PNAS
www.pnas.org

Supplementary Information for

Rational Thoughts in Neural Codes

Zhengwei Wu, Minhae Kwon, Saurabh Daptardar, Paul Schrater, and Xaq Pitkow

Xaq Pitkow.
E-mail: xaq@rice.edu

This PDF file includes:

Figs. S1 to S3
Table S1
SI References

Belief MDP. In a belief MDP, an agent chooses actions based on the belief state b_t, so the agent must compute the belief state at each time given its observations and actions up to that time. This can be computed online using the Markov property, according to

$$B(s_t|b_t) = B(s_t|o_{1:t}, a_{1:t-1}) \quad [1]$$
$$= B(s_t|o_t, a_{t-1}, b_{t-1}) \quad [2]$$
$$= \frac{1}{Z} O(o_t|s_t) \int ds_{t-1}\, T(s_t|s_{t-1}, a_{t-1}) B(s_{t-1}|o_{t-1}, a_{t-2}, b_{t-2}) \quad [3]$$
$$= \frac{1}{Z} O(o_t|s_t) \int ds_{t-1}\, T(s_t|s_{t-1}, a_{t-1}) B(s_{t-1}|b_{t-1}) \quad [4]$$

To find the optimal policy, an agent evaluates the value of each action and state. If the agent were given future observations and actions, then its future beliefs would be known. But when observations are unknown, the agent has only a distribution over beliefs, arising from the distribution of future observations it may encounter from the distribution of future world states. The transition probability between belief states is then

$$\overline{T}(b_{t+1}|b_t, a_t) = \int do_{t+1}\, P(b_{t+1}|b_t, a_t, o_{t+1}) \overline{O}(o_{t+1}|b_t, a_t) \quad [5]$$

where

$$\overline{O}(o_{t+1}|b_t, a_t) = \int ds_{t+1}\, ds_t\, O(o_{t+1}|s_{t+1}) T(s_{t+1}|s_t, a_t) B(s_t|b_t) \quad [6]$$

is the distribution of future observations given the present belief and action. The parameters of this belief transition probability $\overline{T}(b_{t+1}|b_t, a_t)$ therefore include parameters from both the world state transitions $T(s_{t+1}|s_t, a_t)$ and observation functions $O(o_t|s_t)$.

The true instantaneous reward function $R(s, a)$ depends on the actual state and action. But for planning into the future, the agent must consider the reward as a function of its *beliefs*, which it expects to be

$$\overline{R}(b_t, a_t) = \int ds_t\, R(s_t, a_t) B(s_t|b_t) \quad [7]$$

These beliefs, belief transitions $\overline{T}$, and rewards $\overline{R}$ then determine the value of any policy through the recursive Bellman equation (1),

$$Q(b_t, a_t) = \overline{R}(b_t, a_t) + \gamma \iint da_{t+1} db_{t+1}\, \overline{T}(b_{t+1}|b_t, a_t) \pi(a_{t+1}|b_{t+1}) Q(b_{t+1}, a_{t+1}) \quad [8]$$

The optimal policy deterministically selects whichever action maximizes that value function given the current belief, $a_t = \text{argmax}_a Q(b_t, a)$. As a generalization, here we allow actions to be sampled randomly from the softmax policy

$$a_t \sim \pi(a|b_t) = \frac{1}{Z} e^{\frac{Q(b_t, a)}{\tau}} \quad [9]$$

with temperature τ and normalization Z.

One subtle point to note is that for a deterministic policy, for which $\pi(a|b) = \delta(a - a^*(b))$ with optimal action $a^*(b)$, the next belief state is only a function of the last belief state and the current observation. Although the next world state does of course depend on the selected action, this action is perfectly predictable from the last belief state and so $P(b_{t+1}|b_t, a_t, o_{t+1}) = P(b_{t+1}|b_t, o_{t+1})$. In contrast, for a stochastic policy like the one we allow in our simulations, the belief state does not fully determine the action, and thus to know future belief states the agent would have to also observe the action that was actually sampled from the policy. This could be notated in multiple equivalent ways. For example, the graphical models in Figure 1 could include arrows from each action to the next belief state. Alternatively, the selected action could be considered part of the world state that is perfectly observed, and so it would be transmitted automatically to the next belief state without the addition of any explicit arrows. For ease of exposition and clarity of the trellis diagram we have chosen this latter approach. The mathematics, however, make these particular dependencies explicit.

