Rational Thoughts in Neural Codes

Zhengwei Wu[a,b,c], **Minhae Kwon**[a,b,c], **Saurabh Daptardar**[b,d], **Paul Schrater**[e], **and Xaq Pitkow**[a,b,c]

This manuscript was compiled on September 11, 2019

Complex behaviors are often driven by an internal model, which integrates sensory information over time and facilitates long-term planning to reach subjective goals. We interpret behavioral data by assuming an agent behaves rationally — that is, they take actions that optimize their subjective reward according to their understanding of the task and its relevant causal variables. We apply a new method, Inverse Rational Control (IRC), to learn an agent's internal model and reward function by maximizing the likelihood of its measured sensory observations and actions. This thereby extracts rational and interpretable thoughts of the agent from its behavior. We also provide a framework for interpreting encoding, recoding and decoding of neural data in light of this rational model for behavior. When applied to behavioral and neural data from simulated agents performing suboptimally on a naturalistic foraging task, this method successfully recovers their internal model and reward function, as well as the computational dynamics within the neural manifold that represents the task. This work lays a foundation for discovering how the brain represents and computes with dynamic beliefs.

Cognition | Neuroscience | Computation | Rational | Neural coding

Understanding how the brain works requires interpreting neural activity. The behaviorist tradition (1) aims to understand the brain as a black box solely from its inputs and outputs. Modern neuroscience has been able to gain major insights by looking inside the black box, but still largely relates measurements of neural activity to the brain's inputs and outputs. While this is the basis of both sensory neuroscience and motor neuroscience, most neural activity supports computations and cognitive functions that are left unexplained — we might call these functions 'thoughts'. To understand brain computations, we should relate neural activity to thoughts. The trouble is, how do you measure a thought?

Here we propose to model thoughts as dynamic beliefs that we impute to an animal, by combining explainable Artificial Intelligence (AI) cognitive models for naturalistic tasks with measurements of the animal's sensory inputs and behavioral outputs. We define an animal's task by the relevant dynamics of its world, observations it can make, actions it can take, and the goals it aims to achieve. The AI models that solve these tasks generate beliefs, their dynamics, and actions that reflect the essential computations needed to solve the task and generate behavior like the animal. With these estimated thoughts in hand, we propose an analysis of brain activity to find neural representations and transformations that potentially implement these thoughts.

Our approach combines the flexibility of complex neural network models while maintaining the interpretability of cognitive models. It goes beyond black-box neural network models that solve one particular task and find representational similarity with the brain (2–4). Instead, we solve a whole family of tasks, and then find the task whose solution best describes an animal's behavior. We then associate properties of this best-matched task with the animal's mental model of the world, and call it 'rational' since it is the right thing to do under this internal model of the world. Our method explains behavior and neural activity based on underlying latent variable dynamics, but it improves upon usual latent variable methods for neural activity that just compress data without regard to tasks or computation (5, 6). In contrast, our latent variables inherit meaning from the task itself, and from the animal's beliefs according to its internal model. This provides interpretability to both our behavioral and neural models.

We also want to ensure we can explain crucial neural computations that underlie ecological behavior in natural tasks. We can accomplish this by using tasks with key properties that ensure our model solutions implement these neural computations. First, a natural task should include latent or hidden variables: animals do not act directly upon their sensory data, as that data is merely an indirect observation of a hidden real world (7). Second, the task should involve uncertainty, since real-world sense data are fundamentally ambiguous and behavior improves when weighing evidence according to its reliability. Third, the relationships between latent variables and sensory evidence should be nonlinear in the task, since if linear computation were sufficient then animals would not need a brain: they could just wire sensors to muscles and compute the same result in one step. Fourth, the task should have relevant temporal dynamics, since actions affect the future, and useful properties of the world change; animals must account for this.

While natural tasks that animals perform every day do indeed have these properties, most neuroscience studies isolate a subset of them for simplicity. Although this has revealed important aspects of neural computation, it also potentially misses some of the fundamental structure of brain computation. Recent progress warrants increasing the naturalism and complexity of the tasks and models.

One major challenge for practical studies with increased complexity and naturalism is to record from many neurons with enough spatial and temporal precision to reveal the relevant computational dynamics for these tasks. Specifically, the dimensionality of neu-

[a]Baylor College of Medicine, Department of Neuroscience; [b]Rice University, Department of Electrical and Computer Engineering; [c]Baylor College of Medicine, Center for Neuroscience and Artificial Intelligence; [d]Google Maps; [e]University of Minnesota, Departments of Psychology and Computer Science

Conceptual framework: XP, PS. Discrete control: ZW, XP, PS. Continuous control: SD, MK, XP, PS. Neural simulations: ZW. Neural analysis: ZW, MK, XP. Initial draft: XP, ZW, SD. Editing: XP, ZW, SD, PS. Funding acquisition: XP, PS.

[2]To whom correspondence should be addressed. E-mail: xaq@rice.edu

ral data needs to be bigger than the dimensionality of our target tasks (8). Modern neurotechnology now affords us this opportunity: brain-wide calcium imaging at cellular resolution and fine-grained electrophysiological recording can record from thousands of neurons simultaneously at high frequency. Limited experimental time and coverage still hinder our ability to explore the neural representations. But with current large-scale neural data, we will increasingly have enough power to find neural representations and dynamics in naturalistic and cognitively interesting tasks.

This paper makes progress towards understanding how the brain produces complex behavior by providing methods to estimate thoughts and interpret neural activity. We first describe a model-based technique we call Inverse Rational Control for inferring latent dynamics which could underlie rational thoughts. Then we offer a theoretical framework about neural coding that shows how to use these imputed rational thoughts to construct an interpretable description of neural dynamics.

We illustrate these contributions by analyzing a task performed by an artificial brain, showing how to test the hypothesis that a neural network has an implicit representation of task-relevant variables that can be used to interpret neural computation. We choose an ecologically relevant foraging task that requires sensitivity to past rewards, current observations, and an internal memory state. Our approaches should serve as valuable tools for interpreting behavior and brain activity for real agents performing naturalistic tasks.

Results

Modeling behavior as rational. In an uncertain and partially observable environment, animals learn to plan and act based on limited sensory information and subjective values. To better understand these natural behaviors and interpret their neural mechanisms, it would be beneficial to estimate the internal model and reward function that explains animals' behavioral strategies. In this paper, we model animals as rational agents acting optimally to maximize their own subjective rewards, but under a family of possibly incorrect assumptions about the world. We then invert this model to infer the agent's internal assumptions and rewards and estimate the dynamics of internal beliefs. We call this approach Inverse Rational Control (IRC), because we infer the reasons that explain an agent's suboptimal behavior to control its environment.

This method creates a probabilistic model for an agent's trajectory of observations and actions, and selects model parameters that maximize the likelihood of this trajectory. We make assumptions about the agent's internal model, namely that it believes that it gets unreliable sensory observations about a world that evolves according to known stochastic dynamics. Finally, we assume that the agent's actions are chosen to maximize its own subjectively expected long-term utility. This utility includes both benefits, such as food rewards, and costs, such as energy consumed by actions; it should also account for internal states describing motivation, like hunger or fatigue, that modulate the subjective utility. We then use the agent's sequence of observations and actions to learn the parameters of this internal model for the world. Without a model, inferring both the rewards and latent dynamics is an underdetermined problem leading to many degenerate solutions. However, under reasonable model constraints, we demonstrate that the agent's reward functions and assumed dynamics can be identified. Our learned parameters includes the agent's assumed stochastic dynamics of the world variables, the reliability of sensory observations about those world states, and subjective weights on action-dependent costs and state-dependent rewards.

Partially Observable Markov Decision Process. To define the Inverse Rational Control problem, we first formalize the agent's task as a Partially Observable Markov Decision Process (POMDP, Figure 1A) (9), a powerful framework for modeling agent behavior under uncertainty. A Markov chain is a temporal sequence of states $s \in \mathcal{S}$ for which the transition probability T to the next state depends only on the current state, not on any earlier ones: $T(s_{t+1}|s_{0:t}) = T(s_{t+1}|s_t)$. A Markov Decision Process (MDP) is a Markov chain where at each time an agent can influence the world state transitions by deciding to take an action $a \in \mathcal{A}$, according to $T(s_{t+1}|s_t, a_t)$. At each time step the agent receives a reward or incurs a cost (negative reward) that depends on the world state and action, $R(s_t, a_t)$. The agent's goal is to choose actions that maximize its value V, measured by total expected future reward (negative cost) with a temporal discount factor $\gamma \in (0, 1)$, so that $V = \left\langle \sum_{t=1}^{\infty} \gamma^t R(s_t, a_t) \right\rangle_{p(s_{1:\infty}, a_{1:\infty})}$. The actions are drawn from a state-dependent probability distribution called a policy, $\pi(a|s_t)$, which may be concentrated entirely on one action or may have some width. In a normal MDP, the agent can fully observe the current world state, but must plan for an unknown future. In a Partially Observed MDP (POMDP), the agent again does not know the future, but does not even know the current world state exactly. Instead the agent only gets unreliable observations $o \in \Omega$ about it, drawn from the distribution $o_t \sim O(o|s_t)$. The agent's goal is the same, to maximize the total expected temporally-discounted future reward. The POMDP $\mathcal{M}$ is then a tuple of all of these mathematical objects: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \Omega, R, T, O, \gamma)$. Different POMDPs tuples reflect different tasks.

