048$), baseline b (mean = 0.93, $P < 1.0 \times 10^{-5}$) and discount factor γ (mean = 0.79, $P < 0.0069$).



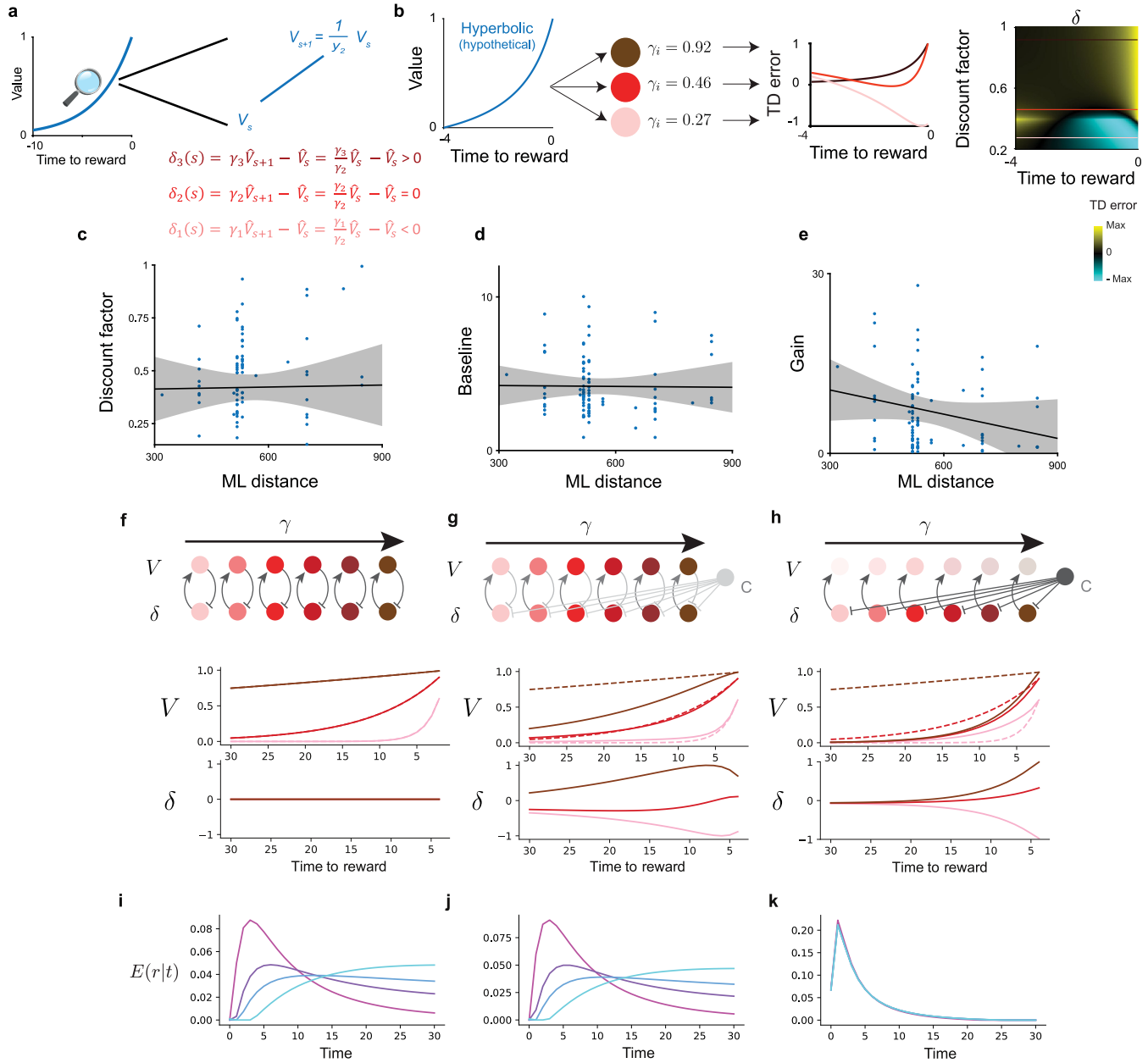
Extended Data Fig. 4 | Decoding reward timing using the regularized pseudo-inverse of the discount matrix. (a-c), Singular value decomposition (SVD) of the discount matrix. **a**, left singular vectors (in the neuron space). **b**, Singular values. The black line at 2 indicates the values of the regularization term α. **c**, right singular vectors (in the time space). **d**, Decoding matrix based on the regularized pseudo-inverse. **e**, Distribution of 1-wassertein distances between the reward timing and the predicted reward timing from the decoding on the test data from exponential fits (shown in Fig. 2k, top row) and on the average exponential model (shown in Fig. 2k, bottom row). Decoding is better for the exponential model from Fig. 2 than the average exponential model except for the shortest delay ($P(t = 0.6\text{ s}) = 1$, $P(t = 1.5\text{ s}) < 1.0 \times 10^{-31}$, $P(t = 3.75) = 0.0135$, $P(t = 9.375\text{ s}) < 1.0 \times 10^{-14}$), one-tailed Wilcoxon signed rank test, see Methods). **f**, The ability to decode the timing of expected future reward is not due to a general property of the discounting matrix and collapses if we randomize the identity of the cue responses (see Methods). **g**, Distribution of 1-Wassertein distances between the reward timing and the predicted reward timing from the decoding on the test data exponential fits (shown in Fig. 2k, top row) and on the shuffled data (shown in panel **f**). The prediction from the test data are better predictions (smaller 1-Wasserstein distance) than shuffled data ($P = 1.2 \times 10^{-4}$ for 0.6 s reward delay, $P < 1.0 \times 10^{-20}$ for the other delays, one-tailed Wilcoxon signed rank test, see Methods).