Markov structure in Inverse Rational Control. The log-likelihood of the observed data $\mathcal{L}(\theta)$ can be decomposed as the product of terms at each time point.

$$\mathcal{L}(\theta) = \log p(o_{1:T}, a_{1:T}, s_{1:T}|\theta, \phi) \quad [10]$$
$$= \log \int db_{1:T}\, p(b_{1:T}, o_{1:T}, a_{1:T}, s_{1:T}|\theta, \phi) \quad [11]$$
$$= \log \int db_{1:T}\, p(s_1|\phi) p(b_1, o_1|\theta) \prod_t \pi(a_t|b_t, \theta) p(b_{t+1}|b_t, a_t, o_{t+1}, \theta) O(o_{t+1}|s_{t+1}, \phi) T(s_{t+1}|s_t, a_t, \phi) \quad [12]$$

According to the graphical model in Figure 1, the true world states s are outside the Markov blanket of the belief states b, and only appear in terms with the experimental parameters ϕ. They do not appear with the agent's parameters θ in this likelihood, because what matters to our model is not what actually happens in the world but rather what the agent *thinks* happens. As a result, the first term and the last two terms in the integral [12], which depend on experimental parameters ϕ, can be factored out, simplifying the calculation of the log-likelihood of the observed data $\mathcal{L}(\theta)$.

Another factor we did not consider here is that the agent's observations $o_{1:T}$ could have appreciable sensory noise that is private to the agent and thus latent for IRC. The model likelihood [11] would then have to integrate over them as well (2, 3). In this paper we assumed tasks are structured so that uncertainty about the world state s_t is dominated by external noise, not internal noise, so the observations $o_{1:T}$ are known to the experimenter.

As in the Expectation-Maximization (EM) algorithm (4)*, the log-likelihood of the observed data $\mathcal{L}(\theta)$ [10] can be written as the sum of the expected complete data log-likelihood $\mathcal{Q}(\theta)$ and the entropy H of the posterior over beliefs, $\mathcal{L}(\theta) = \mathcal{Q}(\theta) + H$. Each of these terms can be decomposed into sums of transition probabilities and policies at each time, due to the Markov property. Using the graphical model structure shown in Figure 1B, we have

$$\mathcal{Q}(\theta) = \Big\langle \log p(b_1, o_1, s_1|\theta, \phi) \tag{13}$$

$$+ \sum\nolimits_t \log p(b_{t+1}|b_t, a_t, o_{t+1}, \theta) \tag{14}$$

$$+ \sum\nolimits_t \log O(o_{t+1}|s_{t+1}, \phi) \tag{15}$$

$$+ \sum\nolimits_t \log T(s_{t+1}|s_t, a_t, \phi) \tag{16}$$

$$+ \sum\nolimits_t \log \pi(a_t|b_t, \theta) \Big\rangle_{p(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta, \phi)} \tag{17}$$

The term in [14] depends only on the parameters for the state dynamics and observations, while the policy term in [17] depends on the dynamics, observation parameters, and reward functions.

According to (5), the entropy H of the posterior over beliefs can be calculated recursively as

$$H(b_{1:t-1}|b_t, o_{1:t}, a_{1:t}, s_{1:t}, \theta, \phi) = \int db_t \, H(b_{1:t-2}|b_{t-1}, o_{1:t-1}, a_{1:t-1}, s_{1:t-1}, \theta, \phi) p(b_{t-1}|b_t, o_{1:t}, a_{1:t}, s_{1:t}, \theta, \phi) \tag{18}$$

$$+ H(b_{t-1}|b_t, o_{1:t}, a_{1:t}, s_{1:t}, \theta, \phi) \tag{19}$$

where $p(b_{t-1}|b_t, o_{1:t}, a_{1:t}, s_{1:t}, \theta, \phi)$ can be calculated with Bayes rule. For the last time point, $t = T$, the entropy of the entire belief sequence can be obtained similarly as

$$H(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta, \phi) = \int db_T \, H(b_{1:T-1}|b_T, o_{1:T}, a_{1:T}, s_{1:T}, \theta, \phi) p(b_T|a_{1:T}, o_{1:T}, s_{1:T}, \theta, \phi) \tag{20}$$

$$+ H(b_T|a_{1:T}, o_{1:T}, s_{1:T}, \theta, \phi) \tag{21}$$

Line search method. In small problems like the foraging task considered in the main text, we can sometimes optimize the log-likelihood function $\mathcal{L}(\theta)$ directly by a greedy line search method. Here we iteratively perform one-dimensional grid searches along random directions in parameter space. Once we find the optimal parameters on a line, we choose a new direction randomly from that starting point. We repeat this procedure until convergence.

