nature neuroscience

SUPPLEMENTARY INFORMATION

https://doi.org/10.1038/s41593-018-0147-8

In the format provided by the authors and unedited.

Prefrontal cortex as a meta-reinforcement learning system

Jane X. Wang[1,5], Zeb Kurth-Nelson[1,2,5], Dharshan Kumaran[1,3], Dhruva Tirumala[1], Hubert Soyer[1], Joel Z. Leibo[1], Demis Hassabis[1,4] and Matthew Botvinick[1,4]*

[1]DeepMind, London, UK. [2]Max Planck-UCL Centre for Computational Psychiatry and Ageing Research, University College London, London, UK. [3]Institute of Cognitive Neuroscience, University College London, London, UK. [4]Gatsby Computational Neuroscience Unit, University College London, London, UK. [5]These authors contributed equally: Jane X. Wang and Zeb Kurth-Nelson. *e-mail: botvinick@google.com

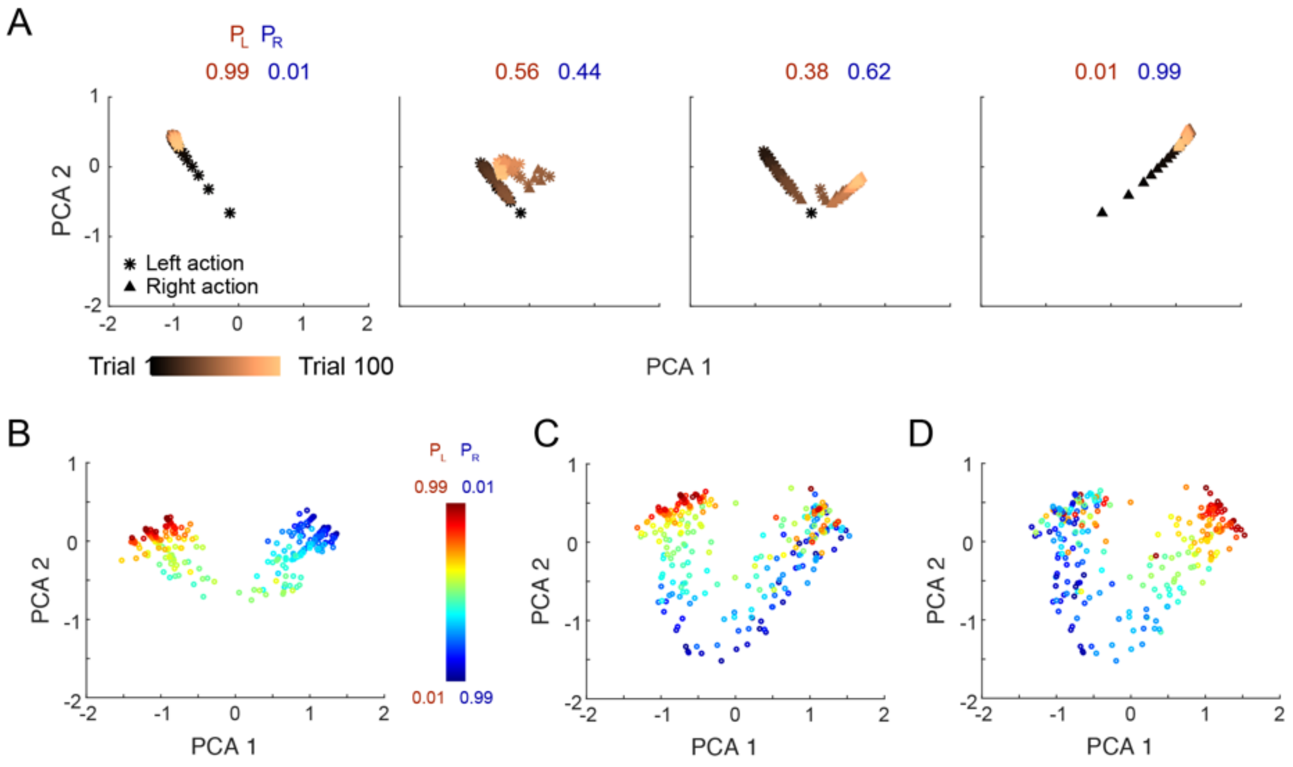
Algorithm 1 Advantage actor-critic - pseudocode

```
// Assume parameter vectors $\theta$ and $\theta_v$
Initialize step counter $t \leftarrow 1$
Initialize episode counter $E \leftarrow 1$
repeat
    Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
    $t_{start} = t$
    Get state $s_t$
    repeat
        Perform $a_t$ according to policy $\pi(a_t|s_t;\theta)$
        Receive reward $r_t$ and new state $s_{t+1}$
        $t \leftarrow t + 1$
    until terminal $s_t$ or $t - t_{start} == t_{max}$
    $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta_v) & \text{for non-terminal } s_t \text{ //Bootstrap from last state} \end{cases}$
    for $i \in \{t-1, \ldots, t_{start}\}$ do
        $R \leftarrow r_i + \gamma R$
        Accumulate gradients wrt $\theta$: $d\theta \leftarrow d\theta + \nabla_\theta \log \pi(a_i|s_i;\theta)(R - V(s_i;\theta_v)) + \beta_e \partial H(\pi(a_i|s_i;\theta))/\partial\theta$
        Accumulate gradients wrt $\theta_v$: $d\theta_v \leftarrow d\theta_v + \beta_v(R - V(s_i;\theta_v))(\partial V(s_i;\theta_v)/\partial\theta_v)$
    end for
    Perform update of $\theta$ using $d\theta$ and of $\theta_v$ using $d\theta_v$.
    $E \leftarrow E + 1$
until $E > E_{max}$
```