Extended Data Fig. 5 | Comparing behavioral and neural discounting and decoding reward timing in single animals. (a-c): mouse 3044. **a**, left panel. Normalized lick responses to the cues predicting reward delays across the population. For each neuron, the response was normalized to the highest response across the 4 possible delays. Neurons are sorted by the inferred behavioral discount factor. Right panel: Normalized neural responses to the cues predicting reward delays across the population (sorted by the behavioral discount factor). **b**, The behavioral and neural discount factors are not correlated ($r = -0.29$, $P = 0.27$, Spearman's rank correlation, two-tailed Student's t-test). **c**, Discount matrix for the neurons recorded in mouse 3044. This is the matrix used for decoding in panel g, top row. **(d-f)**: same as panels (**a-c**) for mouse 3054. **e**, The behavioral and neural discount factors are not correlated in mouse 3054 ($r = -0.029$, $P = 0.9$, Spearman's rank correlation, two-tailed Student's t-test). **f**, Discount matrix for the neurons recorded in mouse 3054. This is the matrix used for decoding in panel **g**, bottom row. **g**, Decoding of reward timing at the single animal level for mouse 3044 (top row) and mouse 3054 (bottom row). The decoding is present but slightly less accurate as expected from the smaller number of neurons. **h**, discount factor inferred for neurons in mouse m3044 when dividing trials between low and high anticipatory lick rate. left panel, scatter plot of the value across neurons. right panel, the distribution across neurons of differences in inferred discount across the two conditions is not significant (mean = −0.0024, $P = 0.96$, two-tailed Student's t-test). **i**, Same as panel **h** for mouse m3054. The difference in inferred value between low and high lick rate is significant (mean = 0.086, $P = 0.0062$, two-tailed Student's t-test) but the mean effect is small compared to the standard deviation of inferred discount factors across neurons (s.d. = 0.19 for neurons in m3054).