EM algorithm. The EM algorithm (4) enables us to solve for the parameters that give best explanation of the observed data, while inferring unobserved states in the model. Recall that the log-likelihood of the observed data $\log \mathcal{L}(\theta)$ can be written as

$$\mathcal{L}(\theta) = \log \int db_{1:T} \, p(b_{1:T}, o_{1:T}, a_{1:T}, s_{1:T}|\theta, \phi) \tag{22}$$

Here θ is a parameter vector which includes both assumptions about the world dynamics and the parameters determining the subjective magnitudes of rewards and action costs. We alternately update the parameters θ to improve the expected complete-data log-likelihood, and calculate the posterior over latent states based on the estimated parameters from the most recent iteration.

In the E-step of the EM algoritm, the estimated parameters θ^{old} from the previous iteration determine the posterior distribution over the latent variables given the observed data, $p(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi)$. In the M-step, the observed data log-likelihood function to be maximized reduces to

$$\overline{\mathcal{L}}(\theta) = \mathcal{Q}(\theta, \theta^{\text{old}}) + H(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi) \tag{23}$$

*Unfortunately, the conventional notations in EM and reinforcement learning collide here, both using the same letter: this $\mathcal{Q}$ auxiliary function is denoted in the Calligraphic font to distinguish it from the state-action value function Q in the MDP model.

To be consistent with (6), we use $\mathcal{Q}(\theta, \theta^{\text{old}})$ as the auxiliary function that describes the expected complete data log likelihood; $H(\cdot)$ is the entropy of the posterior of the latent variable. Note that $H(\cdot)$ is not a function of θ, and thus has a fixed value if θ^{old} is fixed.

The $\mathcal{Q}$-auxiliary function can be expressed as:

$$\mathcal{Q}(\theta, \theta^{\text{old}}) = \langle \log p(b_{1:T}, a_{1:T}, o_{1:T}, s_{1:T}|\theta, \phi) \rangle_{p(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi)} \quad [24]$$

where ϕ are the parameters in the experimental setup that determine the world dynamics. Since ϕ are fixed in the experiment and known in the analysis, they do not affect the model likelihood.

The complete data likelihood $p(b_{1:T}, a_{1:T}, o_{1:T}, s_{1:T}|\theta, \phi)$ can be factorized into transition probabilities and policies at each time due to the Markov property. We can therefore decompose the expected complete data log likelihood $\mathcal{Q}(\theta, \theta^{\text{old}})$ using the graphical model structure, as described in [13–17], except now the posterior distribution over beliefs is based on the previous iteration's parameters:

$$\begin{aligned}
\mathcal{Q}(\theta, \theta^{\text{old}}) = \Big\langle & \quad \log P(b_1, o_1, s_1|\theta, \phi) && [25]\\
& + \sum\nolimits_t \log p(b_{t+1}|b_t, a_{t+1}, o_{t+1}, \theta) && [26]\\
& + \sum\nolimits_t \log O(o_{t+1}|s_{t+1}, \phi) && [27]\\
& + \sum\nolimits_t \log T(s_{t+1}|s_t, a_t, \phi) && [28]\\
& + \sum\nolimits_t \log \pi(a_t|b_t, \theta) \Big\rangle_{p(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi)} && [29]
\end{aligned}$$

Instead of solving for the optimal θ in a closed form, we use gradient descent to update the parameter θ in the M-step.

With fixed parameters θ^{old} from the previous iteration, the entropy of the latent state $H(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi)$ is fixed. As a result, we only need to update parameter θ to maximize function $\mathcal{Q}(\theta, \theta^{\text{old}})$ in the M-step. The first term in [25] reflects the initial belief distribution, and it has a negligible contribution to $\mathcal{Q}$ when there are many time points t. In [26], the transition probability $p(b_{t+1}|b_t, a_{t+1}, o_{t+1}, \theta)$ is a function of the dynamics parameters, while in [29], the policy term $\pi(a_t|b_t, \theta)$ is a function of both the dynamic parameters and the rewards. Since the transition probability is a matrix whose elements are functions of the dynamics parameters, the gradients can be taken element-wise. We will show how the gradient of the policy function can be derived based on the Q value function in the next part.

Over iterations of the EM algorithm, the value of the log-likelihood $\mathcal{L}(\theta)$ always increases toward a (possibly local) maximum.