Supplementary Figure 1

Pseudo-code for advantage actor-critic algorithm.

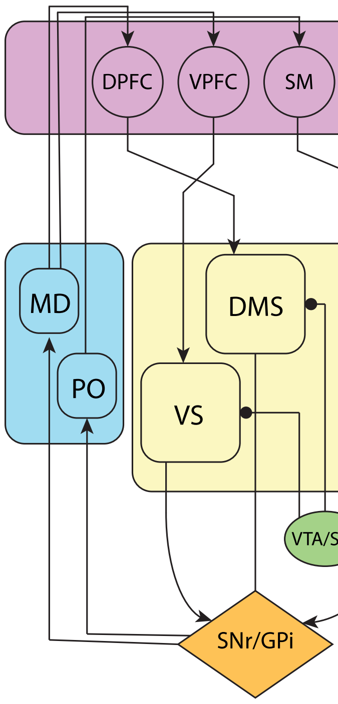
-



Supplementary Figure 2

Visualization of RNN activations in illustrative example (bandit task).

A) Evolution of RNN activation pattern during individual trials while testing on correlated bandits, after training with independent arm parameters. Format analogous to Figure 1E in the main text. P_L= probability of reward for action *left*. P_R= probability of reward for action *right*. Scatter points depict the first two principal components of the RNN activation (LSTM output) vector across steps of the bandit task, drawn from single trials across a range of randomly sampled payoff parameter settings. As evidence accrues, the activation pattern follows a coherent path in one of two directions, corresponding to the arms of the bandit. Interestingly, in more difficult discrimination problems (smaller gap between two arm reward probabilities; middle two columns, also see Figure 1E), activity sometimes appears to initially move toward one extreme of the manifold in early trials, then later reverses when late-coming observations contradict initial ones, to end up at other extremity by the end of the episode (see Ito and Doya, *PLoS Comput Biol, 11,* e1004540, 2015), which posits a related mechanism, though without t discussing its origins). **B**) RNN activity patterns from step 100 (episode completion) while testing on correlated bandits, across a range of payoff parameters. Format analogous to Figure 1F. In contrast to Figure 1F, in which the network is trained on linked arm parameters and thus the activity pattern traces out a one-dimensional manifold, the activity pattern after having trained on independent bandits is correspondingly more complex. **C**) RNN activity patterns when testing on independent bandits using the same network as in A and B, colored according to the left arm probability of reward (red = higher). **D**) same data as C, but colored according to the right arm probability of reward PR (red = higher). For all plots, analyses were done for 300 evaluation episodes, each consisting of 100 trials, performed by 1 fully trained network.



Supplementary Figure 3

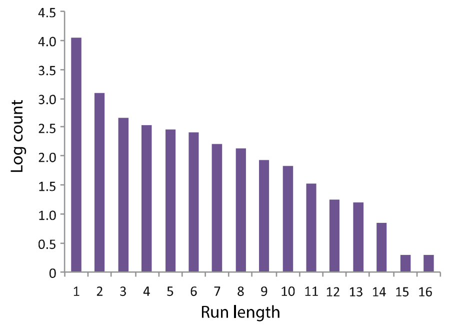
Cortico–basal ganglia–thalamic loops.