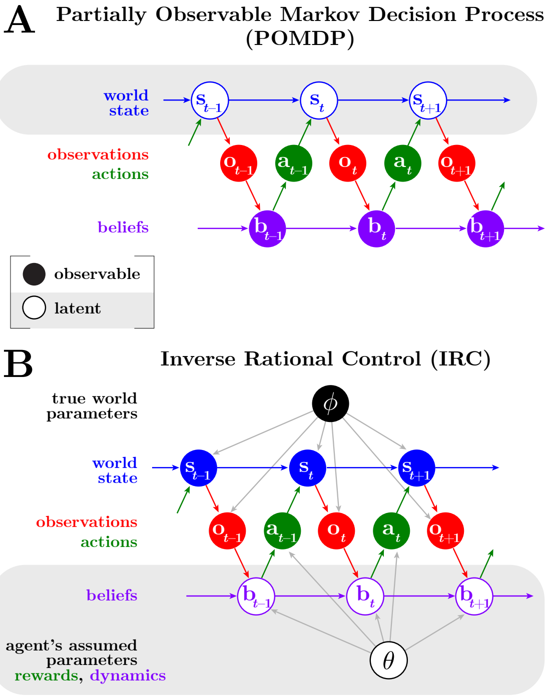

Fig. 1. Graphical model of a Partially Observable Markov Decision Process (POMDP) (**A**) and the Inverse Rational Control (IRC) problem (**B**). Empty circles denote latent variables, and solid circles denote observable variables. For the POMDP, the agent knows its beliefs but must infer the world state. For IRC, the scientist knows the world state but must infer the beliefs. The real world dynamics depends on parameters ϕ, while that of the agent assumes parameters θ which include both its assumption about the stochastic world dynamics and its own subjective rewards and costs.

Optimal solution of a POMDP requires the agent to compute a time-dependent posterior probability over the possible current world state, given its history of observations and actions. All of that history can be summarized concisely in a single distribution, the posterior B. It is useful to define a belief state b as completely summarizing the posterior, so we can write $B(s_t|b_t) = B(s_t|o_{1:t}, a_{0:t-1})$. This belief state can be expressed recursively using the Markov property as a function of its previous value (Supplemental Information):

$$b_t = \frac{1}{Z} O(o_t|s_t) \left\langle T(s_t|s_{t-1}, a_{t-1}) \right\rangle_{s_{t-1}|b_{t-1}} \qquad [1]$$

where Z is a normalization constant.[*]

We can express the entire partially observed MDP as a fully-observed MDP called a Belief MDP, where the relevant fully-observed state is not the world state s but instead the agent's own belief state b (10). To do so, we must re-express the transitions and rewards as a function of these belief states, $\overline{T}(b_{t+1}|b_t, o_{t+1}, a_t)$ and $\overline{R}(b_t, a_t)$, as described in the Supplemental Information.

The optimal agent then determines a value function $Q(b, a)$ over this belief space and allowed actions, based on its own subjective rewards and costs. This value can be computed recursively through the Bellman equation (11)

$$\begin{aligned} Q(b_t,a_t) = &\ \overline{R}(b_t, a_t)+ \qquad [2] \\ &\gamma \iint da_{t+1} db_{t+1}\, \overline{T}(b_{t+1}|b_t, a_t) \pi(a_{t+1}|b_{t+1}) Q(b_{t+1}, a_{t+1}) \end{aligned}$$

The optimal policy deterministically selects whichever action maximizes the value $Q(b, a)$. An alternative stochastic policy samples from a softmax function over actions, $\pi(a|b) \sim \frac{1}{Z} \exp\left(Q(b, a)/\tau\right)$with a temperature parameter τ and normalization constant Z. The randomness introduces a new sub-optimality to the agent: instead of choosing the action with the maximal value, the agent has some chance of choosing a worse action. In the limit of a low temperature τ we recover the optimal policy, but a real agent may be better described by a stochastic policy with some controlled exploration.

Inverse Rational Control. Despite the appeal of optimality, animals rarely appear optimal in experimentally defined tasks, and not just by exhibiting more randomness. Short of optimality, what principled guidance can we have about an animal's actions that would help us understand its brain? One possibility is that an animal is 'rational' — that is, optimal for different circumstances than those being tested. In this section we present a behavioral analysis based on the possibility that agents are rational in this sense. The core idea is to parameterize possible strategies of an agent by those tasks under which each is optimal, and find which of those best explains the behavioral data.

We specify a *family* of POMDPs where each member has its own task dynamics, observation probabilities, and subjective rewards, together constituting a parameter vector θ. These different tasks yield a corresponding family of optimal agents, rather than a single optimized agent. We then define a log-likelihood over the tasks in this family, given the experimentally observed data and marginalized over the agent's latent beliefs (Figure 1B):

$$\mathcal{L}(\theta) = \log \int db_{1:T}\, p(b_{1:T}, o_{1:T}, a_{1:T}, s_{1:T}|\theta, \phi) \qquad [3]$$

In other words, we find a likelihood over *which tasks* an agent solves optimally. In [3] ϕ are known parameters in the experimental setup that determine the world dynamics. Since they only affect observed quantities in the graphical model, they do not affect the model likelihood over θ (Supplementary Information).

This mathematical structure connects interpretable models directly to experimentally observable data. We can now formalize important scientific problems in behavioral neuroscience. For example, we can maximize the likelihood to find the best interpretable explanation of an animal's behavior as rational within a model class, as we show below. We can also compare categorically different model classes that attribute to the agent different reward structures or assumptions about the task.

The log-likelihood [3] seems complicated, as it depends on the entire sequence of observations and actions and requires marginalization over latent beliefs. Nonetheless it can be calculated using the Markov property of the POMDP: the actions and observations constitute a Markov chain where the agent's belief state is a hidden variable. We show that it is possible to exploit this structure to compute this likelihood efficiently (Supplemental Information).

Challenges and solutions for rationalizing behavior. To solve the IRC problem, we need to parameterize the task, beliefs, and policies, and then we need to optimize the parameterized log-likelihood to find the best explanation of the data. This raises practical challenges that we need to address.

Our core idea for interpreting behavior is to parameterize everything in terms of tasks. All other elements of our models are ultimately referred back to these tasks. Consequently, the beliefs and transitions are distributions over latent task variables, the policy is expressed as a function of task parameters and preferences, and the log-likelihood is a function of the task parameters that we assume the agent assumes.

Thus, whatever representations we use for the belief space or policy, we need to be able to propagate our optimization over the task parameters through those representations. This is one requirement for practical solutions of IRC. A second requirement is that we can actually compute the optimal policies.

Efficient representation of general beliefs and transitions is hard since the space of probabilities is much larger than the state space it measures. The belief state is a probability distribution and thus takes on continuous values even for discrete world states. For continuous variables the space of probabilities is potentially infinite-dimensional. This poses a substantial challenge both for machine learning and for the brain, and finding neurally plausible representations of uncertainty is an active topic of research (12–17). We consider two simple methods to solve IRC using lossy compression of the beliefs: discretization, or distributional approximation. We then provide a concrete example application in the discrete case.

Discrete beliefs and actions. If we have a discrete state space then we can use conventional solution strategies for Markov Decision Processes. For a small enough world space, we can exhaustively discretize the complete belief space, and then solve the Belief MDP problem with standard MDP algorithms (11, 18). In particular, the state-action value function $Q(b, a)$ under a softmax policy $\pi(a|b)$ can be expressed recursively by a Bellman equation, which we solve using value iteration (10, 11). The resultant value function then determines the softmax policy π, and thereby determines the policy-dependent term in the log-likelihood [3].

Finally, to solve the IRC problem we can directly optimize this log-likelihood, for example by greedy line search (Supplementary Information). An alternative in higher-dimensional problems is

[*] A minor notational point is that we assume that the agent is a function of the belief state, either because it is a deterministic function of the belief or because the stochastic output action is fully observed or appended to the belief state. Then we can write $B(s_t|o_t, a_{t-1}, b_{t-1}) = B(s_t|o_t, b_{t-1})$. This justifies the omission of an arrow in Figure 1A from $a_{t-1} \to b_t$. Alternatively we can allow the action to be partially observable and add another arrow to that figure.

to use Expectation-Maximization to find a local optimum, with a gradient ascent M-step (Supplementary Information, (19, 20)). To compute the gradient of the log-likelihood, we again use recursion to calculate the value gradient $\partial Q/\partial\theta$ exactly, and use the chain rule to derive the policy gradient and then the $\mathcal{Q}$ auxiliary function gradient (Supplementary Information).

Continuous beliefs and actions. The computational expense of the discrete solution grows rapidly with problem size, and become intractable for continuous state spaces and continuous controls. A practical choice is to approximate posteriors by a finite set of summary statistics, and update them by a method like expectation propagation (21). The simplest example is to use quadratic statistics, *i.e.* Gaussian posterior. This belief state can then be updated according to an extended Kalman filter that accounts for the agent's internal model of the stochastic nonlinear dynamics.For more general belief representations, the belief update equations may require additional flexibility.

Rational control with continuous actions also requires us to implement a family of continuous policies π that map from beliefs to actions. We use deep neural networks to implement these policies (22) through an actor-critic method (Deep Deterministic Policy Gradient, (23)), by which one 'critic' network estimates the value of each action taken by the 'actor' network.