Value gradient in IRC. To take gradient of the $\mathcal{Q}(\theta, \theta^{\text{old}})$ auxiliary function, it is critical to have the gradient of the policy function. For a softmax policy based on the value function, $\pi(a|b) \sim \frac{1}{Z} e^{Q(b,a)/\tau}$, if we have the gradient of the value function with respect to the parameters, we can then obtain the gradient of the policy function using the chain rule:

$$\frac{\partial \pi(a|b)}{\partial \theta_i} = \frac{\partial \pi(a|b)}{\partial Q(b,a)} \frac{\partial Q(b,a)}{\partial \theta_i} + \int_{a' \neq a} da' \frac{\partial \pi(a|b)}{\partial Q(b,a')} \frac{\partial Q(b,a')}{\partial \theta_i}. \quad [30]$$

Recall that the Q value function for belief state-action pairs can be written as

$$\begin{aligned}
Q(b_t,a_t) = & \ \overline{R}(b_t, a_t) + \\
& \gamma \iint da_{t+1} db_{t+1} \, \overline{T}(b_{t+1}|b_t, a_t) \pi(a_{t+1}|b_{t+1}) Q(b_{t+1}, a_{t+1})
\end{aligned}$$

Consider now a specific element θ_i of the parameter vector θ. For a particular (b_t, a_t) pair, taking the derivative of both sides with respect to θ_i, we have

$$\begin{aligned}
\frac{\partial Q(b_t, a_t)}{\partial \theta_i} = & \frac{\partial \overline{R}(b_t, a_t)}{\partial \theta_i} && [31]\\
& + \gamma \int db_{t+1} \frac{\partial \overline{T}(b_{t+1}|b_t, a_t)}{\partial \theta_i} \int da_{t+1} \pi(a_{t+1}|b_{t+1}) Q(b_{t+1}, a_{t+1}) && [32]\\
& + \gamma \int db_{t+1} \overline{T}(b_{t+1}|b_t, a_t) \int da_{t+1} \frac{\partial \pi(a_{t+1}|b_{t+1})}{\partial \theta_i} Q(b_{t+1}, a_{t+1}) && [33]\\
& + \gamma \int db_{t+1} \overline{T}(b_{t+1}|b_t, a_t) \int da_{t+1} \pi(a_{t+1}|b_{t+1}) \frac{\partial Q(b_{t+1}, a_{t+1})}{\partial \theta_i} && [34]
\end{aligned}$$

Note here $\frac{\partial Q(b_t, a_t)}{\partial \theta_i}$ is a scalar. We define $c_i(\cdot)$ as the sum of the first two lines [31–32]:

$$c_i(b_t, a_t) = \frac{\partial \overline{R}(b_t, a_t)}{\partial \theta_i} + \gamma \int db_{t+1} \frac{\partial \overline{T}(b_{t+1}|b_t, a_t)}{\partial \theta_i} \int da_{t+1} \pi(a_{t+1}|b_{t+1}) Q(b_{t+1}, a_{t+1}) \quad [35]$$

With this substitution we have

$$\frac{\partial Q(b_t,a_t)}{\partial\theta_i} = c_i(b_t,a_t)+\gamma\int db_{t+1}\overline{T}(b_{t+1}|b_t,a_t)\int da_{t+1}\Big[\frac{\partial\pi(a_{t+1}|b_{t+1})}{\partial\theta_i}Q(b_{t+1},a_{t+1})+\pi(a_{t+1}|b_{t+1})\frac{\partial Q(b_{t+1},a_{t+1})}{\partial\theta_i}\Big] \quad [36]$$

where $\frac{\partial\pi(a_{t+1}|b_{t+1})}{\partial\theta_i}$ can be written as a function of $\frac{\partial Q(b_{t+1},a_{t+1})}{\partial\theta_i}$ according to the chain rule [30].

Suppose there are $|\mathcal{B}|$ distinct belief states, and $|\mathcal{A}|$ actions. If we vectorize the matrices $Q(b_t,a_t)$, $\pi(a_t|b_t)$ and $c_i(b_t,a_t)$ over these discrete belief states and actions, denoting them as $\boldsymbol{Q}_t^V$, $\boldsymbol{\pi}_t^V$ and $\boldsymbol{c}_{i,t}^V$ respectively, then these are vectors with length $|\mathcal{B}||\mathcal{A}|$. Equation [36] can then be rewritten as a linear function

$$\begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i}\\ \vdots\end{bmatrix} = \begin{bmatrix}\vdots\\ \boldsymbol{c}_{i,t}^V\\ \vdots\end{bmatrix} + \gamma \underbrace{\begin{bmatrix}\vdots & \vdots & \vdots\\ \overline{T}(b_{t+1}|b_t,a_t) & \overline{T}(b_{t+1}|b_t,a_t) & \overline{T}(b_{t+1}|b_t,a_t)\\ \vdots & \vdots & \vdots\end{bmatrix}}_{\boldsymbol{\Gamma}(\overline{T}(b_{t+1}|b_t,a_t))} \Bigg(\begin{bmatrix}\ddots & & \\ & \boldsymbol{Q}_t^V & \\ & & \ddots\end{bmatrix} \begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{\pi}_t^V}{\partial\boldsymbol{Q}_t^V}\\ \vdots\end{bmatrix} + \begin{bmatrix}\ddots & & \\ & \boldsymbol{\pi}_t^V & \\ & & \ddots\end{bmatrix}\Bigg) \begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i}\\ \vdots\end{bmatrix},$$