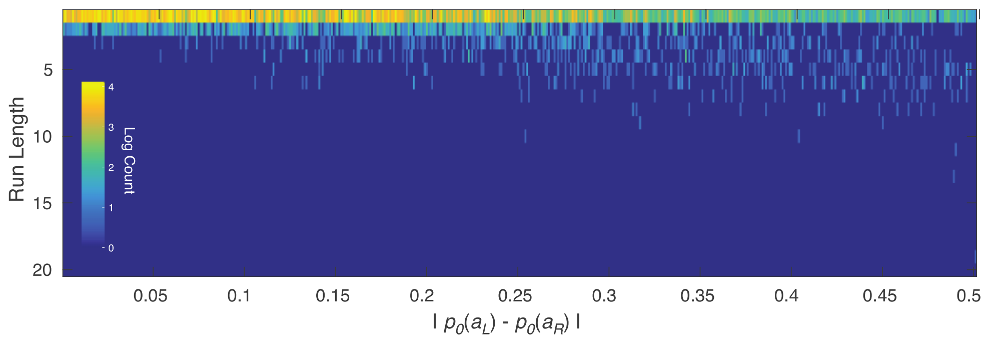
As noted in the main paper, the recurrent neural system to which we refer as the prefrontal network includes not only the PFC itself, including orbitofrontal, ventromedial frontal, cingulate, dorsolateral prefrontal and frontopolar regions, but also the subcortical structures to which it most strongly connects, including ventral striatum, dorsomedial striatum (or its primate homologue), and mediodorsal thalamus. The recurrent connectivity that we assume is thus not only inherent within the prefrontal cortex itself but also spans parallel cortico-basal ganglia-thalamic loops. The figure diagrams a standard model of these loops, based on, based on Glimcher, P. W., & Fehr, E. (Eds.). (2013). *Neuroeconomics: Decision making and the brain*. Academic Press, page 378. *DPFC*: Dorsal prefrontal cortex, including dorsolateral prefrontal and cingulate cortices. *VPFC*: Ventral prefrontal cortex, including ventromedial and orbitofrontal prefrontal cortices. *SM*: (Somato-) sensorimotor cortex. *VS*: Ventral striatum. *DMS*: Dorsomedial striatum. *DLS*: Dorsolateral striatum. *VTA*/*SNc*: Ventral tegmental area and substantia nigra pars compacta, with rounded arrowheads indicating dopaminergic projections. *SNr/GPi*: Substantia nigra pars reticulate/Globus pallidus pars interna. *MD*: Mediodorsal thalamus. *PO*: Posterior thalamus. The circuit running through DPFC has been referred to as the "associative loop"; the circuit running through the VPFC as the "limbic loop"; and the circuit running through SM as the "sensorimotor loop" (Haber, S., *Neuroscience, 282*, 248-257, 2014).

One way of organizing the functions of the various regions of PFC and associated sectors of the striatum is by reference to the actor-critic architecture introduced in RL research, with ventral regions performing the critic role (computing estimates of the value function) and dorsal regions performing the role of the actor (implementing the policy; see Botvinick, M. M., Niv, Y. & Barto, A. C., *Cognition, 113*, 262-280, 2009, and Joel, D., Niv, Y. & Ruppin, E., *Neural Networks, 15*, 535-547, 2002). It is no coincidence that our assumptions concerning the outputs of the prefrontal network include both value estimates and actions. In this sense, we are conceptualizing the prefrontal network in terms of the actor-critic schema, and further elaborations of the paradigm might attain finer-grained architectural differentiation by importing existing ideas about the mapping between neuroanatomy and the actor-critic architecture (see, e.g., Song, H. F., Yang, G. R. & Wang, X.-J., *eLife, 6*, e21492, 2017). The implementation that we introduced in Simulation 6 takes a step in this direction, by dividing the prefrontal network into two halves, corresponding functionally to actor and critic.

The loss function employed in optimizing our networks includes terms for value regression and policy gradient. Actor-critic models suggest that something very much like these two losses is implemented in cortex and basal ganglia (e.g. Joel, Niv & Ruppin, *Neural Networks, 15*, 535-547, 2002). Value regression is believed to be implemented in the basal ganglia as dopamine prediction errors drive value estimates toward their targets (Montague, Dayan & Sejnowski, *Journal of Neuroscience, 16*, 1936-1947, 1996). Meanwhile, the policy gradient loss simply increases the strength of connections that participated in generating an action (when that action led to positive reward), which is believed to be the role of dopamine acting as modulator of plasticity at corticostriatal synapses.