Deep learning methods are commonly used in reinforcement learning to provide flexibility, but they lack interpretability: information about the policy is distributed across the weights and biases of the network. Crucially, to maintain interpretability, we parameterize this family by the task. Specifically, we provide the model parameters as *additional inputs* to a policy network, and learn the optimal policies simultaneously over a prior distribution on task parameters $p(\theta)$ (22). This allows the network to generalize its optimal strategies across POMDPs in the task family. It also allows us to compute policy gradients simply using auto-differentiation, which we exploit when optimizing the log-likelihood to find the parameters that best match for an agent's behavior.

Ultimately, after optimizing the log-likelihood for either discrete or continuous representations, the end result is a set of parameters θ that best explain the observed behavioral data, and define the agent's assumed internal task model and subjective preferences. Within this model class, we have therefore found the best rational explanation for the agent's behavior.

Finding a neural code for rational thoughts. We don't presume that any real brain explicitly calculates a solution to the Bellman equation, but rather learns a policy by combining experience and mental modeling. With enough training, the result is an agent that behaves 'as if' it were solving the POMDP (Figure 2A).

If an animal's behavior is well-described as depending on latent beliefs, as we assume in Inverse Rational Control, then it makes sense that we should find neural correlates of these beliefs in the brain. If we can find such correlates, does this mean that the neurons encode or represent those beliefs? Some have argued that the notion of a neural code is a poor metaphor because it captures neither the causal or mechanistic structure of the brain, nor its relation to actions and affordances (24–26). For example, it may be that the brain does not *use* the neural signals that a neuroscientist can use to extract information about a task.

In contrast, here we argue instead that the linked processes of encoding, recoding, and decoding can be a useful way of explaining task-relevant computation in the brain at the algorithmic or representational level (27). The brain's 'encoding' specifies how neural activity can be used to estimate task variables (Figure 2B), including both rewarded variables and irrelevant or nuisance ones that must be disentangled from them. 'Recoding' describes how that encoding is transformed over time and space by neural processing (Figure 2C). 'Decoding' describes how those estimates predict future actions (Figure 2D).

(In our use of these terms, we are taking the brain's perspective. The term 'decoding' more often reflects the scientist's perspective, where the scientist decodes brain activity to estimate encoding quality. Instead, we reserve the term decoding to describe how neural activity affects actions: we say that the brain decodes its own activity to generate behavior.)

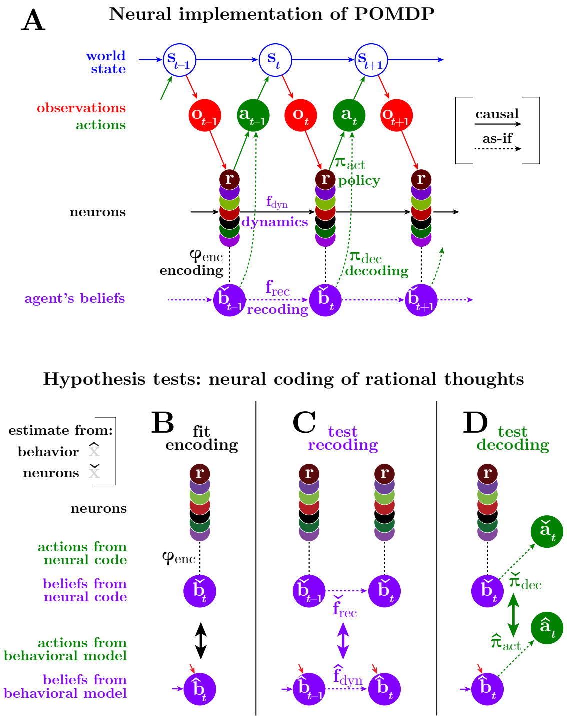

Fig. 2. Schematic for analyzing a dynamic neural code. (**A**) Graphical model of a POMDP problem with a solution implemented by neurons implicitly encoding beliefs. (**B**) We find how behaviorally relevant variables (here, beliefs) are *encoded* in measured neural activity through the function $\check{b}_t = \varphi_{\text{enc}}(\boldsymbol{r}_t)$. (**C**) We then test our hypothesis that the brain *recodes* its beliefs rationally by testing whether the dynamics of the behaviorally estimated belief $\hat{b}$ match the dynamics of the neurally estimated beliefs $\check{b}$, as expressed through the update dynamics $\hat{f}_{\text{dyn}}(\hat{b}_t, o_t)$ and recoding function $\check{f}_{\text{rec}}(\check{b}_t, o_t)$. (**D**) Similarly, we test whether the brain *decodes* its beliefs rationally by comparing the behaviorally and neurally derived policies $\hat{\pi}_{\text{act}}$ and $\check{\pi}_{\text{dec}}$. Quantities estimated from behavior or from neurons are denoted by up-pointing or down-pointing hats, ˆ and ˇ (Table S1).

This level of explanation need not capture every facet of neural responses nor the mechanism by which they evolve. Obviously it cannot explain responses to untested task variables. Nonetheless, it would be great progress if we can account for stimulus- and action-dependent neural dynamics within a task-relevant submanifold (28) that explains how pieces of information interact and predict behavior. Although this 'as-if' description cannot legitimately claim to be causal, it can be promoted to a causal description since it does provide useful predictions for causal tests about what neural features should influence computation and action (29, 30).

Next we describe the general structure of such a representation-level explanation. We then follow this approach to analyze an artificial brain performing a specific foraging task.

To begin the analysis, we propose to use Inverse Rational

Control to construct a behavioral model from the sensory inputs and actions that we observe the agent makes. The inferred internal model allows us to impute the agent's time-dependent beliefs b about the partially observed world state s. Such a belief vector might include full posterior over the world state, $B(s_t|o_{1:t}, a_{1:t-1})$ as we used for the discrete IRC above, or a point estimate $\hat{s}$ of the world state and a measure of uncertainty about it, say a covariance Σ_s, as in the Gaussian approximation we used for continuous IRC. To us, as scientists, the agent's beliefs are latent variables, so our algorithm can at best create a posterior $p(b)$ over those beliefs, or a point estimate $\hat{b}$ indicating the most probable belief. Here we will base our analyses on a point estimate over beliefs. First we will describe the general approach, and then we will apply this approach in an example analysis.

Encoding. First we aim to find the brain's encoding of the beliefs about latent variables. Specifically, we look for neural correlates of the estimate $\hat{b}$ of the agent's beliefs that we inferred by IRC.

While there is little doubt that real behavior is influenced by uncertainty (31–34), it remains unclear how uncertain beliefs that influence actions are encoded by the brain. These beliefs could be represented in the brain in a multitude of ways,and resolving this question is an active topic of current research. One reason it is hard to make progress on this topic is that we cannot measure the agent's beliefs directly, except by assuming optimal inferences. IRC gives us a way to estimate suboptimal beliefs, so we can examine how the brain represents them.

Given beliefs $\hat{b}_t$ imputed by IRC, we can estimate how they are encoded in the neural responses $\boldsymbol{r}$ using a (potentially nonlinear, potentially spatiotemporal) readout function $\varphi_{\rm enc}(\boldsymbol{r}_t)$. This can be accomplished by minimizing an encoding loss such as $L_{\rm enc} = \sum_t \ell_{\rm enc}\left[\hat{b}_t, \varphi_{\rm enc}(\boldsymbol{r}_t)\right]$ where at each time $\ell_{\rm enc}$ measures the distance between the behavioral target belief $\hat{b}_t$ and the neural estimate $\check{b}_t = \varphi_{\rm enc}(\boldsymbol{r}_t)$ (Figure 2B). After training $\varphi_{\rm enc}$ using the behavioral targets $\hat{b}$, we can then cross-validate it on new estimates $\check{b}$ from fresh neural data. (Estimates based on the behavioral model are consistently denoted by an up-pointing hat, $\hat{x}$, as distinguished from estimates based on the neural responses denoted by a down-pointing hat, $\check{x}$, as indicated in Table S1.)

Recoding. While neural dynamics may affect every dimension of neural activity, we focus only on the interpretable dynamics within the lower-dimensional task manifold. By construction, those dynamics reflect the changes in the agent's beliefs.

The rational control model predicts that beliefs are updated by sensory observations and past beliefs, with interactions that are determined by the internal model according to a function $b_{t+1} = f_{\rm dyn}(b_t, o_t) + \eta_t$ where $f_{\rm dyn}$ and η_t reflect the deterministic and stochastic parts of the dynamics. If our neural analysis correctly identifies dynamics responsible for behavior, then the beliefs $\check{b}$ estimated from the neural encoding should be recoded over time following those same update rules. We estimate this neural recoding function $\check{f}_{\rm rec}(\check{b}_t, o_t)$ directly from the sequence of neurally estimated beliefs $\check{b}$ by minimizing a recoding prediction loss, such as $L_{\rm rec} = \sum_t \ell_{\rm rec}\left[\check{b}_{t+1}, \check{f}_{\rm rec}(\check{b}_t, o_t)\right]$ where $\ell_{\rm rec}$ penalizes differences between the actual and predicted future beliefs. We then compare $\check{f}_{\rm rec}$ to the update dynamics posited by the behavioral model $\hat{f}_{\rm dyn}$ (Figure 2C). (We should compare these only over the distribution of experienced beliefs, *i.e.* those beliefs for which the recoding function matters in practice.) Agreement between these recoding functions implies that we have successfully understood the 'recoding' process. Even for good encoding models this is not guaranteed, since activity outside the encoding manifold could influence the neural dynamics.

