A Appendix

A.1 Reinforcement learning notation

We adopt standard reinforcement learning notation [58]. An agent progresses through a sequence of states $\mathbf{s}_t$ by executing actions $\mathbf{a}_t$ which influence transition probabilities between states. We focus on the case where actions are continuous, vector valued quantities $\mathbf{a}_t \in \mathbb{R}^D$, where D is the dimension of the action space. The agent's policy—the probability distribution from which it samples actions given the state $\mathbf{s}$—is written as $\pi(\mathbf{a}|\mathbf{s})$. States are associated with scalar rewards $r_t = r(\mathbf{s}_t)$. The *value function* associated with a policy π is defined as follows

$$V_\pi(\mathbf{s}) = E_\pi\left[\sum_{t=0}^{T} \gamma^t r_t\right]. \tag{16}$$

Where the expectation is taken over the policy π and transitions of the environment, and conditioned on $\mathbf{s}_{t=0} = \mathbf{s}$, $\gamma \in [0, 1]$ is a discount factor, and T indicates the potentially finite horizon of an episode. An agent's goal is to learn a policy π that maximizes $V_\pi(\mathbf{s})$. We are often also interested in the *action-value* function $Q_\pi(\mathbf{s}, \mathbf{a})$, defined exactly the same as $V_\pi(\mathbf{s})$ except the sum is also conditioned on the initial action $\mathbf{a}_{t=0} = \mathbf{a}$. The Q-learning algorithm in Eq. 6 learns an estimate $\hat{Q}(\mathbf{s}, \mathbf{a})$ of the Q-function under the assumption that the current policy π to be deterministic such that the action selected in state $\mathbf{a}$ is always $\text{argmax}_{\mathbf{a}'}\hat{Q}(\mathbf{s}, \mathbf{a}')$.

A.2 Biological implementations of three-factor learning rules for continuous action spaces

The three-factor learning rule in Eq. 3 and the analogous rule for the action surprise model involve a product of a presynaptic term $\phi_j(\mathbf{s}_t)$, a dopaminergic term δ_t (or δ_t^+ in the action surprise model model), and a postsynaptic factor $(\mathbf{a}_t)_i - \mu_i(\mathbf{s}_t)$. Three-factor plasticity rules involving pre-synaptic, post-synaptic, and neuromodulatory factors have been observed experimentally [63, 9, 11] and are commonly used in RL modeling [58, 44]. However, the difference $(\mathbf{a}_t)_i - \mu_i(\mathbf{s}_t)$ involved in the postsynaptic term, which arises in our framework from the use of a continuous action space, requires more biological justification. Note that $\mu_i(\mathbf{s}_t)$ is the contribution to to the postsynaptic neuron's activity driven by the cortico-striatal synapses subject to the RL algorithm, while $(\mathbf{a}_t)_i$ takes into account exploration noise and influence from external controllers. Biologically, the difference between these terms may contribute to the plasticity in a number of ways, which we summarize here.

Option 1: Multi-compartment neurons. Potentially, the original basal ganglia-driven action signal $\mu_i(\mathbf{s}_t)$ and the difference signal $(\mathbf{a}_t)_i - \mu_i(\mathbf{s}_t)$ could arrive at different dendritic compartments. In this case, the total activity of the neuron would reflect the efferent copy $(\mathbf{a_i})_t$, but a compartment-specific plasticity rule would enable the synaptic weight update to depend only on the appropriate term.

Option 2: Time-varying striatal activity and a temporal plasticity kernel. If striatal projection neuron activity initially represents $\mu_i(\mathbf{s}_t)$ before receiving additional inputs which cause it to represent $(\mathbf{a}_t)_i$, then a spike-timing dependent three-factor learning rule with a suitable temporal kernel can result in an update that makes use of the difference in activity between the two phases. Indeed, such temporal kernels have been observed at cortico-striatal synapses [51].

Option 3: Normalization. Alternatively, the plasticity rule may only explicitly depend on projection neuron activity representing $(\mathbf{a_t})_i$, with normalization mechanisms across the striatal population implicitly contributing the $-\mu_i(\mathbf{s}_t)$ term. For concreteness, suppose the ith action dimension of the basal ganglia network policy $\mu_i(\mathbf{s})$ is a linear function of the *normalized, nonnegative* firing rates $\mathbf{x}$ of the striatal projection neuron population in response to $\mathbf{s}$:

$$\mu_i(\mathbf{s}) = \frac{\mathbf{w} \cdot \mathbf{x}(\mathbf{s})}{\sum_i x_i(\mathbf{s})}. \tag{17}$$

Further suppose that as the agent takes an action, the population activity is updated to $\mathbf{y}$ to reflect an efferent copy of the action $\mathbf{a}$:

$$a_i = \frac{\mathbf{w} \cdot \mathbf{y}(\mathbf{s})}{\sum_i y_i(\mathbf{s})}. \tag{18}$$