Under the meta-RL theory, the role of DA in modulating synaptic function should play out only over a relatively long time-scale, serving to sculpt prefrontal dynamics over the course of multiple tasks. This is in contrast to the standard model, which assumes that DA can drive synaptic change sufficiently quickly to affect behavior on the scale of a few seconds. Interestingly, despite decades of intensive research, no direct evidence has yet been reported to indicate that phasic DA signals can drive synaptic change this rapidly. Indeed, we are aware of no experimental evidence that phasic elevations in DA concentrations spanning less than a full minute can significantly impact synaptic efficacy, and in many studies the impact of phasic fluctuations in DA can take minutes (or even tens of minutes) to ramp up (Brzosko, Z., Schultz, W. & Paulsen, O., *Elife, 4*, e09685, 2015; Otmakhova, N. A. & Lisman, J. E., *Journal of Neuroscience, 16*, 7478-7486, 1996; Yagishita, S. et al., *Science, 345*, 1616-1620, 2014). The lack of evidence for faster effects of phasic DA on synaptic plasticity presents another difficulty for the standard model. In contrast, the meta-RL framework directly predicts that such short-term effects of DA should in fact be absent in the prefrontal network, and that the effect of phasic DA signaling on synaptic efficacy in this network should instead operate on a significantly longer time-scale.

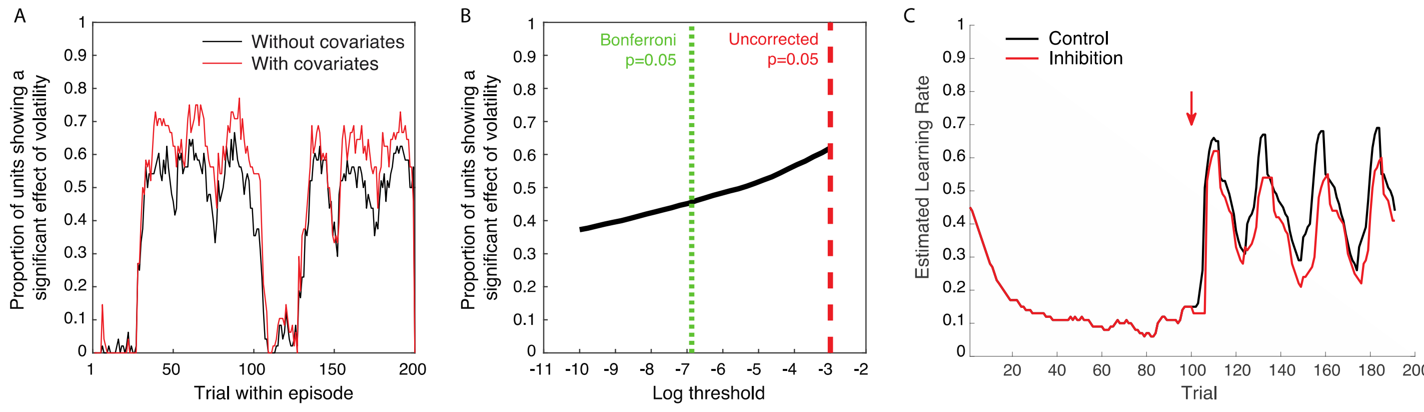




Supplementary Figure 4

Detailed behavioral analysis of simulation 1: reinforcement learning in the prefrontal network.