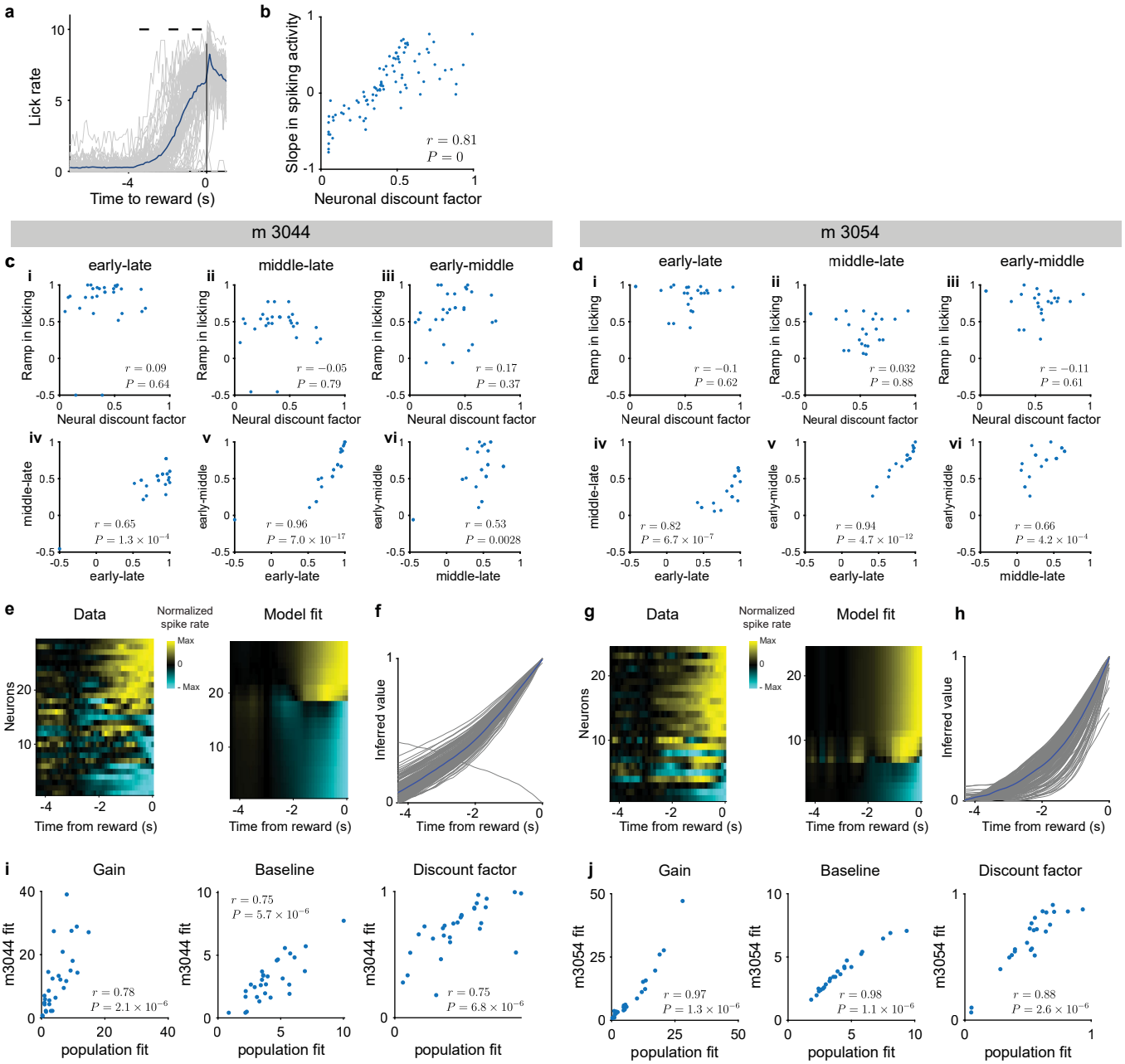


Extended Data Fig. 6 | Decoding reward timing from the hyperbolic model and exponential model simulations. a, Distribution of the inferred discount parameter k across the neurons. **b**, Correlation between the discount factor inferred in the exponential model of the discount parameter k from the hyperbolic model ($r = -0.9$, $P < 1.0 \times 10^{-30}$, Student's t-test). Note the in the hyperbolic model a larger value of k implies faster discounting hence the negative correlation. **c**, Discount matrix for the hyperbolic model. For each neuron we plot the relative value of future events given its inferred discount parameter. Neurons are sorted by decreasing estimated value of the discount parameter. **d**, Decoded subjective expected timing of future reward $E(r|t)$ using the discount matrix from the hyperbolic model (see Methods). **e**, Distribution of 1-Wassertein distances between the reward timing and the predicted reward timing from the decoding on the test data with the exponential model (shown in Fig. 2k, top row) and on the test data with the hyperbolic model (shown in **d**). Decoding is better for the exponential model from Fig. 2 than the hyperbolic model except for the shortest delay ($P(t = 0.6\text{ s}) = 1$, $P(t = 1.5\text{ s}) < 1.0 \times 10^{-31}$, $P(t = 3.75) < 1.0 \times 10^{-33}$, $P(t = 9.375\text{ s}) < 1.0 \times 10^{-3}$), one-tailed Wilcoxon signed rank test, see Methods). **f**, Decoded subjective expected timing of future reward $E(r|t)$ using simulated data based on the parameters of the exponential model (see Methods). **g**, Distribution of 1-Wassertein distances between the reward timing and the predicted reward timing from the decoding on the test data from exponential fits (shown in Fig. 2k, top row) and on the simulated data from the parameters of the exponential fits (shown in **f**). Decoding is marginally better for the data predictions ($P(t = 0.6\text{ s}) = 0.002$, $P(t = 1.5\text{ s}) = 0.999$, $P(t = 3.75) < 1 \times 10^{-12}$, $P(t = 9.375\text{ s}) = 0.027$), one-tailed Wilcoxon signed rank test, see Methods), suggesting that decoding accuracy is limited by the number of trials.