If cortico-striatal synapses are updated according to a three factor learning rule $\Delta \mathbf{w} \propto \delta \boldsymbol{\phi}(\mathbf{s}) \mathbf{y}(\mathbf{s})$, then at subsequent occurrences of state $\mathbf{s}$, the basal ganglia network will drive will evoke population activity:

$$\mathbf{x}^{\text{new}} = \mathbf{x} + \epsilon \delta \mathbf{y}, \tag{19}$$

for some learning rate ϵ. Note that, taking into account the normalization mechanism, the directional derivative of μ_i with respect to $\mathbf{x}$ along $\mathbf{y}$ is

$$D_{\mathbf{y}} \mu_i(\mathbf{x}) = \frac{\left(\sum_j x_j(\mathbf{s})\right) \mathbf{w} - (\mathbf{w} \cdot \mathbf{x}) \mathbf{1}}{\left(\sum_j x_j(\mathbf{s})\right)^2}. \tag{20}$$

And so following the weight update the action μ_i^{new} driven by the basal ganglia network in state $\mathbf{s}$ will be

$$\begin{aligned}
&(\mu_i)^{\text{new}}(\mathbf{s}) && (21)\\
&= (\mu_i)^{\text{old}}(\mathbf{s}) + \epsilon \frac{(\sum_i x_i(\mathbf{s})) \mathbf{w} \cdot \mathbf{y} - (\mathbf{w} \cdot \mathbf{x}) \mathbf{1} \cdot \mathbf{y}}{(\sum_i x_i(\mathbf{s}))^2} && (22)\\
&= (\mu_i)^{\text{old}}(\mathbf{s}) + \epsilon \delta \frac{\mathbf{w} \cdot \mathbf{y} - (\mathbf{w} \cdot \mathbf{x}) \mathbf{1} \cdot \mathbf{y}}{(\sum_i x_i(\mathbf{s}))} && (23)\\
&= (\mu_i)^{\text{old}}(\mathbf{s}) + \epsilon \delta \frac{(\sum_i y_i(\mathbf{s}))}{(\sum_i x_i(\mathbf{s}))} \left(\frac{\mathbf{w} \cdot \mathbf{y}}{\sum_i y_i(\mathbf{s})} - \frac{\mathbf{w} \cdot \mathbf{x}}{\sum_i x_i(\mathbf{s})} \right) && (24)\\
&= (\mu_i)^{\text{old}}(\mathbf{s}) + \epsilon \delta \frac{(\sum_i y_i(\mathbf{s}))}{(\sum_i x_i(\mathbf{s}))} (a - \mu(\mathbf{s})) && (25)\\
&= (\mu_i)^{\text{old}}(\mathbf{s}) + c \delta (a - \mu(\mathbf{s})). && (26)
\end{aligned}$$

for some constant c, which up to a scalar factor is exactly the update that would be induced by Eq. 3 in the absence of a normalization mechanism. Note that the update to the cortico-striatal weights does not necessarily follow the policy gradient in weight space, but we have shown that it induces the appropriate update to the basal ganglia network policy.

A.3 Alternative approaches to off-policy learning in continuous action spaces

A popular approach to continuous off-policy deep RL, adopted in the DDPG [35] and SAC [24] algorithms, is to parameterize $\hat{Q}(\mathbf{s}, \mathbf{a})$ with a neural network taking $\mathbf{s}$ and $\mathbf{a}$ as inputs, and to use a second neural network to learn $\text{argmax}_{\mathbf{a}'} \hat{Q}(\mathbf{s}, \mathbf{a}')$ (or, in the case of SAC, a probability distribution peaked at this value). This second network can be used both as the policy network and for the purpose of computing the factor $\max_{\mathbf{a}'} \hat{Q}(\mathbf{s}, \mathbf{a}')$ used in updates to $\hat{Q}$. While this strategy has proven to be effective in deep RL, a biological implementation is not readily apparent. In particular, in this approach, the parameter updates for the actor require computing $\nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})$, which depends a substantial amount of information that is not local to the actor. By contrast, in our approach, updates to the actor network depend on the critic network only through the scalar factor δ, which biologically can be signalled by dopamine release. Previous studies have suggested that a quadratic approximation to the Q-function, as we implement in our model, can approach or exceed the performance of DDPG [23].

Another class of approaches to off-policy learning, which does not involve any form of Q-learning, is based on importance sampling [48, 40, 59]. These methods modulate the size of learning updates by

the likelihood of an action (or sequence of actions) given the current policy. Such an approach could be implemented by decreasing the gain of phasic dopamine fluctuations following surprising actions, quite unlike the action surprise model we propose. We regard this possibility as unlikely given the experimental evidence (discussed in the main text) for surprise and action initiation signals in dopamine activity. Moreover, importance sampling-based approaches have a disadvantage compared to Q-learning-based methods in that they learn slowly from data that is sufficiently far off-policy. However, it should be noted that our approach to circumventing this problem, while it is able to learn faster from off-policy data, requires assuming a parameterization of the Q-function which, if inaccurate, can ultimately harm performance.