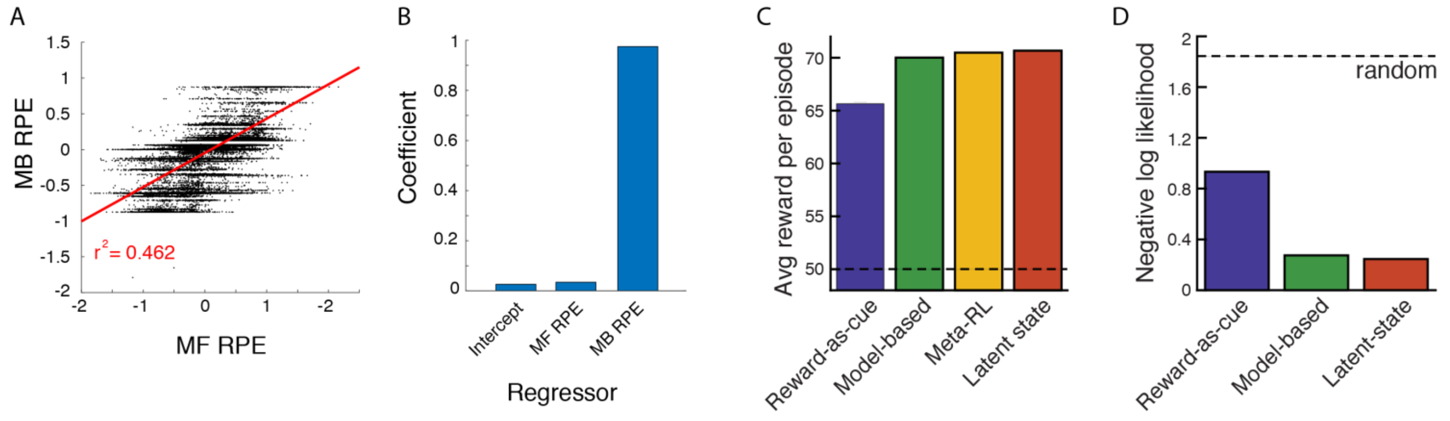
Left: Run lengths from the model. Bars indicate log trial counts for each run length, based on data from the last two-thirds of trials in each episode, and pooling across all reward-probability assignments. The pattern fits with the exponential decay typically reported in matching experiments (Corrado, Sugrue, Seung and Newsome, *Journal of the experimental analysis of behavior*, *84*, 581-617, 2005), paired with a tendency to alternate responses. The latter is typical in tasks not involving a changeover penalty, and was observed in the study by Tsutsui et al. (Fabian Grabenhorst, personal communication). **Right:** Histograms showing log counts across run lengths individually for a range of task parameterizations. $p_0(a_i)$ denotes the ground-truth reward probability for action *i.* The pattern appears bimodal toward the right of the figure, suggesting that the model may approximate a fixed alternation between poorer and richer arms, remaining at the poorer arm for only one step and at the richer arm for a fixed number of steps. Houston and McNamara (Houston and McNamara, *Journal of the experimental analysis of behavior*, *35*, 367-396, 1981) have shown that the optimal policy for discrete-trial variable-interval tasks, like the one studied here, takes this form.



Supplementary Figure 5

Additional simulations and analyses for simulation 2: adaptation of prefrontal-based learning to the task environment.

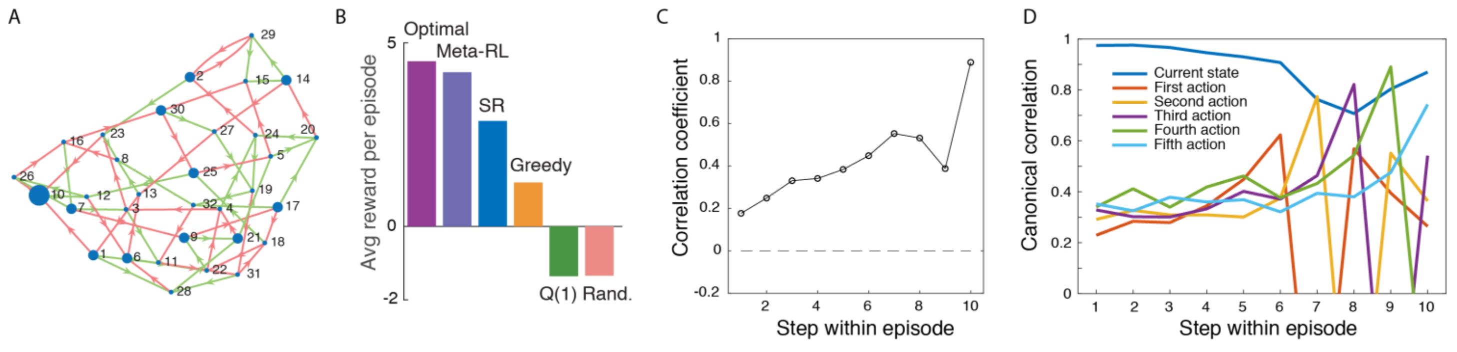
A) Proportion of LSTM units coding for volatility changes over the course of an episode. Y-axis shows the fraction of units whose activity significantly correlated (across episodes) with the true volatility. Volatility coding emerged shortly after the 25th trial, which is when the first reversal occurred if the episode began with high volatility. Volatility coding was assessed by regressing hidden activations against the true volatility in a single regression (black line), or by multiple regression of hidden activations against true volatility, previous reward, next action, and true reward probability (red line). A unit was labelled as coding volatility if the slope of the regression was significantly different from zero (2-tailed t-test) at a threshold of 0.05/n, where n=48 is the number of hidden units. **B**) Because units were not independent, the true correction for multiple comparisons lies somewhere between 1 and 1/n. The significance threshold, for declaring a unit as coding volatility, varies on the x-axis in this plot, with uncorrected and Bonferroni corrected levels indicated by dashed lines. The y-axis shows the mean over trials of number of units that exceeded this threshold. Although changing the threshold had some effect on the number of units significantly coding volatility, there was a population of units that appeared to not code volatility at all, and another population that coded it very strongly. **C**) Causal role of volatility-coding units in controlling learning rate. After identifying the two units whose activity correlated most strongly (positively) with volatility, we inhibited these units by deducting a fixed quantity (0.3) from their activity level (LSTM cell output), analogous to an optogenetic inhibition. The black curve shows the learning rate (estimated as described in Methods) without this "optogenetic" manipulation. The red curve shows the learning rate when "optogenetic" inhibition was applied starting at the onset of the high volatility period (arrow). Inhibition of these two units resulted in a slower measured learning rate, establishing a causal connection between the volatility-coding units and learning dynamics. Note that we treated the relevant ‘pretraining’ to have occurred outside the lab, and to reflect statistics of naturalistic task situations. For additional related work on the neural basis of learning rate adjustment, see Soltani et al., *Neural Networks, 19*, 1075-1090, 2006; and Farashahi et al., *Neuron, 94*, 401-414, 2017. Like the comparable work cited in the main text, these studies posit special-purpose mechanisms. In contrast, our theory posits the relevant mechanisms as one manifestation of a more general meta-learning phenomenon.