The encoding dimensions may seem to change over time or context (2, 35). Perhaps this too should count as recoding, such that our approach of estimating beliefs from neural activity using a nonadaptive function $\varphi_{\rm enc}(\boldsymbol{r})$ would then miss important computations. However, this only indicates that our way of measuring the encoding is too limited. The real encoding model could be fixed but nonlinear (36), and can appear adaptive when measured by an inadequate model (37). More complex functions are harder to fit but the brain's neural code may require this added complexity.

Decoding. These encodings and recodings do not matter if the brain never decodes that information into behavior. We can evaluate how the brain uses this information by predicting actions from the neurally encoded beliefs, minimizing a decoding loss between observed actions and distribution of actions $\check{a}$ predicted from neurally estimated belief by the policy $\check{\pi}(\check{a}|\check{b})$: $L_{\rm dec} = \sum_t \ell_{\rm dec}\left[a_t, \check{\pi}_{\rm dec}(\check{a}|\check{b}_t)\right]$ where $\ell_{\rm dec}$ penalizes actions that are unexpected according to the given policy. We then test the hypothesis that the brain decodes neurally encoded rational thoughts by comparing the neurally-derived policy $\check{\pi}_{\rm dec}$ against the behavioral policy, $\hat{\pi}_{\rm act}$ (Figure 2D).

Application to Foraging. We applied our analyses to understand the workings of a neural network performing a foraging task. The task requires an agent to combine unreliable sensory data with an internal memory to infer when and where rewards are available, and how to best acquire them. We train an artificial recurrent neural network to solve this task in a suboptimal but rational way, use Inverse Rational Control to infer its assumptions, subjective preferences, and beliefs, and then analyze its neural responses to test our coding framework.

Task description. Two locations ('feeding boxes') have hidden food rewards that appear and disappear according to independent telegraph processes with specified transition probabilities (Figure 3, (38)). The boxes provide unreliable color cues about the current reward availability, ranging from blue (probably unavailable) to red (probably available).

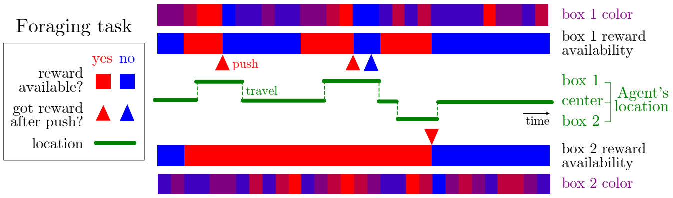

Fig. 3. Illustration of foraging task with latent dynamics and partially observable sensory data. The reward availability in each of two boxes evolves according to a telegraph process, switching between available (red) and unavailable (blue), and colors give the animal an ambiguous sensory cue about the reward availability. The agent may travel between the locations of the two boxes. When a button is pressed to open a box, the agent receives any available reward.

We assume there are three possible locations for the agent: the locations of boxes 1 and 2, and a middle location 0. We include a small 'grooming' reward for staying at the middle location, to allow the agent to stop and rest. A few discrete actions are available to the agent: it can push a button to open a box to either get reward or observe its absence, it can move toward a new location, or it can do nothing. Traveling and pushing a button to open the box each have an associated cost. This disincentivizes the agent from repeating fruitless actions. When a button-press action is taken to open a box, any available reward there is acquired. Afterwards, the animal knows there is no more food available now in the box (since it was either unavailable or consumed) and the belief about food availability in that box is reset to zero.

Neural network agent. We first create a rational agent that solves a POMDP problem in this family, and then we use supervised learning to train a nonlinear recurrent neural network to match the belief dynamics and policy of that agent.

To create the rational agent, we discretize beliefs about reward availability for each box into $N = 10$ belief states. We define the transition matrix in the discretized belief space by binning the continuous transition matrix $\overline{T}(b_{t+1}|b_t, a_t)$. We allow 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.

Figure S1A shows the architecture of our recurrent network. After training to match the rational agent, readouts of the neural activities closely match the POMDP agent's beliefs and policies (Figure S1B,C), but these task-relevant quantities are encoded implicitly in a large population of neurons.

We then collected sensory observations and actions from the neural network agent while it was challenged by a different task than the one for which it was optimized (Methods). These inputs led to a time series of observations o_t, actions a_t, and neural activity $\boldsymbol{r}_t$. Together these constitute the experimental measurements.

Inverse Rational Control for foraging. We don't know the agent's assumed world parameters, nor do we know the agent's subjective costs, nor the amount of randomness (softmax temperature). Our goal is to estimate a simulated agent's internal model and belief dynamics from its chosen actions in response to its sensory observations. We infer all of these using IRC.

The actions and sensory evidence (color cues, locations and rewards) obtained by the agent all constitute observations for the experimenter's learning of the agent's internal model. Based on these observations over 1000 time points, including 364 movements and 109 button presses, we use IRC to infer the parameters of the internal model that can best explain the behavioral data (Figure 4A). The comparison between the true parameters and the estimated parameters are shown in Figure 4B.

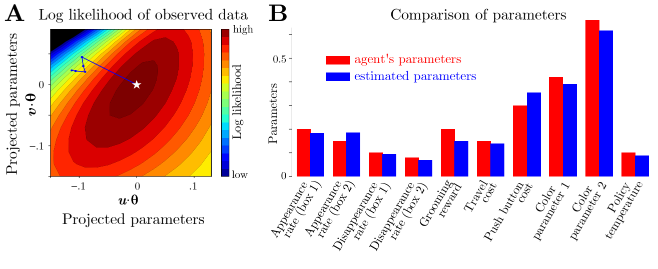

Fig. 4. Fitting internal model and subjective rewards using IRC. **A**: The estimated parameters converge to the optimal point of the observed data log-likelihood. Since the parameter space is high dimensional, we project it onto the first two principal components $\boldsymbol{u}, \boldsymbol{v}$ of the learning trajectory for θ. **B**: Comparison of the true parameters of the agent and the estimated parameters.

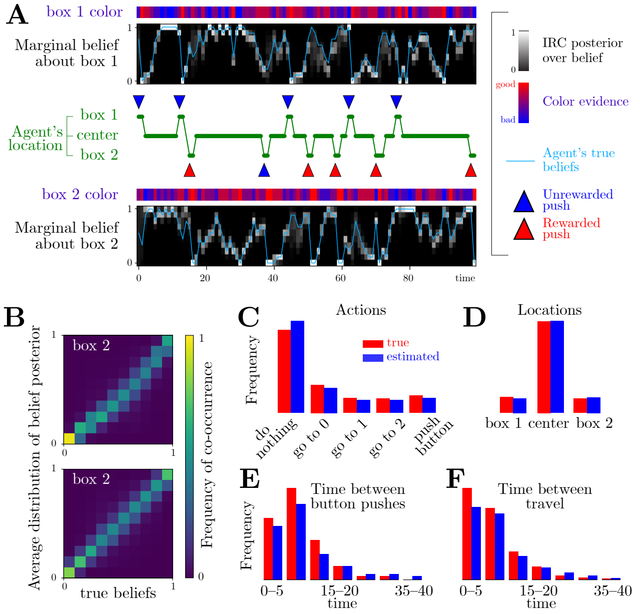

Fig. 5. Successful recovery of beliefs by Inverse Rational Control. **A**: Estimated and true marginal belief dynamics over latent reward availability. These estimates are informed by the noisy color data at each box and the times and locations of the agent's actions. The posteriors over beliefs are consistent with the dynamics of the true beliefs. **B**: The averaged posteriors of the estimated beliefs $\hat{b}_t$, $\frac{1}{T}\sum_t^T p(\hat{b}_t|a_{1:T}, o_{1:T})$, correlate strongly with the agent's true beliefs. Inferred distributions of (**C**) actions, (**D**) residence times, (**E**) intervals between consecutive button-presses, and (**F**) intervals between movements.

Data limitations imply some discrepancy between the true parameters and the estimated parameters which can be reduced with more data. With the estimated parameters, we are able to infer a posterior over the dynamic beliefs (Figure 5A). (Note that this is an experimenter's posterior over the agent's subjective posterior!) The inferred posterior is consistent with the agent's true subjective probability of the food availability in each box. The inferred distributions over beliefs reveals strong correlations between the true and estimated belief state (Figure 5B).

Figure 5C–F shows that the artificial brain and inferred agent choose actions with similar frequencies, occupy the three locations for the same fraction of time, and wait similar amounts of time between pushing buttons or travelling. This demonstrates that the IRC-derived agent's internal model generates behaviors that are consistent with behaviors of the agent from which it learned.

Neural analysis of rational foraging. We can now use our neural coding framework to look inside the brain.

We assume that beliefs b_t are linearly encoded instantaneously in neural activity $\boldsymbol{r}_t$. For our example synthetic brain, this is correct by construction. After performing linear regression of behaviorally derived beliefs $\hat{b}$ against neural activity $\boldsymbol{r}$, we can estimate other beliefs $\check{b}$ from previously unseen neural data. Figure 6A shows that these beliefs estimated from neural data are accurate.