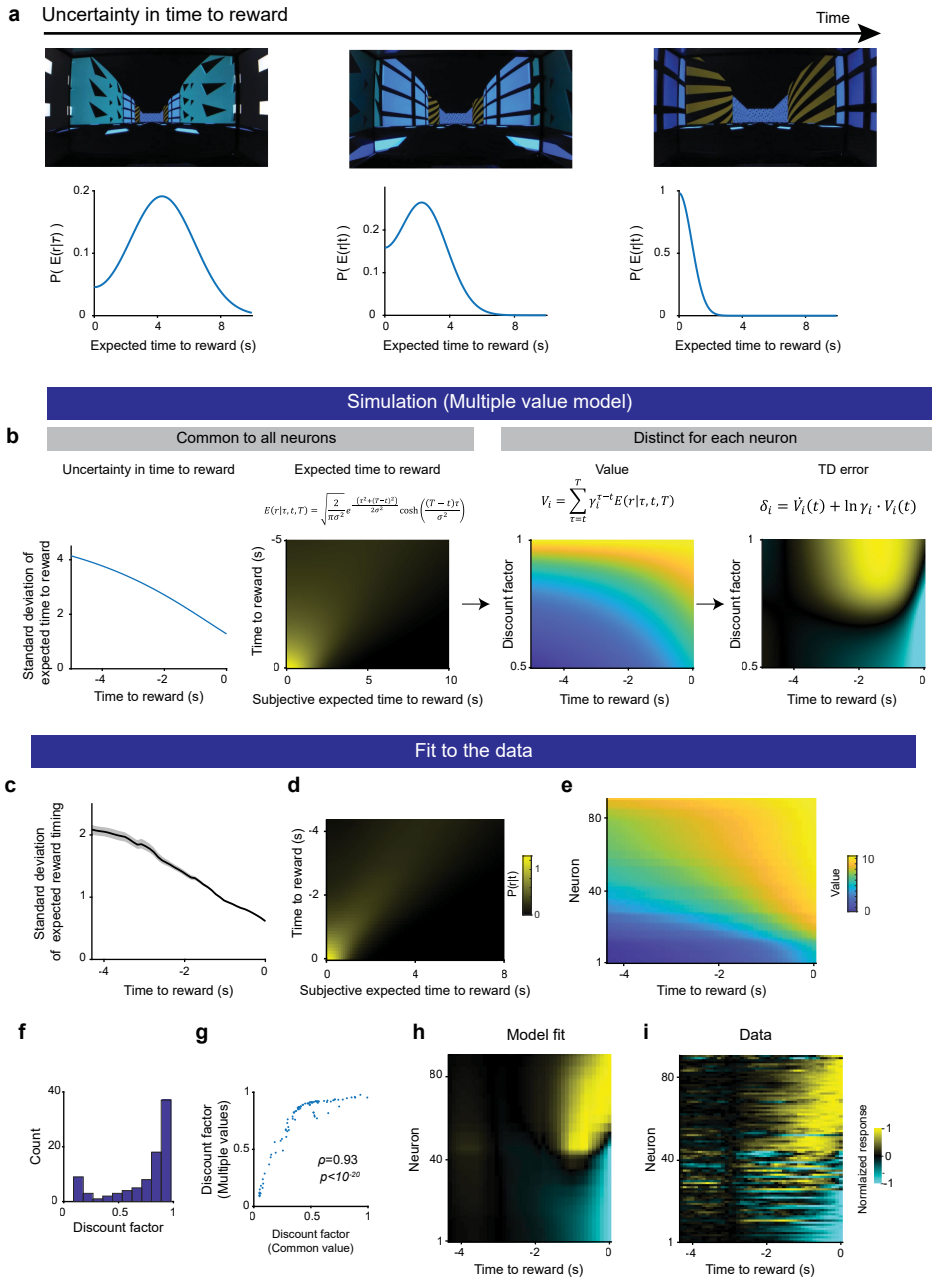
Extended Data Fig. 7 | See next page for caption.