Supplementary Figure 6

Additional analyses for simulation 4: model-based behavior: the two-step task.

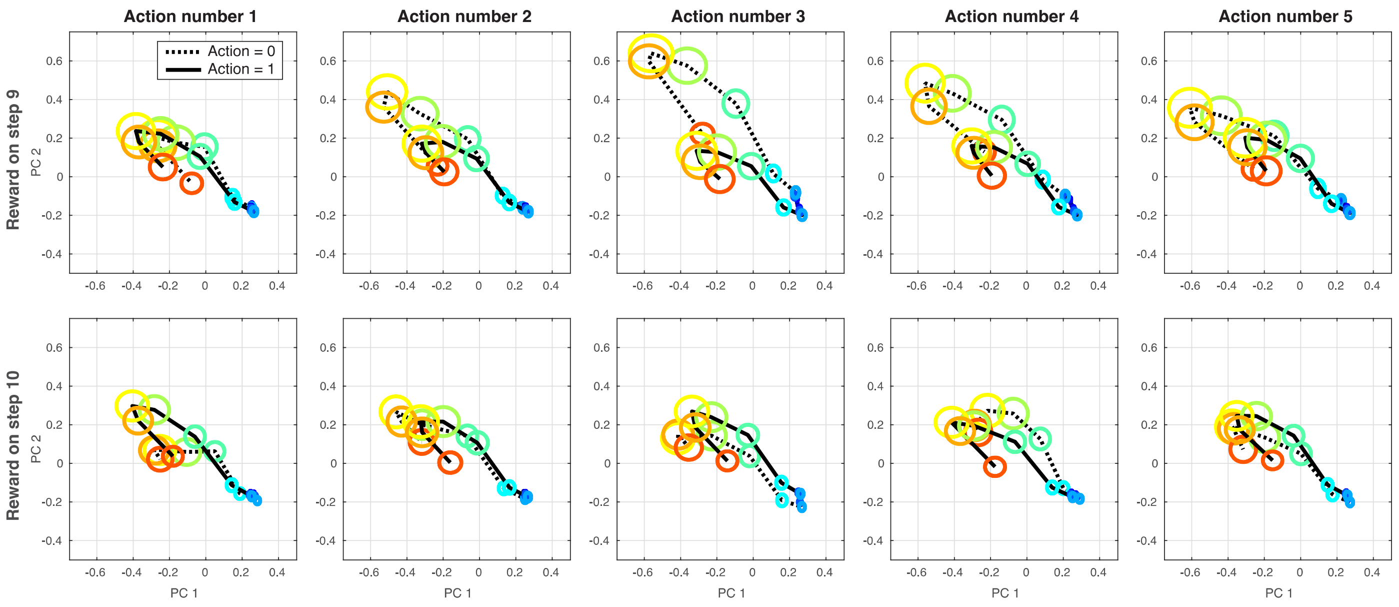
A) Model-based and model-free RPEs, derived from prospective planning and SARSA, respectively, were partially correlated with one another. (This motivated the hierarchical regression in our primary analyses; see Figure 5F,G in main text.) **B**) Beta coefficients from multiple regression using both MF and MB RPEs to explain meta-RL's RPEs. Only MB RPEs significantly explained meta-RL's RPEs. **C**) Two-step task performance for meta-RL, model-based RL, and two alternative models considered by Akam et al. (*PLoS Comput Biol 11,* e1004648, 2015). Akam and colleagues (see also Miller et al., *bioRxiv 096339*, 2016) have shown that a reward-by-transition interaction is not uniquely diagnostic for model-basedness. It is possible to construct agents using model-free learning, which display a reward-by-transition interaction effect on the probability of repeating an action. Here we simulated and fit two models from Akam et al. (2015): (1) the "latent-state" model, which does inference on the binary latent state of which stimulus currently has high reward probability and learns separate Q-values for two the different states, and (2) the "reward-as-cue" model, which uses for its state representation the cross-product of which second-stage state occurred on the previous trial and whether or not reward was received. The model-based and latent-state strategies yielded comparable levels of reward, while the reward-as-cue strategy paid less well. Although the model-based agent earned slightly less than the latent-state agent, this difference was eliminated if the former was modified to take account of the anti-correlation between the second-step state payoff probabilities, as in the agent used to predict model-based RPE signals in Simulation 4 (see Methods). The performance of meta-RL was comparable to that of both the model-based and latent-state agents. We note that meta-RL is therefore unlikely to implement the reward-as-cue strategy; this is further ruled out by regression results presented in Figure 5E, which are inconsistent with the pattern produced by reward-as-cue (see Akam et al., 2015, and Miller et al., 2016). The dashed line indicates performance for random action selection. Model free Q-learning performed significantly worse than reward-as-cue (data not shown). **D**) Fitting three models to meta-RL's behavior. Model-based and latent-state strategies yielded comparable fits, and again the small difference between them was eliminated if the model-based strategy was given access to a full model of the task. The reward-as-cue strategy fit significantly worse. All analyses here are done over 299 evaluation episodes, using 1 fully trained network out of 8 replicas. Results are highly similar for other replicas.