A.4 Learning confidence parameters

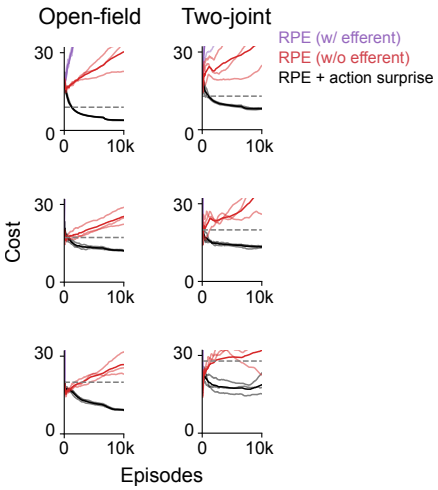

Figure 5: Same information as Fig. 3C, but in this case the black trace was learned with the action surprise coefficient $\frac{1}{\sigma^2}$ learned online, following the update rule Eq. 28.

In the on-policy case introduced in Section 2, following the policy gradient (Eq. 1) yields the following update for the variance parameter σ:

$$\Delta\sigma \propto -\frac{1}{\sigma^2} + \frac{2}{\sigma^3}\delta_t \left\|\mathbf{a}_t - \boldsymbol{\mu}(\mathbf{s}_t)\right\|^2. \tag{27}$$

In the action surprise model, under the parameterization of Eq. 8, following gradient updates of the loss function of Eq. 6 gives the following update for the action surprise coefficient σ:

$$\Delta\sigma = \frac{2}{\sigma^3}\delta_t^+ \left\|\mathbf{a}_t - \boldsymbol{\mu}(\mathbf{s}_t)\right\|^2. \tag{28}$$

Note that this update is exactly the same as in the off-policy case but without the decay term $-\frac{1}{\sigma^2}$ providing an impulse toward increasing confidence (decaying σ) with increasing training data. This difference reflects the fact that the role of σ in the Q-learning model algorithm is not to measure uncertainty as in the on-policy case, but rather to approximate the width of the Q-function, which will have some finite value independent of the amount of data observed.

In Fig A.4 we show the result of using this update rule to learn the coefficient σ online (after initializing $\frac{1}{\sigma^2}$ at 0, rather than optimizing it as a hyperparameter) for the full off-policy learning case. The model is capable of converging to an appropriate σ such that it learns successfully, while (as discussed in the main text) the RPE-only models are incapable of learning.

A.5 Further simulation details

For all experiments, we optimized hyperparameters over the following ranges: exploration noise variance $\in \{0.5, 1.0, 2.0, 4.0, 8.0\}$, learning rate $\in \{0.05, 0.1, 0.2\}$. The critic learning rate was

always set to 0.1. For the RPE-only algorithms, the learning rate of the actor was optimized over $\{0.0315, 0.0625, 0.125, 0.25, 0.5\}$, and for the action surprise model, the value of $\frac{1}{\sigma^2}$ was optimized over the same set. The velocity and acceleration penalties in the task cost functions, α_1 and α_2, were each always set to 0.1, and the discount factor γ always set to 0.99.

For all simulations, training was conducted for 100000 episodes, each with a length of 10 timesteps. Each training run was performed on a single NVIDIA GeForce RTX 2080 Ti GPU on an internal cluster.

A.6 Results for averaging-based combination of basal ganglia network and external controller policies

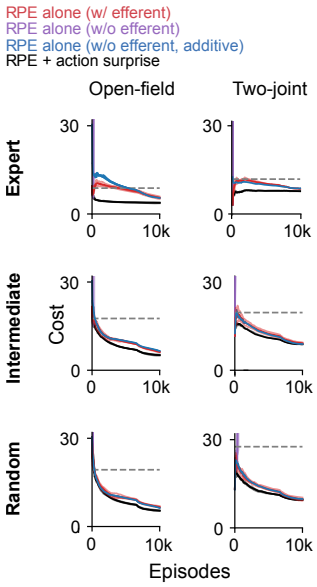

Figure 6: Same information as Fig. 3B, for the case where the basal ganglia policy and external controller policy are combined by taking their mean, rather than sampling. Additionally blue trace shows an alternative in which output of the basal ganglia network is added to (rather than averaged with) the external controller policy.

A.7 Results for task variant with sparse rewards for reaching the goal location

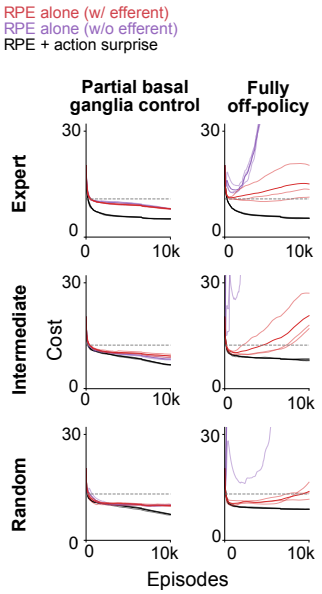

Figure 7: For the two-joint arm task, same information as Fig. 3B and Fig. 3C, for the case where the cost associated with distance from the goal location is sparse (equal to 1 when squared distance exceeds 0.25, and 0 otherwise).