where $\begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i}\\ \vdots\end{bmatrix}$ is a $|\mathcal{B}||\mathcal{A}|\times 1$ vector, $\begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{\pi}_t^V}{\partial\boldsymbol{Q}_t^V}\\ \vdots\end{bmatrix}$ is a $|\mathcal{B}||\mathcal{A}|\times|\mathcal{B}||\mathcal{A}|$ matrix, $\begin{bmatrix}\ddots & & \\ & \boldsymbol{Q}_t^V & \\ & & \ddots\end{bmatrix}$ and $\begin{bmatrix}\ddots & & \\ & \boldsymbol{\pi}_t^V & \\ & & \ddots\end{bmatrix}$ are diagonal matrices with vectors $\boldsymbol{Q}_t^V$ and $\boldsymbol{\pi}_t^V$ along the diagonal, and $\boldsymbol{\Gamma}(\overline{T}(b_{t+1}|b_t,a_t))$ is a function of the belief transition probability $\overline{T}(b_{t+1}|b_t,a_t)$. The derivative of $\boldsymbol{Q}_t^V$ with respect to the parameter θ_i can then be solved as

$$\begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i}\\ \vdots\end{bmatrix} = \Bigg(\boldsymbol{I}-\gamma\boldsymbol{\Gamma}(\overline{T}(b_{t+1}|b_t,a_t))\Bigg(\begin{bmatrix}\ddots & & \\ & \boldsymbol{Q}_t^V & \\ & & \ddots\end{bmatrix} \begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{\pi}_t^V}{\partial\boldsymbol{Q}_t^V}\\ \vdots\end{bmatrix} + \begin{bmatrix}\ddots & & \\ & \boldsymbol{\pi}_t^V & \\ & & \ddots\end{bmatrix}\Bigg)\Bigg)^{-1} \begin{bmatrix}\vdots\\ \boldsymbol{c}_{i,t}^V\\ \vdots\end{bmatrix} \quad [37]$$

Without the brackets indicating the matrix shapes, finally we obtain

$$\frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i} = \left(\boldsymbol{I}-\gamma\boldsymbol{\Gamma}\left(\mathrm{Diag}(\boldsymbol{Q}_t^V)\frac{\partial\boldsymbol{\pi}_t^V}{\partial\boldsymbol{Q}_t^V}+\mathrm{Diag}(\boldsymbol{\pi}_t^V)\right)\right)^{-1}\boldsymbol{c}_{i,t}^V. \quad [38]$$

With the chain rule [30], we can obtain the gradients of the policy with respect to the parameters θ, which lets us calculate the gradient of the $\mathcal{Q}(\theta,\theta^{\mathrm{old}})$ function in [25–29], and use them in the M-step of the EM algorithm applied to IRC. The result is an improved estimate of the agent's internal model based on its sensory observations and actions.

Foraging task and POMDP agent parameters. The foraging task described in the Results has two reward boxes for which the true reward availability followed a telegraph process, alternating between available and unavailable at uniform switching rates. For the two boxes, the true appearance and disappearance probabilities in one time step were $\gamma_1^*=0.2$, $\gamma_2^*=0.12$ and $\epsilon_1^*=0.05$, $\epsilon_2^*=0.07$.

Each box also displayed a sensory cue at each time conditioned on the reward availability, comprising five possible colors, with redder (bluer) colors indicating higher (lower) probability that food is currently available in the box. To be an interesting task, the distributions under the two states should overlap enough that the animal cannot depend primarily on the color cue to anticipate the food availability. Color values for both boxes are drawn independently at each time from a binomial distribution with five states, with mean $q_1^*=0.4$ when food is available in the box, and $q_2^*=0.6$ otherwise, and variance 0.96 for both of the two cases.

The target agent makes wrong assumptions about all of these parameters, acting rationally for a task where $\gamma_1=0.17$, $\gamma_2=0.1$, $\epsilon_1=0.1$, $\epsilon_2=0.03$, $q_1=0.45$, and $q_2=0.55$.

We measure gains and losses in currency of reward, $R\equiv 1$. In those units, our target agent incurs a subjective cost of 0.3 when pressing the button, and a cost of 0.15 when traveling. Switching between boxes requires two steps, for a total cost of 0.3. We also allow a 'grooming' reward $R=0.2$ for waiting at the center location. Our agent uses a softmax policy with temperature $\tau=0.1$.