Supplementary Figure 7

New simulation to directly test model-based reasoning in meta-RL.

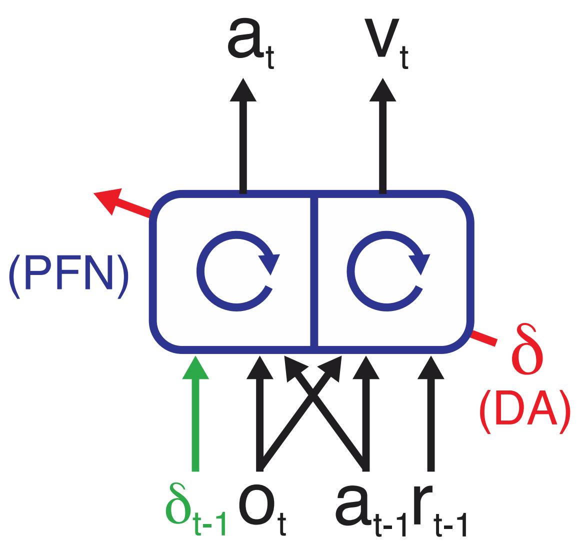
The two-step task alone is not sufficient to determine whether meta-RL can learn to implement truly prospective planning. Akam et al. (2015) have identified at least two non-prospective strategies ("reward-as-cue" and "latent-state") that can masquerade as prospective in the two-step task. Although we can rule out meta-RL using the reward-as-cue strategy (see Supplemental Figure 4), we cannot rule out the latent-state strategy. (Akam and colleagues, 2015, acknowledge that the latent-state strategy requires a form of model to infer the current latent state from observed reward outcomes (Kolling et al., *Nature Neuroscience, 19*, 1280, 2016) -- however, this is a relatively impoverished definition of model-based compared to fully prospective reasoning using a transition model.) However, other studies hint that true prospective planning may arise when recurrent neural networks are trained by RL in a multi-task environment. Recurrent neural networks can be trained to select reward-maximizing actions when given inputs representing Markov decision problem (MDP) parameters (e.g., images of random mazes) (Tamar et al., *Neural Information Processing Systems*, 2146, 2016; Werbos et al., *IEEE International Conference on Systems, Man and Cybernetics 1764,* 1996). A recent study by Duan and colleagues (*arXiv, 1611.02779*, 2016) also showed that recurrent networks trained on a series of random MDPs quickly adapt to unseen MDPs, outperforming standard model-free RL algorithms. And finally, recurrent networks can learn through RL to perform tree-search-like operations supporting action selection (Silver et al., *arXiv 1612.08810,* 2016; Hamrick et al., *Neural Information Processing Systems*, 2016; Graves et al., *Nature 538*, 471, 2016). We therefore sought to directly test prospective planning in meta-RL. To this end, we designed a novel revaluation task in which the agent was given five steps to act in a 32-state MDP whose transition structure was fixed across training and testing (for experiments using related tasks, see Kurth-Nelson et al., *Neuron, 91,* 194, 2016; Huys et al., *PNAS, 112*, 3098, 2015; Keramati et al., *PNAS*, *113*, 12868, 2016; Lee et al., *Neuron, 81*, 687, 2014). The reward function was randomly permuted from episode to episode, and given as part of the input to the agent. Because planning requires iterative computations, the agent was given a ‘pondering' (Graves, *arXiv:1603.08983*, 2016) period of five steps at the start of each episode. **A**) Transition structure of revaluation task. From each state, two actions were available (red and green arrows). For visualization, nodes are placed to minimize distances between connected notes. Rewards were sampled on each episode by randomly permuting a 32-element vector containing ten entries of +1, 21 entries of -1, and one entry of +5 -- node size in diagram shows a single sampled reward function. There were over a billion possible distinct reward functions: orders of magnitude more than the number of training episodes. Episodes began in a random state. Meta-RL's network architecture was identical to our other simulations, except the LSTM had 128 units. **B**) Meta-RL reached near optimal performance, when tested on reward functions not included in training. Significantly, meta-RL outperformed a ‘greedy' agent, which followed the shortest path toward the largest reward. Meta-RL also outperformed an agent using the successor representation (SR). Q(1): Q-learning with eligibility trace; Rand.: random action selection. Bars indicate standard error. **C**) Correlation between network value output (‘baseline') and ground-truth future reward grows during pondering period (steps 1 through 5). This indicates that the network used the pondering period to perform calculations that steadily improved its accuracy in predicting future reward. **D**) Canonical correlation between LSTM hidden state and several task variables, performed independently at each time step. Because “last action” was given as part of the input to the network, we orthogonalized the hidden state against this variable before calculating the canonical correlation. Unsurprisingly, we found that a linear code for action appeared most robustly at the step when the action was taken (the first action occurred on step 6, after five steps of ‘pondering'). However, this signal also ramped up prior to the onset of action. (The network cannot have a perfect representation of which action it will take, because the network's output is passed through a noisy softmax function to determine the actual action given to the environment.) The network also maintained a strong representation of which state it currently occupied. We note that the network's knowledge of the transition probabilities was acquired through RL. However, our theory does not exclude the existence of other neural learning mechanisms capable of identifying transition probabilities and other aspects of task structure. Indeed, there is overwhelming evidence that the brain identifies sequential and causal structure independent of reward (Glascher et al. *Neuron, 66*, 585, 2010; Tolman et al., *Psychological Review, 55*, 189, 1948). In subsequent work, it will be interesting to consider how such learning mechanisms might interact with and synergize with meta-RL (e.g., Hamrick et al., *Neural Information Processing Systems*, 2016). All analyses here are done over 1000 evaluation episodes, using 1 fully trained network. Other simulations using slightly different hyperparameters were conducted but not reported, and yielded very similar results.