Extended Data Fig. 7 | Ramping, discounting, anatomy and distributed RL models. a, Ramping in the prediction error signal is controlled by the relative contribution of value increases and discounting. If the value increase (middle) exactly matches the discounting, there is no prediction error (middle equation, right). If the discounting is smaller than the value increase (large discount factor) then there is a positive TD error (top equation, right). If the discounting is larger (small discount factor) than the value increase then there a negative TD error (bottom equation, right). A single timescale agent with no state uncertainty will learn an exponential value function but if there is state uncertainty (see ref. 69) or the global value function arises from combining the contribution of single-timescale agents then the value function is likely t be non-exponential. Image of a magnifying glass was created by googlefonts via SVG Repo under an Apache Licence. **b**, Intuition for diversity of ramping with a hyperbolic value function. Agents with a small discount factor exhibit a monotonic downward ramp (pink), while those with a large discount factor exhibit a monotonic upward ramp (brown). Agents with an intermediary discount factor tend to exhibit a downward then upward ramp. The hyperbolic value function gets increasingly convex as the reward approaches, so an increasing fraction of the agents have a positive prediction error as they approach the reward. **c**, The discount factor inferred in the VR task is not correlated with the medio-lateral (ML) position of the implant (Pearson's $r = 0.015$, $P = 0.89$, two-tailed Student's t-test). **d**, The baseline parameter inferred in the VR task is not correlated with the medio-lateral (ML) position of the implant (Pearson's $r = -0.011$, $P = 0.92$, two-tailed Student's t-test). **e**, The inferred gain in the VR task reduces with increasing medio-lateral (ML) position but the effect does not reach significance (Pearson's $r = -0.19$, $P = 0.069$, two-tailed Student's t-test). In panels c-e, the line correspond to the best fit linear regression and the uncertainty shading represents 95% confidence interval on a linear regression fit. **f-h**. Ramping in the reward prediction error with mixing in distributed RL models. Inferred value functions (V) and RPEs (δ) for the mixed RL model as a function of the common value function sharing-parameter λ, in a linear MDP of 30 steps (x-axis in the plots) with a deterministic reward equal to 1 in the last step and 0 everywhere else. Plots are shown after learning has empirically stabilized (after 3,000 TD-learning iterations with a learning rate of 0.1). The dashed value function is the exponential value function without common value sharing ($\lambda = 0$), which would lead to a flat RPE equal to 0 at every state. The actual value functions (solid lines) are not purely exponential, and thus lead to ramping RPEs. **f**, Circuit model in which each value estimation and their corresponding prediction error are part completely independent loops ($\lambda = 0$). At convergence, there is no more prediction error in the reward anticipation period (bottom row). **g**, Circuit model in which the prediction error for each dopamine neuron is influenced by both the independent value signal and the shared a common value signal (C, $\lambda = 0.1$). The dashed line indicate the value function corresponding to completely separate loops, and the solid function the actual value function due to the influence of the common value signal. The difference between them leads to ramping in the reward prediction error signals (bottom row). **h**, Circuit model with a strong influence of the common value signal ($\lambda = 1$) which also leads to ramping in the reward prediction error signals. See Methods for details. **i**, Decoded reward times for 4 experimental conditions with rewards at times 5, 10, 15 and 30 (pink to cyan), by applying a regularized inverse Laplace decoder (analogous to the one used in Fig. 2 of the main text) to the values at the moment of the cue, under the model without mixing $\lambda = 0$. **j**, Same as (i) but using a mixing factor of $\lambda = 0.1$. The small mixing factor does not affect the quality of the temporal decoding, while creating a ramping reward prediction error (panel g). Therefore, a small mixing factor constitutes a common model that can qualitatively account for the two tasks studied in the paper. **k**, Same as (i) but using a mixing factor of $\lambda = 1$. Using a fully shared value function the relative differences between discount factors disappear, so temporal decoding is no longer possible.