Simulated brain. We trained a neural network to match the behavior of multiple rational agents. The target behaviors were implemented by agents that used optimal belief updates and a softmax policy with nonzero temperature. For simplicity, we discretized beliefs about reward availability for each box into $N = 10$ belief states. We defined the transition matrix in the discretized belief space by binning the continuous transition matrix $\overline{T}(b_{t+1}|b_t, a_t)$. We allowed a small diffusion between neighboring bins, which reflects dynamic belief stochasticity. With the defined transition matrices and reward functions for different actions for the internal model, we can solve for the optimal softmax policy by value iteration (1).

Our neural network used one recurrently connected layer of 100 tanh units that received external inputs from the world-generated observations, agent-generated actions, and the task parameters. The recurrently connected neurons provided input to a two-layer perceptron, with 50 ReLU neurons followed by 5 policy neurons (Figure S1).

The architecture was built in PyTorch and optimized by supervised learning using gradient descent on a loss function given by the average KL-divergence between the neural network's output policy and the teacher's POMDP policy:

$$L = \frac{1}{T} \sum_t D_{\mathrm{KL}}[\pi_{\mathrm{NN}}(a|\boldsymbol{r}_t)||\pi_{\mathrm{POMDP}}(a|b_t)] \quad [39]$$

The neural network policy π_{NN} samples actions according to a softmax over the five output neurons $\pi_{\mathrm{NN}}(a|\boldsymbol{r}_t) = \mathrm{softmax}(\boldsymbol{r}_t^{\mathrm{act}})$. We trained the neural network to match policies with 31 teacher agents following POMDPs with different parameters. After 50 iterations of 310 batches with 1000 time points per batch, the trained neural network successfully reproduced the target distribution of actions at each time, within an average KL divergence of 0.0047.

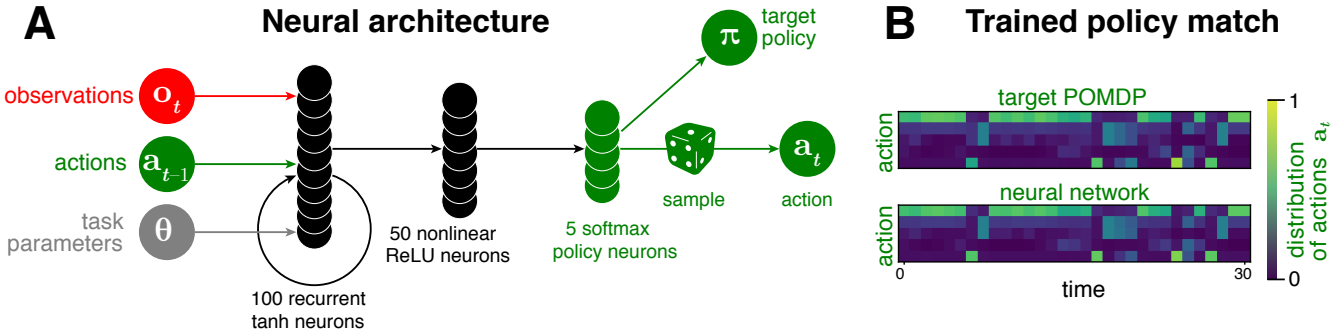

Fig. S1. $\mathbf{A}$: Architecture of a synthetic brain trained to behave rationally by matching the policy π of a POMDP agent. The recurrent network uses 100 fully-connected neurons with a tanh nonlinearity, and the feedforward layer uses 50 ReLU neurons. There are 5 policy neurons, one for each possible action, and at each time step the network samples an action from a softmax applied to these policy neurons' outputs. $\mathbf{B}$: Neural network reproduces the time-dependent action distribution of a rational agent when tested on a novel task.