Figure 6B shows that the recoding dynamics obtained from the neural belief dynamics also match the dynamics described by the rational model. We characterize these neural dynamics using kernel ridge regression between $\check{b}_t$ and $\check{b}_{t+1}$ (Methods). The resultant temporal changes in the neurally-derived beliefs $\Delta\check{b}_t = \check{f}_{\mathrm{rec}}(\check{b}_t, o_t) - \check{b}_t$ agree with the corresponding changes in the behavioral model beliefs, $\Delta\hat{b}_t = \hat{f}_{\mathrm{dyn}}(\hat{b}_t, o_t) - \hat{b}_t$. Although some of these changes are driven directly by the sensory observations (colors), that only explains part of the belief updates: even

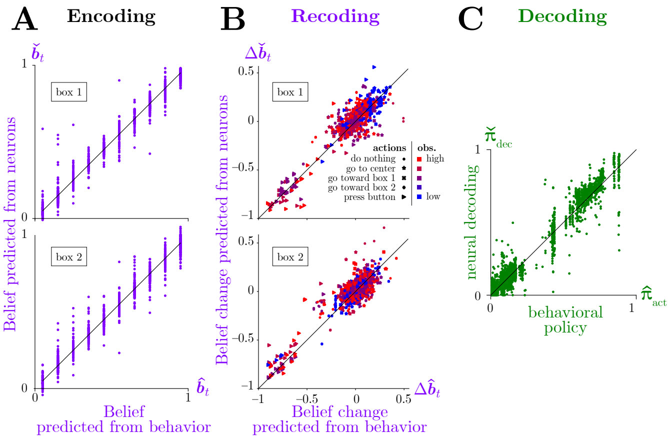

Fig. 6. Analysis of neural coding of rational thoughts. (**A**) Neurally estimated beliefs $\check{b}$ match the true belief projection of the neural network activity, $b = W\boldsymbol{r}$ for a trained matrix W. (**B**) Belief updates $\Delta\check{b}_t = \check{f}_{\text{rec}}(\check{b}_t, o_t) - \check{b}_t$ from the neural recoding function $\check{f}_{\text{rec}}(\check{b}_t, o_t)$ match the corresponding belief updates from the task dynamics $\Delta\hat{b}_t = \hat{f}_{\text{dyn}}(\hat{b}_t, o_t) - \hat{b}_t$. (**C**) The policy $\check{\pi}_{\text{dec}}$ predicted from decoding neural beliefs approximately matches the policy $\hat{\pi}_{\text{act}}$ estimated from behavior by IRC.

conditioned on a given sensory input at one time, the updates agree between the neurons and the behavioral model. This provides evidence that we understand the internal model that governs recoding at the algorithmic level.

Similarly, our analysis of neural decoding uses nonlinear multinomial regression to fit the probabilities $\check{\pi}_{\text{dec}}(a|\check{b})$ of allowed actions as a function of neurally derived beliefs (Methods). A comparison of the resultant function to the rational policy $\hat{\pi}_{\text{act}}$ shows that these two decoding functions match reasonably well (Figure 6C). This provides evidence that we understand the decoding process by which task-relevant neural activity generates behavior.

Discussion

In this work we used an explainable AI paradigm to infer an internal model, latent beliefs, and subjective preferences of a rational agent that solves a POMDP. We fit the model by maximizing the likelihood of the agent's sensory observations and actions over a family of tasks. We then described a neural coding framework for testing whether the imputed latent beliefs encoded in a low-dimensional manifold of neural responses are recoded and decoded in a manner consistent with this behavioral model. We illustrated these two contributions by analyzing the neural coding of an implicit computational model by an artificial neural network trained to solve a simple foraging task requiring memory, evidence integration, and planning. For this simulated data, we successfully recovered the agent's internal model and subjective preferences, and found neural computations consistent with that model.

Related work. Our approach generalizes previous work in artificial intelligence on the inverse problem of learning agents by observing behavior. Methodologically, other studies of inverse problems address parts of Inverse Rational Control, but with a non-scientific goal — getting artificial agents to solve tasks by learning from demonstrations of expert behavior. Inverse Reinforcement Learning (IRL) tackles the problem of learning how an agent judges rewards and costs based on observed actions (39), but assumes a known dynamics model (20, 40). Conversely, Inverse Optimal Control (IOC) learns the agent's internal model for the world dynamics (41) and observations (42), but assumes the reward functions. In (43, 44) both reward function and dynamics were learned, but only the fully-observed MDP case is explored. We solve the natural but more difficult partially-observed setting, and ensure these solutions provide a scientific basis for interpreting animal behavior.

As a cognitive theory, by positing a rational but possibly mistaken agent, our approach resembles Bayesian Theory of Mind (BToM) (45–50). Previous work in BToM has considered tasks with uncertainty about static latent variables that were unknown until fully observed (50), or tasks with partially observed variables but simpler trial-based structure (45, 46). Here we allow for a more natural world, with dynamic latent variables and partial observability, and we infer models where agents make long-term plans and choose sequences of actions. Where prior work in BToM learned subjective rewards (50) *or* internal models (48), our Inverse Rational Control infers both internal models *and* subjective preferences in a partially observable world.

In addition, BToM studies have focused their attention on models of behavior, whereas our purpose is to connect dynamic model computations to brain dynamics. Some work has posited a POMDP model for behavior and hypothesized how specific brain regions might implement the relevant computations (51). Here we demonstrate an analysis framework to test such connections, by examining neural representations of latent variables and showing how computational functions could be embodied by low-dimensional neural dynamics.

While low-dimensional neural dynamics is an important topic for emerging studies of large-scale neural activity (2, 6, 52), few have been able to relate these dynamic activity patterns to interpretable latent model variables. Far more commonly, these low-dimensional manifolds are attributed to an intrinsically generated manifold (28, 53), or are related to measurable quantities like sensory inputs or behavioral outputs (2, 54, 55). Population activity in the visual system is known to relate to latent representations extracted by trained deep networks (3, 4), and while this shows that many task-relevant features extracted by machine learning solutions are also task-relevant for the visual system, these feature sets yet account for neither temporal dynamics nor uncertainty, nor are they readily interpretable (56). Our proposed model-based analysis of population activity is currently our best bet for finding interpretable computational principles.

Virtues of representation-level explanations. Many researchers in machine learning express skepticism that we can find much that is human-interpretable about either artificial or biological neural networks (57, 58). One interesting counterargument is that near any solution found by machine learning optimization, there may be other solutions that perform similarly while retaining interpretability (59). More humbly, even if we cannot find an interpretable network that exactly instantiates the brain's computations, we may still glean satisfying and useful insights from partial explanations at a higher level of abstraction (60–62).

Although the brain may not be thoroughly interpretable, we may benefit from imposing some interpretability, even at the cost of a perfect model. On the other hand, we may find instead that this imposition may lead us to more accurate neural representations that better reflect our abilities to interact with latent variables at many scales, things that brute force deep learning methods fail to find without explicit training. Finally, task-based cognitive models may reveal core principles that appear canonically across the brain.

Our recoding and decoding analysis does not apply to neural responses directly, but rather to the task-relevant information encoded in those responses. This targeted dimensionality reduction abstracts away the fine details of the neural signals in favor of an algorithmic- or representational-level description. This decreases the number of parameters needed to characterize dynamics, reducing overfitting. More importantly, it can avoid the massive degeneracies inherent in neuron-level mechanisms: different neural networks could have entirely different neural dynamics but could share the task-relevant computations. This illustrates how a deeper, more invariant understanding of neural computations is possible at the algorithmic level than at the mechanistic level.

Limitations. Future states are relevant for selecting actions, but in our formalism they are embedded implicitly in a learned policy, so an agent does not need to imagine any possible futures once learning is completed. Introspectively, our thoughts are often dedicated to anticipating what might happen the future, and neural activity shows signatures of such predictions (63). Thus a natural extension of our approach would be to examine the neural coding of these types of rational thoughts directed at future (and past) world states, both for learning and for re-evaluating policies dynamically.

We applied our method only to a fairly simple task, but our framework is quite general and can scale to much more complex tasks, and can model common errors of cogntive systems. It can be used to infer false beliefs derived from incorrect or incomplete knowledge of task parameters. It can also be used to infer incorrect *structure* within a given model class. For example, it is natural for animals to assume that some aspects of the world, such as reward rates at different locations, are not fixed, even if an experiment actually uses fixed rates (64). Similarly, an agent may have a superstition that different reward sources are correlated even when they are independent in reality. Given a model class that includes such counterfactual relationships between task variables, our method can test whether an agent holds these incorrect assumptions.

However, our approach does require a model, and it is unlikely that the brain's full internal model is easily expressible compactly. Large-scale tasks are being solved with neural networks (65, 66) that provide rich state representations, but may not permit interpretation. This may be an unavoidable limitation in a world of complex structure (57, 58). Or it may be that these uninterpretable representations are insufficiently constrained, and that richer tasks, multi-task training, and more latent variables may bias networks toward more human-interpretable representations (59, 67, 68) that relate more closely to actionable causal latent variables (24).