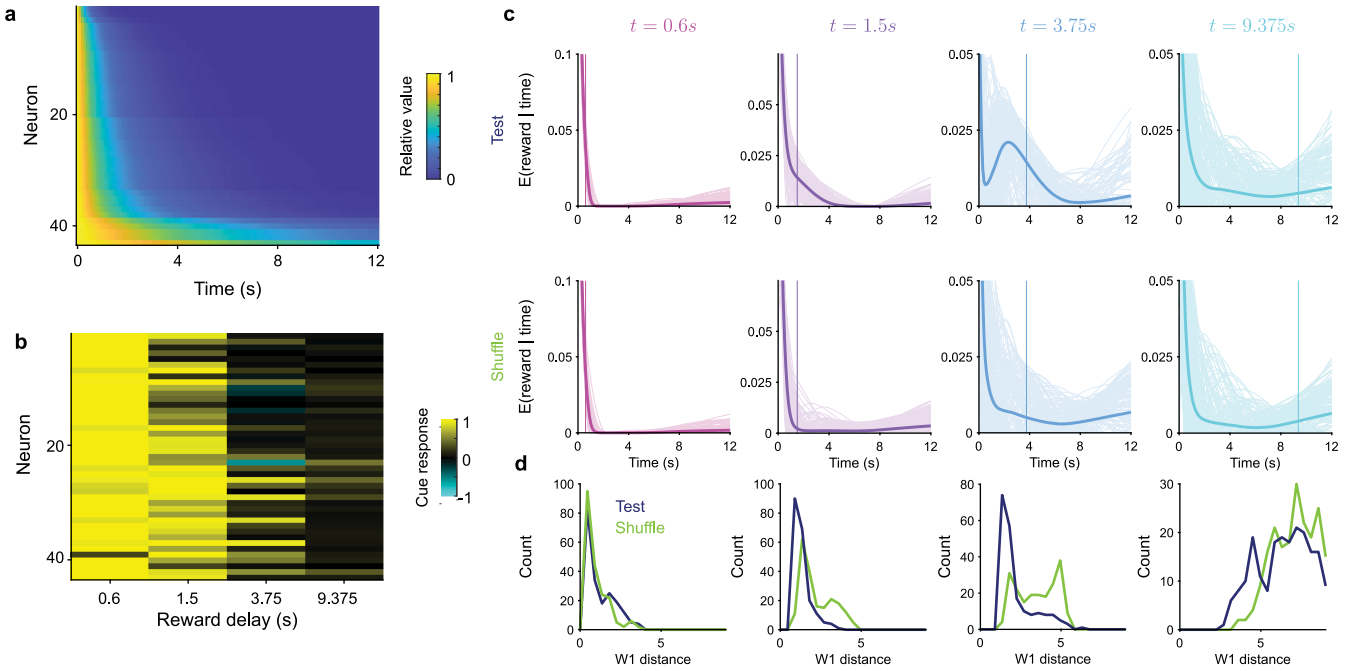


Extended Data Fig. 8 | See next page for caption.

Extended Data Fig. 8 | Behavioral and neural discounting at the single animal. a, Time course of the lick rate in the VR task as mice approach the reward location. gray line, lick rate for individual neurons, blue line, mean lick rate. The three black lines on top indicate the three windows used to compute early, middle and late lick rate in the analysis presented in panels c-d. **b**, The inferred discount factor and the slope in spiking activity (see Fig. 3b) are strongly correlated ($r = 0.81$, $P = 0$, Spearman rank correlation, two-tailed Student's t-test) suggesting that slope is a good measure of discounting. **c**, Correlations of measures of behavioral and neural discounting for mouse m3044 (Spearman rank correlation, two-tailed Student's t-test). **i-iii**: the neural discount factor and the ramp in licking activity is not correlated irrespective of the window used to compute the ramp in licking activity when using the following windows to compute ramping activity in the lick rate: **i**, modulation between the late and early window, $r = 0.09$, $P = 0.64$. **ii**, modulation between the late and middle windows, $r = -0.05$, $P = 0.79$. **iii**, modulation between the early and middle windows, $r = 0.17$, $P = 0.37$. **iv-vi**: The measures of ramping in licking activity are strongly correlated to each other: **i**, correlation between the late-middle and late-early modulation measures, $r = 0.65$, $P = 1.3 \times 10^{-4}$. **ii**, correlation between the middle-early and late-early modulation measures, $r = 0.96$, $P = 7 \times 10^{-17}$. **iii**, correlation between the middle-early and late-middle modulation measures, $r = 0.53$, $P = 0.0028$. **d**, Correlations of measures of behavioral and neural discounting for mouse m3054 (Spearman rank correlation, two-tailed Student's t-test). **i-iii**: the neural discount factor and the ramp in licking activity is not correlated irrespective of the window used to compute the ramp in licking activity when using the following windows to compute ramping activity in the lick rate: **i**, modulation between the late and early window, $r = -0.1$, $P = 0.62$. **ii**, modulation between the late and middle windows, $r = 0.032$, $P = 0.88$. **iii**, modulation between the early and middle windows, $r = -0.11$, $P = 0.61$. **iv-vi**: The measures of ramping in licking activity are strongly correlated to each other: **i**, correlation between the late-middle and late-early modulation measures, $r = 0.82$, $P = 6.7 \times 10^{-7}$. **ii**, correlation between the middle-early and late-early modulation measures, $r = 0.94$, $P = 4.7 \times 10^{-12}$. **iii**, correlation between the middle-early and late-middle modulation measures, $r = 0.66$, $P = 4.2 \times 10^{-4}$. **e**, The VR model fits (right panel) to m3044 neurons alone captures the diversity of ramping activity observed across single neurons (left panel). **f**, Inferred value function for m3044. Thin gray line, individual bootstrap fits. Blue line, mean value fit. **g**, The VR model fits (right panel) to m3054 neurons alone captures the diversity of ramping activity observed across single neurons (left panel). **h**, Inferred value function for m3054. Thin gray line, individual bootstrap fits. Blue line, mean value fit. **i**, The inferred parameter values between the fit for m3044 and the full population fit are strongly correlated (Spearman rank correlation, two-tailed Student's t-test) for the gain parameter (left panel, $r = 0.78$, $P = 2.1 \times 10^{-6}$), the baseline parameter (middle panel, $r = 0.75$, $P = 5.7 \times 10^{-6}$) and the discount factor (right panel, $r = 0.75$, $P = 6.8 \times 10^{-6}$). **j**, The inferred parameter values between the fit for m3054 and the full population fit are strongly correlated (Spearman rank correlation, two-tailed Student's t-test) for the gain parameter (left panel, $r = 0.97$, $P = 1.3 \times 10^{-6}$), the baseline parameter (middle panel, $r = 0.98$, $P = 1.1 \times 10^{-6}$) and the discount factor (right panel, $r = 0.88$, $P = 2.6 \times 10^{-6}$). All reported correlations are Spearman rank correlations.