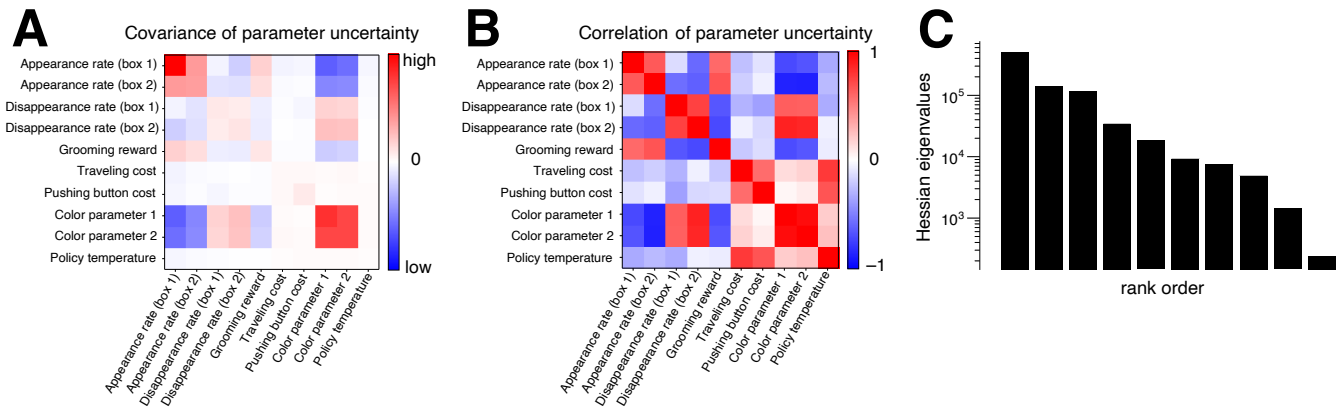

Fig. S2. Uncertainty for the parameters fit by IRC. Confidence intervals on parameters were estimated by first using finite differences to compute the curvatures (Hessian) of the observed data log-likelihood $H_{ij} = \partial_{\theta_i}\partial_{\theta_i} \log p(\theta|o_{1:\infty}, a_{1:\infty})$, and then calculating the negative inverse $\Sigma = -H^{-1}$ to give the local covariance of the equivalent gaussian at the most probable value of the parameters. ($\mathbf{A}$) The covariance matrix Σ of the local uncertainty of the model parameters. ($\mathbf{B}$) Corresponding matrix of Pearson correlation coefficients. ($\mathbf{C}$) Eigenvalues of the curvature matrix reveal a spectrum of uncertainties. The lowest curvature mode, $i.e.$ the sloppiest direction in parameter space, has an eigenvector especially concentrated on the disappearance rate on box 2. This rate was already so low that IRC cannot tell precisely how low.