In experiments, uncontrolled but structured variability could arise from internal noise sources, internal states, or thoughts about other tasks. Here we have neglected these effects, but when analyzing task-relevant computation in real brains it may help to allow for structured latent dynamics that have no grounding in a task or model (69, 70) or may have an implementational purpose (13, 15, 71, 72).

Conclusion. The success of our methods on simulated agents suggests it could be fruitfully applied to experimental data from real animals performing such foraging tasks (38, 73), as well as to richer tasks requiring even more sophisticated computations. Using explainable AI to construct belief states, their dynamics, and their utility for solving interesting tasks will provide useful targets for interpreting dynamic neural activity patterns, which could help identity the neural substrates of thoughts.

Materials and Methods

Inverse Rational Control. Full mathematical details for IRC are available in Supplementary Information. Code for the discrete case is available at https://github.com/XaqLab/IRC_TwoSiteForaging.

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 $\gamma_1^* = 0.15$, $\gamma_2^* = 0.1$ and $\epsilon_1^* = 0.05$, $\epsilon_2^* = 0.04$.

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 $\gamma_1 = 0.2$, $\gamma_2 = 0.15$, $\epsilon_1 = 0.1$, $\epsilon_2 = 0.08$, $q_1 = 0.42$, and $q_2 = 0.66$.

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 0.2 when traveling. Switching between boxes requires two steps, for a total cost of 0.4. 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 a rational agent. The target behavior was implemented by an agent that used optimal belief updates and a softmax policy trained to solve a Belief MDP by value iteration (11).

Our neural network used one recurrently connected layer of 300 rectified linear units (ReLUs) that received external inputs from the world-generated observations and agent-generated actions. Beliefs were estimated from this recurrent layer by a linear weighted sum. In parallel, the recurrently connected neurons provided input to a two-layer perceptron, with 100 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 mean-squared error loss function and KL-divergence loss function, in two phases respectively. First, the recurrent connection strengths and the linear belief readout were jointly optimized by backpropagation through time to match the dynamic beliefs of the target agent. Second, the linear belief readout was discarded, and the recurrent units' outputs were passed through the two nonlinear stages and were optimized so that the 5 policy neurons matched the target POMDP policy at all times. After 60 iterations of 20 batches of 500 time points per batch, the trained neural network successfully reproduced the target beliefs within a mean squared error of 0.003, and the target policy within an average KL divergence of 0.005.

The trained neural network could then be run autonomously in closed-loop mode, sampling its own actions from a softmax distribution applied to the 5 output neurons.

Neural coding analysis. *Encoding*: We find an encoding matrix $\check{W}$ by regressing b against $\boldsymbol{r}$. This produces neural estimates of task-relevant variables $\check{b} = \check{W}\boldsymbol{r} + \boldsymbol{c}$ for new data. *Recoding*: We find dynamics by regressing $\check{b}_t$ against $(\check{b}_{t-1}, o_t)$ with kernel ridge regression. The kernel functions are radial basis functions with centers on all possible target beliefs and a width at half-max equal to the spacing between beliefs. This yields the 'recoding' function $\check{f}_{\mathrm{rec}}(\check{b}_t, o_t)$ representing the nonlinear dynamics of the neural beliefs. We compare the belief updates $\Delta\check{b}_t = \check{f}(\check{b}_t, o_t) - \check{b}_t$ from the recoding function $\check{f}_{\mathrm{rec}}(\check{b}_t, o_t)$ and the corresponding belief updates from the task dynamics $\Delta\hat{b}_t = \hat{f}_{\mathrm{dyn}}(\hat{b}_t, o_t) - \hat{b}_t$. *Decoding*: We compute the brain's 'decoding' function, *i.e.* an approximate policy $\check{\pi}_{\mathrm{dec}}$, using nonlinear multinomial regression of $\check{b}$ against a with the same radial basis functions as used in recoding. We use a feature space of radial basis functions with centers on a 9×9 grid over beliefs, with width equal to the center spacing, and an outer product space over locations.

Acknowledgments

The authors thank Baptiste Caziot, Dora Angelaki, Neda Shahidi, Valentin Dragoi, Rajkumar Raju, and Zhe Li for useful discussions. ZW, PS, and XP were supported in part by BRAIN Initiative grant NIH 5U01NS094368. ZW and XP were supported in part by an award from the McNair Foundation. SD and XP were supported in part by the Simons Collaboration on the Global Brain award 324143 and NSF 1450923 BRAIN 43092. XP and MK were supported in part by NSF CAREER Award IOS-1552868.

1 BF Skinner, *About behaviorism.* (Vintage), (2011).

2 V Mante, D Sussillo, KV Shenoy, WT Newsome, Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013) Mante, Valerio Sussillo, David Shenoy, Krishna V Newsome, William T eng 1DP1OD006409/OD/NIH HHS/ Howard Hughes Medical Institute/ England Nature. 2013 Nov 7;503(7474):78-84. doi: 10.1038/nature12742.

3 N Kriegeskorte, Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. review vision science* **1**, 417–446 (2015).

4 DL Yamins, et al., Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci.* **111**, 8619–8624 (2014).

5 Y Gao, EW Archer, L Paninski, JP Cunningham, Linear dynamical neural population models through nonlinear embeddings in *Advances in neural information processing systems.* pp. 163–171 (2016).

6 R Chaudhuri, B Gercek, B Pandey, A Peyrache, I Fiete, The population dynamics of a canonical cognitive circuit. *bioRxiv*, 516021 (2019).

7 Plato, A Bloom, A Kirsch, *The Republic.* (Basic Books), (2016).

8 P Gao, S Ganguli, On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr. opinion neurobiology* **32**, 148–155 (2015).

9 RS Sutton, AG Barto, *Reinforcement learning: An introduction.* (MIT press), (2018).

10 LP Kaelbling, ML Littman, AW Moore, Reinforcement learning: A survey. *J. artificial intelligence research* **4**, 237–285 (1996).

11 R Bellman, *Dynamic programming.* (Princeton University Press), (1957).

12 TS Lee, D Mumford, Hierarchical bayesian inference in the visual cortex. *JOSA A* **20**, 1434–1448 (2003).

13 P Berkes, G Orban, M Lengyel, J Fiser, Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science* **331**, 83–7 (2011).

14 WJ Ma, JM Beck, PE Latham, A Pouget, Bayesian inference with probabilistic population codes. *Nat. neuroscience* **9**, 1432 (2006).

15 C Savin, S Deneve, Spatio-temporal representations of uncertainty in spiking neural networks in *Advances in Neural Information Processing Systems.* pp. 2024–2032 (2014).

16 RV Raju, X Pitkow, Marginalization in random nonlinear neural networks in *APS March Meeting Abstracts.* Vol. 1, p. 1107P (2015).

17 E Vértes, M Sahani, Flexible and accurate inference and learning for deep generative models in *Advances in Neural Information Processing Systems.* pp. 4166–4175 (2018).

18 RA Howard, *Dynamic programming and Markov processes.* (Wiley for The Massachusetts Institute of Technology), (1964).

19 AP Dempster, NM Laird, DB Rubin, Maximum likelihood from incomplete data via the EM algorithm. *J. royal statistical society. Ser. B (methodological)*, 1–38 (1977).

20 M Babes, V Marivate, K Subramanian, ML Littman, Apprenticeship learning about multiple intentions in *Proceedings of the 28th International Conference on Machine Learning (ICML-11).* pp. 897–904 (2011).

21 TP Minka, Expectation propagation for approximate bayesian inference in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence.* (Morgan Kaufmann Publishers Inc.), pp. 362–369 (2001).

22 S Daptardar, S Paul, X Pitkow, Inverse rational control with partially observable nonlinear dynamics in *ArXiv 1908.04696.* (2019).

23 TP Lillicrap, et al., Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

24 JJ Gibson, The theory of affordances. *Hilldale, USA* **1** (1977).

25 MN Shadlen, R Kiani, TD Hanks, AK Churchland, An intentional framework. *Better than conscious*, 71–101 (2008).

26 R Brette, Is coding a relevant metaphor for the brain? *Behav. Brain Sci.*, 1–44 (2019).

27 D Marr, *Vision: A computational investigation into the human representation and processing of visual information.* (MIT Press), (1982).

28 R Chaudhuri, B Gerçek, B Pandey, A Peyrache, I Fiete, The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. Neurosci.* (2019).

29 PT Sadtler, et al., Neural constraints on learning. *Nature* **512**, 423 (2014).

30 JD Semedo, A Zandvakili, CK Machens, MY Byron, A Kohn, Cortical areas interact through a communication subspace. *Neuron* **102**, 249–259 (2019).

31 H von Helmholtz, JPC Southall, *Treatise on physiological optics.* (Courier Corporation) Vol. 3, (2005).

32 D Kahneman, A Tversky, Prospect theory: An analysis of decision under risk in *Handbook of the fundamentals of financial decision making: Part I.* (World Scientific), pp. 99–127 (2013).

33 MO Ernst, MS Banks, Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* **415**, 429–433 (2002).

34 SW Wu, MR Delgado, LT Maloney, Economic decision-making compared with an equivalent motor task. *Proc. Natl. Acad. Sci.* **106**, 6088–6093 (2009).

35 AL Fairhall, GD Lewen, W Bialek, RRdR van Steveninck, Efficiency and ambiguity in an adaptive neural code. *Nature* **412**, 787 (2001).