Extended Data Fig. 9 | Discounting heterogeneity explains ramping diversity in a common reward expectation model. a, Uncertainty in reward timing reduces as mice approach the reward zone. Not only does the mean expected reward time reduces but the standard deviation of the estimate also reduces. Distribution in the bottom row from fitted data (see panels **c-e**). **b**, A model where each neuron contributes to its individual value function but share a common reward expectation predicts ramping heterogeneity across neurons. Left panel, as mice approach reward, the uncertainty, quantified by the standard deviation, of reward timing reduces. 2nd panel from left, The Expectation of reward timing takes the form of a folded normal distribution. As the mice approach the reward there is a reduction of both the mean and the standard deviation of the expected reward timing distribution. 3rd panel from left, each neuron computes a distinct value function given their individual discount factor and the common expected reward timing distribution with. Right panel, The diverse value functions across neurons lead to ramping heterogeneity across neurons in the reward prediction error. (see Methods 'Common Reward Expectation model'). **c**, The inferred standard deviation of the reward expectation model reduces as a function of time to reward. Line indicates the mean inferred standard deviation and the shading indicates the standard error of the mean over 100 bootstraps. **d**, Expected timing of the reward as a function of true time to reward. As the mice approach the reward not only does the mean expected time to reward reduces but the uncertainty of the reward timing captured by the standard deviation shown in **c** also reduces. This effect leads to increasingly convex value functions that lead to the observed ramps in dopamine neuron activity. **e**, Value function for each individual neuron (same order as in **h-i**). **f**, Distribution of inferred discount factors under the common reward expectation model. **g**, Although the range of discount factor between the fits from the common value (x axis) and common reward expectation (y axis) models differs, the inferred discount factors are strongly correlated for single neurons (Spearman's $\rho = 0.93$, $P < 1.0 \times 10^{-20}$, two-tailed Student's t-test). **h**, Predicted ramping activity from the model fits under the common reward expectation model. **i**, Diversity of ramping activity across single neurons as mice approach reward (aligned by inferred discount factor in the common reward expectation model).



e

Animal	Neurons recorded in cued delay task	Neurons recorded per session on cued delay task	Neurons recorded in VR task	Neurons recorded per session in VR task	Neurons recorded in both tasks	Neurons recorded per session in both tasks
m3014	0	0	2 (2)	1(1)	0	0
m3015	0	0	12 (11)	1.7±0.95(1.58±0.79)	0	0
m3017	0	0	1 (1)	1(1)	0	0
m3021	0	0	2 (2)	1(1)	0	0
m3044	36 (19)	2.4±0.83 (1.6±1)	37 (29)	1.9±0.91(1.71±0.77)	32 (15)	2.1±0.83(1.5±0.71)
m3045	7 (6)	1.2±0.41 (1.2±0.45)	13 (11)	1.2±0.4(1.1±0.32)	7 (5)	1.2±0.41(1.2±0.5)
m3047	1 (1)	1 (1)	1 (1)	1(1)	1 (1)	1(1)
m3048	3 (0)	1(0)	6 (6)	1(1)	3 (0)	1(0)
m3049	4 (0)	1.3±0.58 (0)	9 (0)	1.3±0.49(0)	4 (0)	1.3±0.58(0)
m3050	1 (0)	1 (0)	5 (0)	1(0)	1 (0)	1(0)
m3051	0	0	6 (0)	1(0)	0	0
m3053	3 (3)	1(1)	3 (3)	1(1)	2 (2)	1(1)
m3054	23 (21)	1.9±1.2 (1.75±0.97)	24 (24)	1.8±1.1(1.8±1.1)	22 (20)	1.8±1.0(1.7±0.78)