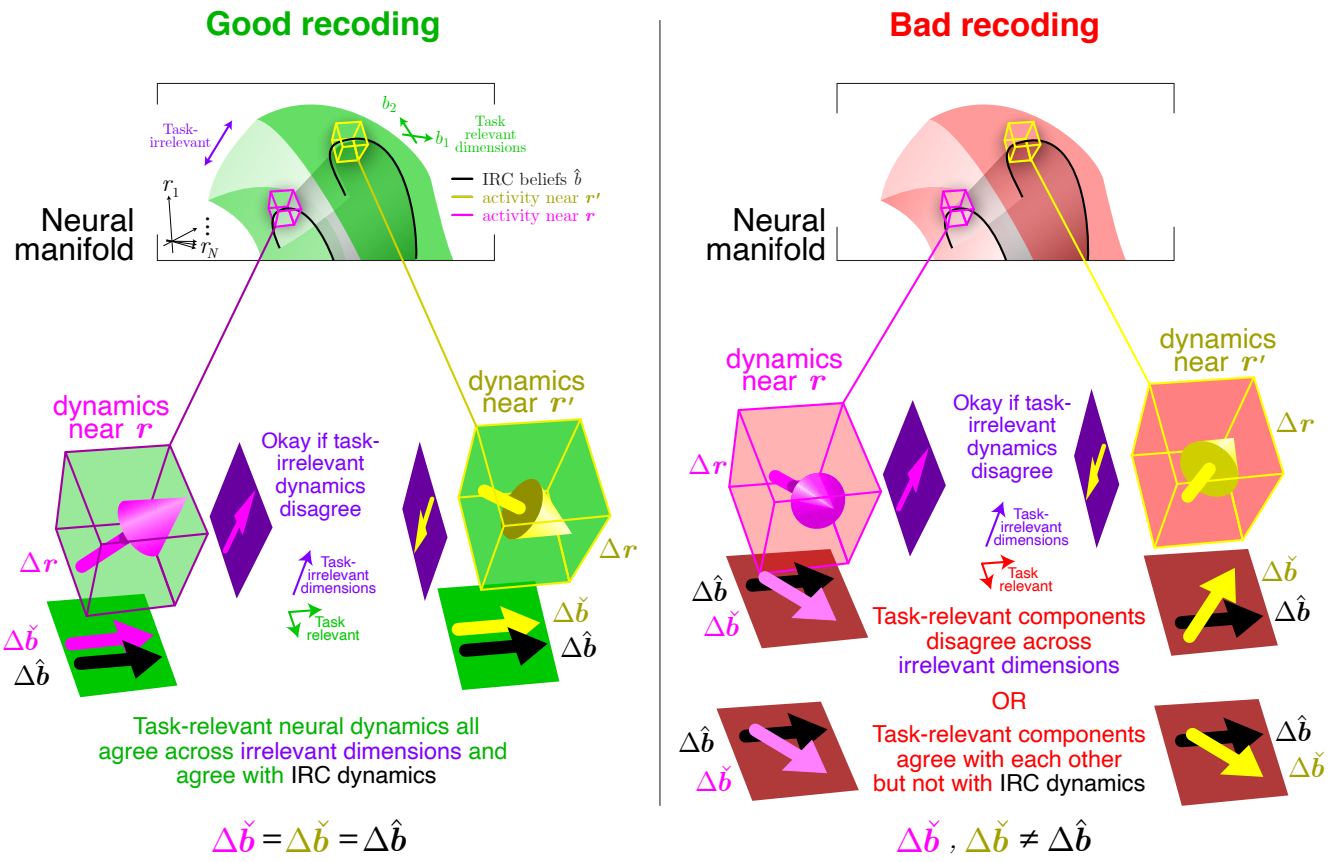

Fig. S3. Expanding on Figure 5, the intrinsic neural dynamics define a vector field over the neural space, where the task variables evolve according to $\dot{\boldsymbol{r}} = \boldsymbol{f}(\boldsymbol{r}) + \boldsymbol{\xi}$ where $\boldsymbol{\xi}$ is a stochastic component that may depend on $\boldsymbol{r}$. An ideal dimensionality reduction from neural responses $\boldsymbol{r}$ (green volumes) to the task-relevant variables $\boldsymbol{b}$ must preserve the task-relevant dynamics, such that $d\boldsymbol{\varphi}_{\text{enc}}(\boldsymbol{r})/dt = \boldsymbol{f}_{\text{dyn}}(\boldsymbol{\varphi}_{\text{enc}}(\boldsymbol{r})) + \boldsymbol{\eta}$ where $\boldsymbol{\eta}$ contains only the stochastic elements of $\boldsymbol{r}$ in the task-relevant directions. This would mean that the vector-valued updates $\Delta\check{\boldsymbol{b}}(\boldsymbol{r})$ to the neural beliefs would be consistent across the (purple) task-irrelevant dimensions (yellow and magenta neural belief update vectors agree), and would also agree with the behaviorally inferred belief updates $\Delta\hat{\boldsymbol{b}}$ (black vectors). However, due to stochasticity, limitations in our discovery of the ideal dimensionality reduction, or a mismatch between our target behavioral model and the brain's true model, we may find a bad representation of the task space (red volumes) for which the (yellow or magenta) task-relevant updates depend on the (purple) task-irrelevant dimensions. Two estimates of task-relevant dimensions can even have the same cross-validated encoding errors while exhibiting different dynamics with different recoding errors.

Table S1. Glossary of notation.

symbol	meaning	symbol	meaning
t	time		
s	world state	$T(s'\|s,a)$, $\overline{T}(b'\|b,a)$	transition probability
o	observation	$O(o\|s)$, $\overline{O}(o\|b,a)$	observation probability
b	belief	$B(s_t\|o_{1:t}, a_{1:t-1})$	posterior
a	action	$\pi(a\|b)$	policy
$\boldsymbol{r}$	neural responses	$R(s,a)$, $\overline{R}(b,a)$	reward
x^*	true world variable	Q	state-action value
x	agent's actual assumption	$\mathcal{Q}$	auxiliary function in EM
$\hat{x}$	estimate from behavior	$\mathcal{L}$	log-likelihood
$\check{x}$	estimate from neurons	L	loss
$\check{\varphi}_{\mathrm{enc}}$	estimate from encoding: $\boldsymbol{r} \to \check{b}$		
$\check{f}_{\mathrm{rec}}$	recoding / neural dynamics: $\check{b} \to \check{b}$	$\hat{f}_{\mathrm{dyn}}$	behavioral dynamics: $\hat{b} \to \hat{b}$
$\check{\pi}_{\mathrm{dec}}$	decoding / neural policy: $\check{b} \to a$	$\hat{\pi}_{\mathrm{act}}$	behavioral policy: $\hat{b} \to a$

References

1. R Bellman, *Dynamic programming*. (Princeton University Press), (1957).
2. NM Houlsby, et al., Cognitive tomography reveals complex, task-independent mental representations. *Curr. Biol.* **23**, 2169–2175 (2013).
3. S Daptardar, S Paul, X Pitkow, Inverse Rational Control with partially observable nonlinear dynamics. *arXiv preprint arXiv:1908.04696* (2019).
4. AP Dempster, NM Laird, DB Rubin, Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. Ser. B (methodological)*, 1–38 (1977).
5. D Hernando, V Crespi, G Cybenko, Efficient computation of the hidden markov model entropy for a given observation sequence. *IEEE transactions on information theory* **51**, 2681–2685 (2005).
6. CM Bishop, *Pattern recognition and machine learning*. (Springer), (2006).