36 Q Yang, XS Pitkow, Revealing nonlinear neural decoding by analyzing choices. *BioRxiv*, 332353 (2018).

37 A Borst, VL Flanagin, H Sompolinsky, Adaptation without parameter change: dynamic gain control in motion detection. *Proc. Natl. Acad. Sci.* **102**, 6172–6176 (2005).

38 LP Sugrue, GS Corrado, WT Newsome, Matching behavior and the representation of value in the parietal cortex. *Science* **304**, 1782–1787 (2004).

39 S Russell, Learning agents for uncertain environments in *Proceedings of the eleventh annual conference on Computational learning theory.* (ACM), pp. 101–103 (1998).

40 J Choi, KE Kim, Inverse reinforcement learning in partially observable environments. *J. Mach. Learn. Res.* **12**, 691–730 (2011).

41 K Dvijotham, E Todorov, Inverse optimal control with linearly-solvable mdps in *Proceedings of the 27th International Conference on Machine Learning (ICML-10).* pp. 335–342 (2010).

42 F Schmitt, HJ Bieg, M Herman, CA Rothkopf, I see what you see: Inferring sensor and policy models of human real-world motor behavior. in *AAAI.* pp. 3797–3803 (2017).

43 M Herman, T Gindele, J Wagner, F Schmitt, W Burgard, Inverse reinforcement learning with simultaneous estimation of rewards and dynamics in *Artificial Intelligence and Statistics.* pp. 102–110 (2016).

44 S Reddy, AD Dragan, S Levine, Where do you think you're going? inferring beliefs about dynamics from behavior in *Arxiv 1805.08010.* (2018).

45 J Daunizeau, et al., Observing the observer (I): Meta-Bayesian models of learning and decision-making. *PLoS One* **5**, e15554 (2010).

46 F Huszár, U Noppeney, M Lengyel, Mind reading by machine learning: A doubly Bayesian method for inferring mental representations in *Proceedings of the Annual Meeting of the Cognitive Science Society.* Vol. 32, (2010).

47 C Baker, R Saxe, J Tenenbaum, Bayesian theory of mind: Modeling joint belief-desire attribution in *Proceedings of the annual meeting of the cognitive science society.* Vol. 33, (2011).

48 AN Rafferty, MM LaMar, TL Griffiths, Inferring learners' knowledge from their actions. *Cogn. Sci.* **39**, 584–618 (2015).

49 K Khalvati, RP Rao, A bayesian framework for modeling confidence in perceptual decision making in *Advances in neural information processing systems.* pp. 2413–2421 (2015).

50 CL Baker, J Jara-Ettinger, R Saxe, JB Tenenbaum, Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nat. Hum. Behav.* **1**, 0064 (2017).

51 RP Rao, Decision making under uncertainty: a neural model based on partially observable markov decision processes. *Front. computational neuroscience* **4**, 146 (2010).

52 C Stringer, M Pachitariu, N Steinmetz, M Carandini, KD Harris, High-dimensional geometry of population responses in visual cortex. *Nature*, 1 (2019).

53 M Tsodyks, T Kenet, A Grinvald, A Arieli, Linking spontaneous activity of single cortical neurons and the underlying functional architecture. *Science* **286**, 1943–1946 (1999).

54 MM Churchland, et al., Neural population dynamics during reaching. *Nature* **487**, 51 (2012).

55 S Musall, MT Kaufman, AL Juavinett, S Gluf, AK Churchland, Single-trial neural dynamics are dominated by richly varied movements. *bioRxiv*, 308288 (2019).

56 MD Zeiler, R Fergus, Visualizing and understanding convolutional networks in *European conference on computer vision.* (Springer), pp. 818–833 (2014).

57 R Sutton, The bitter lesson (2019).

58 TP Lillicrap, KP Kording, What does it mean to understand a neural network? *arXiv preprint arXiv:1907.06374* (2019).

59 C Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**, 206 (2019).

60 AE Orhan, WJ Ma, Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback. *Nat. communications* **8**, 138 (2017).

61 R Geirhos, et al., Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231* (2018).

62 J Frankle, M Carbin, The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635* (2018).

63 K Diba, G Buzsáki, Forward and reverse hippocampal place-cell sequences during ripples. *Nat. neuroscience* **10**, 1241 (2007).

64 CM Glaze, AL Filipowicz, JW Kable, V Balasubramanian, JI Gold, A bias–variance trade-off governs individual differences in on-line learning in an unpredictable environment. *Nat. Hum. Behav.* **2**, 213 (2018).

65 V Mnih, et al., Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*

(2013).

66 D Silver, et al., Mastering the game of go with deep neural networks and tree search. *nature* **529**, 484 (2016).

67 L Gatys, AS Ecker, M Bethge, Texture synthesis using convolutional neural networks in *Advances in neural information processing systems.* pp. 262–270 (2015).

68 F Sinz, X Pitkow, J Reimer, M Bethge, A Tolias, Engineering a less artificial ingelligence, Technical report (2019).

69 J Zylberberg, Untuned but not irrelevant: The role of untuned neurons in sensory information coding. *BioRxiv*, 134379 (2017).

70 MR Whiteway, DA Butts, The quest for interpretable models of neural population activity. *Curr. Opin. Neurobiol.* **58**, 86–93 (2019).

71 A Longtin, Stochastic resonance in neuron models. *J. statistical physics* **70**, 309–327 (1993).

72 RM Haefner, P Berkes, J Fiser, Perceptual decision-making as probabilistic inference by neural sampling. *Neuron* **90**, 649–660 (2016).

73 O Odoemene, S Pisupati, H Nguyen, AK Churchland, Visual evidence accumulation guides decision-making in unrestrained mice. *J. Neurosci.* **38**, 10143–10155 (2018).

74 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).

75 KP Murphy, *Machine learning: a probabilistic perspective.* (MIT press), (2012).

symbol	meaning	symbol	meaning
t	time		
s	world state	$T(s'\|s,a)$, $\overline{T}(b'\|b,a)$	transition probability
o	observation	$O(o\|s)$, $\overline{O}(o\|b)$	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, L	loss
$\check{\varphi}_{\text{enc}}$	estimate from encoding: $\boldsymbol{r} \to \check{b}$		
$\check{f}_{\text{rec}}$	recoding / neural dynamics: $\check{b} \to \check{b}$	$\hat{f}_{\text{dyn}}$	behavioral dynamics: $\hat{b} \to \hat{b}$
$\check{\pi}_{\text{dec}}$	decoding / neural policy: $\check{b} \to a$	$\hat{\pi}_{\text{act}}$	behavioral policy: $\hat{b} \to a$

Table S1. Glossary of notation.

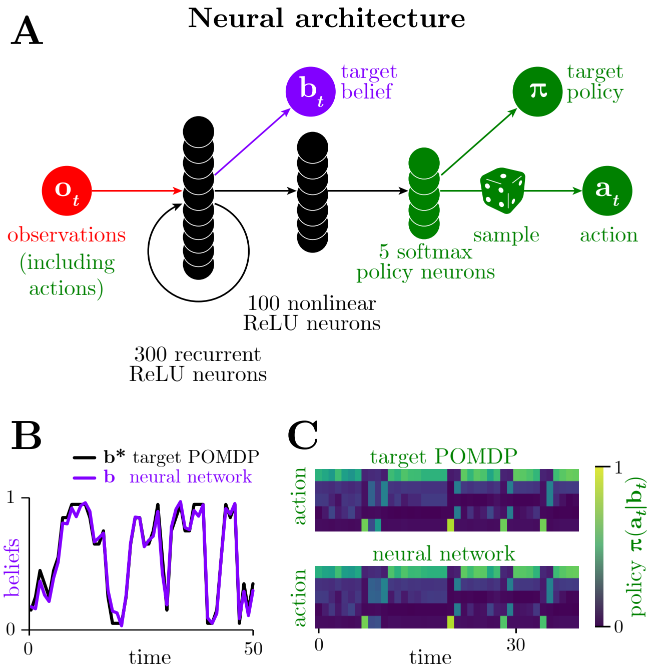

Fig. S1. A: Architecture of a synthetic brain trained to behave rationally by matching the true beliefs b and policy π of a POMDP agent. The recurrent network uses 300 fully-connected neurons with a ReLU nonlinearity. There are 5 policy neurons, one for each possible action, and the network samples an action from the softmax over these policy neurons. Notice that there are no hats over these quantities, because these are not estimates. **B**: The neural network has almost the same beliefs as a rational agent given the same observations. **C**: Neural network reproduces the policy of a rational agent.

Supporting Information Appendix (SI).

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

$$B(s_t|b_t) = B(s_t|o_{1:t}, a_{1:t-1}) \quad [4]$$

$$= B(s_t|o_t, a_{t-1}, b_{t-1}) \quad [5]$$

$$= \frac{1}{Z} O(o_t|s_t) \int ds_{t-1}\, T(s_t|s_{t-1}, a_{t-1}) B(s_{t-1}|o_{t-1}, a_{t-2}, b_{t-2}) \quad [6]$$

$$= \frac{1}{Z} O(o_t|s_t) \int ds_{t-1}\, T(s_t|s_{t-1}, a_{t-1}) B(s_{t-1}|b_{t-1}) \quad [7]$$

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

$$\overline{T}(b_{t+1}|b_t,a_t)=\int do_{t+1}\,P(b_{t+1}|b_t,a_t,o_{t+1})\overline{O}(o_{t+1}|b_t) \qquad [8]$$

where

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

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

$$\overline{R}(b_t,a_t)=\int ds_t\,R(s_t,a_t)B(s_t|b_t) \qquad [9]$$

These beliefs, belief transitions $\overline{T}$, and rewards $\overline{R}$ then determine the optimal policy through the Bellman equation, as described in the main text [2].