Extended Data Fig. 10 | Decoding reward timing in the cued delayed reward task using parameters inferred in the VR task and details of recordings. **a**, Discount matrix computed using the parameters inferred in the VR tasks for neurons recorded across both tasks and used in the cross-task decoding. **b**, Dopamine neurons cue responses in the cued delay task. Neurons are aligned as in **a** according to increasing discount factor inferred in the VR task. **c**, Top row: Decoded reward timing using discount factors inferred in the VR task. Bottom row: The ability to decode reward timing is lost when shuffling the identities of the cue responses. **d**, Except for the shortest delay, decoded reward timing is more accurate than shuffle as measured by the 1-Wassertsein distance ($P_{t=0.6s}=1, P_{t=1.5s}<1.1\times10^{-20}, P_{t=3.75s}<3.8\times10^{-20}, P_{t=9.375s}<2.9\times10^{-5}$). **e**, Breakdown of the number of recorded neurons per animal and task. The numbers in parenthesis indicate the number of neurons included in the analysis. ± indicates standard deviation across sessions. The maximum number of neurons recorded in a single session was 4 in both the cued delay task and the VR task.

nature portfolio

Corresponding author(s): Paul Masset, Alexandre Pouget and Naoshige Uchida

Last updated by author(s): Feb 16, 2025

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	The data was collected as part of a prior study: Kim, HyungGoo R., et al. "A unified framework for dopamine signals across timescales." Cell 183.6 (2020): 1600-1616. We point the reader to the paper in the methods.
Data analysis	Analysis of behavioral and neural data was performed using custom code in MATLAB (MathWorks, r2021a). Data analysis code will be published on Mendeleley data: ttps://doi.org/10.17632/tc43t3s7c5.1 (A preview link is available here: https://data.mendeley.com/preview/tc43t3s7c5?a=32dcede7-b274-4b1e-af5a-18549236d582 and we will publish the repository once a DOI has been assigned to the paper) The code used for simulations can be found at https://github.com/pablotano8/multi_timescale_RL .

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

Raw data is deposited in the repository from the describing the data collection and is available at (https://dandiarchive.org/dandiset/000251)
Curated data will be published on Mendeleley data: ttps://doi.org/10.17632/tc43t3s7c5.1 (A preview link is available here: https://data.mendeley.com/preview/tc43t3s7c5?a=32dcede7-b274-4b1e-af5a-18549236d582 and we will publish the repository once a DOI has been assigned to the paper)

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	This study does not involve human participants, their data or biological materials
Reporting on race, ethnicity, or other socially relevant groupings	This study does not involve human participants, their data or biological materials
Population characteristics	See above
Recruitment	This study does not involve human participants, their data or biological materials
Ethics oversight	This study does not involve human participants, their data or biological materials

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	Data was collected as part of a previous study (Kim, HyungGoo R., et al. "A unified framework for dopamine signals across timescales." Cell 183.6 (2020): 1600-1616.). The goal was to obtain an as exhaustive as possible survey of the function of individual dopamine neurons. Experimental data collection was designed to achieve maximal statistical power within the available time and financial resources to perform experimental data collection. We refer to the table in Extended Data Fig 10 for furtehr details.
Data exclusions	Some neurons are excluded from certain analysis as described in the methods. These include neurons estimated to be located outside of the Ventral Tegmental area and neurons whose firing rates were too low to perform reliable analysis.
Replication	Data was collected from several animals. We describe in the methods the analysis used to ensure the reliability of the inferred parameters. The population level analysis was performed at the single animal level on the two animals with sufficient number of recorded neurons as described in the manuscript.
Randomization	Allocation between the virtual reality and cued delayed task was based on the timing of the experimental data collection.
Blinding	Investigators were not blinded to the identity of the mice. This was required for animal care considerations and the study did not involve controls that required blinding. Analysis was performed after data collection so investigators were blind to inferred neuron parameters during data collection.

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	We used a total of 13 adult C57/BL6J DAT-Cre male mice. Mice were backcrossed for over 5 generations with C57/BL6J mice. All procedures were performed in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals and approved by the Harvard Animal Care and Use Committee. We refer to the paper that describes data collection for more details: Kim, HyungGoo R., et al. "A unified framework for dopamine signals across timescales." Cell 183.6 (2020): 1600-1616.
Wild animals	No wild animals were used in this study.
Reporting on sex	Sex was not considered in the study design and all data presented was collected in male mice only.
Field-collected samples	No field-collected samples were used in this study.
Ethics oversight	All procedures were performed in accordance with the National Institutes of Health Guide for the Care and Use of Laboratory Animals and approved by the Harvard Animal Care and Use Committee.

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

Plants

Seed stocks	*Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.*
Novel plant genotypes	*Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.*
Authentication	*Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.*