Supplementary Figure 8

Trajectory of hidden activations during revaluation task (see Supplementary Figure 7).

Each panel shows the evolution, over steps within an episode, of the first two principal components of the hidden activations of the LSTM network. Each oval is one time step. The width and height of each oval illustrates the standard error (across episodes), and the color corresponds to the index of the time step within the episode (blue at the beginning of the episode and red at the end). Each panel summarizes a subset of episodes. Episodes were randomly generated under the constraint that the large (+5) reward could be reached within exactly four or five steps from the initial state. The top row of panels shows all the episodes where the large reward was reached on the ninth time step (five 'pondering' steps followed by four overt actions). The bottom row shows the episodes where the large reward was reached on the tenth time step (pondering plus five actions). Together these comprise 97% of all episodes. The remaining 3% of trials, where the large reward was not reached, are not depicted. Within each row, the *i*-th panel (from left to right) divides the depicted episodes into two groups: those in which the *i*-th action ultimately executed was 0 (dotted line), and those in which it was 1 (solid line). We found strong interactions between the future reward and the current and future actions. For example, the network's trajectory along the 2nd principal component began to diverge during the pondering period depending on whether it would take action 0 or 1 on the 3rd action. By the final step of pondering, this effect itself was strongly modulated according to whether reward would be received on step nine or ten of the episode. This appears to reflect a sophisticated but idiosyncratic planning algorithm. As in Supplementary Figure 7, PCA analysis was conducted over 1000 evaluation episodes using 1 fully trained network.



Supplementary Figure 9

Agent architecture for simulation 6.

Agent architecture employed in Simulation 6. Instead of a single LSTM, we now employ 2 LSTMs to model an actor (outputting the policy) and a critic (outputting the value estimate). The critic receives the same input as in our original model, while the actor receives the state observation, last action taken, and the reward prediction error that is computed based on the value estimate output from the critic.