Markov structure in Inverse Rational Control. The log-likelihood of the observed data $\mathcal{L}(\theta)$ [3] 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, $\mathcal{L}(\theta)=\mathcal{Q}(\theta)+H$, as in the Expectation-Maximization (EM) algorithm (19).[†] 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

$$\begin{aligned}\mathcal{Q}(\theta)=\Big\langle\quad &\log p(b_1,o_1,s_1|\theta,\phi) && [10]\\ +\sum\nolimits_t &\log \pi(a_t|b_t,\theta) && [11]\\ +\sum\nolimits_t &\log p(b_{t+1}|b_t,a_t,o_t,\theta) && [12]\\ +\sum\nolimits_t &\log O(o_{t+1}|s_{t+1},\phi) && [13]\\ +\sum\nolimits_t &\log T(s_{t+1}|s_t,a_t,\phi)\Big\rangle_{p(b_{1:T}|a_{1:T},o_{1:T},s_{1:T},\theta,\phi)} && [14]\end{aligned}$$

The term in [12] depends only on the parameters for the state dynamics and observations, while the policy term in [11] depends on both the dynamics and observation parameters and reward functions. The entropy H of the posterior can be computed similarly (see below).

Note that the true world state s only appears in terms with the experimental parameters ϕ, and does 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.

According to (74), the entropy of the posterior over beliefs can be calculated recursively as

$$\begin{aligned}H(b_{1:t-1}|b_t,o_{1:t},a_{1:t},s_{1:t},\theta,\phi)=&\int db_t\,H(b_{1:t-2}|b_{t-1},o_{1:t-1},a_{1:t-1},s_{1:t-1},\theta,\phi)p(b_{t-1}|b_t,o_{1:t},a_{1:t},s_{1:t},\theta,\phi) && [15]\\ &+H(p(b_{t-1}|b_t,o_{1:t},a_{1:t},s_{1:t},\theta,\phi)) && [16]\end{aligned}$$

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

$$\begin{aligned}H(p(b_{1:T}|a_{1:T},o_{1:T},s_{1:T},\theta,\phi)=&\int db_T\,H(b_{1:T-1}|b_T,o_{1:T},a_{1:T},s_{1:T},\theta,\phi)p(b_T|a_{1:T},o_{1:T},s_{1:T},\theta,\phi) && [17]\\ &+H(p(b_T|a_{1:T},o_{1:T},s_{1:T},\theta,\phi)) && [18]\end{aligned}$$

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.

[†] 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.

EM algorithm. The EM algorithm (19) 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

$$\mathcal{L}(\theta) = \log \int db_{1:T}\, p(b_{1:T}, o_{1:T}, a_{1:T}, s_{1:T}|\theta, \phi) \tag{19}$$

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.

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

$$\overline{\mathcal{L}}(\theta) = \mathcal{Q}(\theta, \theta^{\text{old}}) + H(P(b_{1:T}|a_{1:T}, o_{1:T}, \theta^{\text{old}})) \tag{20}$$

To be consistent with (75), we use $\mathcal{Q}(\theta, \theta^{\text{old}})$ as the auxiliary function that describes the expected complete data log likelihood, and $H(\cdot)$ 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:

$$\mathcal{Q}(\theta, \theta^{\text{old}}) = \langle \log p(b_{1:T}, a_{1:T}, o_{1:T}, s_{1:T}|\theta, \phi)\rangle_{P(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi^{\text{old}})} \tag{21}$$

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 [10–14], except now the posterior distribution over beliefs is based on the previous iteration's parameters:

$$\begin{aligned}
\mathcal{Q}(\theta, \theta^{\text{old}}) = \Big\langle \quad & \log P(b_1, o_1, s_1|\theta, \phi) && [22]\\
+ \sum\nolimits_t & \log \pi(a_t|b_t, \theta) && [23]\\
+ \sum\nolimits_t & \log p(b_{t+1}|b_t, a_{t+1}, o_t, \theta) && [24]\\
+ \sum\nolimits_t & \log O(o_{t+1}|s_t, \phi) && [25]\\
+ \sum\nolimits_t & \log T(s_{t+1}|s_t, a_t, \phi) \Big\rangle_{P(b_{1:T}|a_{1:T}, o_{1:T}, s_{1:T}, \theta^{\text{old}}, \phi^{\text{old}})} && [26]
\end{aligned}$$

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 $H(P(b_{1:T}|a_{1:T}, o_{1:T}, \theta^{\text{old}})$ 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 [22] reflects the initial belief distribution, and it has a negligible contribution to $\mathcal{Q}$ when there are many time points t. In [24], the transition probability $P(b_{t+1}|b_t, a_{t+1}, o_t, \theta)$ is a function of the dynamics parameters, while in [23], the policy term $P(a_t|b_t, \theta)$ 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, $\pi(a|b) \sim \frac{1}{Z(b)} e^{Q(b,a)/\tau}$, 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:

$$\frac{\partial \pi(a|b)}{\partial \theta_i} = \frac{\partial \pi(a|b)}{\partial Q(b,a)} \frac{\partial Q(b,a)}{\partial \theta_i} + \int_{a' \neq a} da' \frac{\partial \pi(a|b)}{\partial Q(b,a')} \frac{\partial Q(b,a')}{\partial \theta_i}. \tag{27}$$

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

$$\frac{\partial Q(b_t,a_t)}{\partial\theta_i} = \frac{\partial\overline{R}(b_t,a_t)}{\partial\theta_i} \tag{28}$$

$$+\gamma\int db_{t+1}\frac{\overline{T}(b_{t+1}|b_t,a_t)}{\partial\theta_i}\int da_{t+1}\pi(a_{t+1}|b_{t+1})Q(b_{t+1},a_{t+1}) \tag{29}$$

$$+\gamma\int db_{t+1}\overline{T}(b_{t+1}|b_t,a_t)\int da_{t+1}\frac{\partial\pi(a_{t+1}|b_{t+1})}{\partial\theta_i}Q(b_{t+1},a_{t+1}) \tag{30}$$

$$+\gamma\int db_{t+1}\overline{T}(b_{t+1}|b_t,a_t)\int da_{t+1}\pi(a_{t+1}|b_{t+1})\frac{\partial Q(b_{t+1},a_{t+1})}{\partial\theta_i} \tag{31}$$

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 [28–29]:

$$c_i(b_t,a_t) = \frac{\partial\overline{R}(b_t,a_t)}{\partial\theta_i} + \gamma\int db_{t+1}\frac{\overline{T}(b_{t+1}|b_t,a_t)}{\partial\theta_i}\int da_{t+1}\pi(a_{t+1}|b_{t+1})Q(b_{t+1},a_{t+1}) \tag{32}$$

With this substitution we have

$$\frac{\partial Q(b_t,a_t)}{\partial\theta_i} = c_i(b_t,a_t)+\gamma\int db_{t+1}\overline{T}(b_{t+1}|b_t,a_t)\int da_{t+1}\left[\frac{\partial\pi(a_{t+1}|b_{t+1})}{\partial\theta_i}Q(b_{t+1},a_{t+1}) + \pi(a_{t+1}|b_{t+1})\frac{\partial Q(b_{t+1},a_{t+1})}{\partial\theta_i}\right] \tag{33}$$

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 [27].

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 [33] can then be rewritten as a linear function

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

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

$$\begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i}\\ \vdots\end{bmatrix} = \left(\boldsymbol{I}-\gamma\boldsymbol{\Gamma}(\overline{T}(b_{t+1}|b_t,a_t))\left(\begin{bmatrix}\ddots & & \\ & \boldsymbol{Q}_t^V & \\ & & \ddots\end{bmatrix}\begin{bmatrix}\vdots\\ \frac{\partial\boldsymbol{\pi}_t^V}{\partial\boldsymbol{Q}_t^V}\\ \vdots\end{bmatrix} + \begin{bmatrix}\ddots & & \\ & \boldsymbol{\pi}_t^V & \\ & & \ddots\end{bmatrix}\right)\right)^{-1}\begin{bmatrix}\vdots\\ \boldsymbol{c}_{i,t}^V\\ \vdots\end{bmatrix} \tag{34}$$

Without the brackets indicating the matrix shapes, finally we obtain

$$\frac{\partial\boldsymbol{Q}_t^V}{\partial\theta_i} = \left(\boldsymbol{I}-\gamma\boldsymbol{\Gamma}\left(\mathrm{Diag}(\boldsymbol{Q}_t^V)\frac{\partial\boldsymbol{\pi}_t^V}{\partial\boldsymbol{Q}_t^V}+\mathrm{Diag}(\boldsymbol{\pi}_t^V)\right)\right)^{-1}\boldsymbol{c}_{i,t}^V. \tag{35}$$

With the chain rule [27], we can obtain the gradients of the policy with respect to the parameters θ, which lets us calculate the gradient of the $\mathcal{Q}(\theta,\theta^{\text{old}})$ function in [22–26], 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.