arXiv:2009.12576v1 [cs.LG] 26 Sep 2020

Inverse Rational Control with Partially Observable Continuous Nonlinear Dynamics

Minhae Kwon
School of Electronic Engineering
Soongsil University
Seoul, Republic of Korea
minhae@ssu.ac.kr

Saurabh Daptardar
Google Inc.
Mountain View, CA, USA
saurabh.dptdr@gmail.com

Paul Schrater
Department of Computer Science
University of Minnesota
Minnesota, IN, USA
schrater@umn.edu

Xaq Pitkow
Electrical and Computer Engineering
Rice University
Houston, TX, USA
xaq@rice.edu

Abstract

A fundamental question in neuroscience is how the brain creates an internal model of the world to guide actions using sequences of ambiguous sensory information. This is naturally formulated as a reinforcement learning problem under partial observations, where an agent must estimate relevant latent variables in the world from its evidence, anticipate possible future states, and choose actions that optimize total expected reward. This problem can be solved by control theory, which allows us to find the optimal actions for a given system dynamics and objective function. However, animals often appear to behave suboptimally. Why? We hypothesize that animals have their own flawed internal model of the world, and choose actions with the highest expected subjective reward according to that flawed model. We describe this behavior as *rational* but not optimal. The problem of Inverse Rational Control (IRC) aims to identify which internal model would best explain an agent's actions. Our contribution here generalizes past work on Inverse Rational Control which solved this problem for discrete control in partially observable Markov decision processes. Here we accommodate continuous nonlinear dynamics and continuous actions, and impute sensory observations corrupted by unknown noise that is private to the animal. We first build an optimal Bayesian agent that learns an optimal policy generalized over the entire model space of dynamics and subjective rewards using deep reinforcement learning. Crucially, this allows us to compute a likelihood over models for experimentally observable action trajectories acquired from a suboptimal agent. We then find the model parameters that maximize the likelihood using gradient ascent. Our method successfully recovers the true model of rational agents. This approach provides a foundation for interpreting the behavioral and neural dynamics of animal brains during complex tasks.

1 Introduction

Brains evolved to understand, interpret, and act upon the physical world. To thrive and reproduce in a harsh and dynamic natural environment, brains, therefore, evolved flexible, robust controllers. To be the controller, the fundamental function of the brain is to organize sensory data into an internal model of the outside world. The animals are never able to get complete information about the world. Instead,

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

they only get partial and noisy observations of it. Thus, the brain should build its own internal model which necessarily includes uncertainties of the outside world, and base its decision upon that model [1]. However, we hypothesize that this internal model is not always correct, but the animals still behave rationally — meaning that animals act optimally according to their own internal model of the world, which may differ from the true world.

The goal of this paper is to identify the internal model of the agent by observing its actions. Unlike Inverse Reinforcement Learning (IRL) [2, 3, 4] which aims to learn only the reward function of target agent, or Inverse Optimal Control (IOC) [5, 6] to infer only unknown dynamics model, we use Inverse Rational Control (IRC) [7] to infer both. Since we consider neuroscience tasks which include naturalistic controls and complex physics of the world, we substantially extend past work [7] to include continuous spaces of state, action, and parameter with nonlinear dynamics. We parameterize nonlinear task dynamics and reward functions based on a physics model such that the family of tasks shares an overall structure but has different model parameters. In our framework, an *experimentalist* can observe state information of the environment and actions taken by the agent. On the other hand, the experimentalist cannot observe information about the agent's internal model, such as its observations and beliefs. IRC infers the latent internal information of the agent using the data observable by the experimentalist.

The task is formulated as a Partially Observable Markov Decision Process (POMDP) [8, 9], a powerful framework for modeling agent behavior under uncertainty. In order to model an animal's cognitive process whereby the decision-making is based on its own beliefs about the world, we reformulate the POMDP as a belief Markov Decision Process (belief MDP) [10, 11]. The agent builds its belief (*i.e.*, its posterior distribution over world states) based on partial, noisy observations and its internal model, and the decision-making is based on its belief.

We construct a Bayesian agent to learn optimal policies and value functions over an entire parameterized family of models, which can be viewed as an optimized ensemble of agents each dedicated to one task. This then allows us to maximize the likelihood of the state-action trajectories generated by a target agent, by finding which parameters from the ensemble best explain the target agent's data.

The main contributions of this paper are the following. First, our work is able to find both the reward function and internal dynamics model simultaneously in continuous nonlinear tasks. Note that continuous nonlinear dynamical systems are the most general form of tasks, so it is trivial to solve discrete and/or linear systems using the proposed approach. Second, we propose a novel approach to implement the Bayesian optimal control ensembles, including an idea of belief representation and belief updating method using estimators with constrained representational capacity (*e.g.*, an extended Kalman filter). This allows us to build an algorithm that imitates the bounded rational cognitive process of the brain [12] and to perform belief-based decision-making. Lastly, we propose a novel approach to IRC combining Maximum Likelihood Estimation (MLE) and Monte Carlo Expectation-Maximization (MCEM). This method successfully infers the reward function and internal model parameters of the target agent by maximizing the likelihood of state-action trajectories under the assumption of rationality, while marginalizing over latent sensory observations. Importantly, this is possible because we trained ensembles of agents over entire parameter spaces using flexible function approximators. To the best of our knowledge, our work is the first study to infer both the reward and internal model of an unknown agent with partially observable continuous nonlinear dynamics.

2 Related Work

Inverse reinforcement learning (IRL). The goal of IRL or imitation learning is to learn a reward function or a policy from expert demonstrations, and the goal of Inverse Optimal Control (IOC) is to infer an unknown dynamics model. Both approaches solve aspects of the general problem of inferring internal models of an observed agent. For example, some IRL works such as [13, 14, 15, 16] formulate the optimization problems to find features of reward or cost function that best explain the target agent's state-action trajectories. Specifically, [13] finds reward features by solving a linear programming problem, and [15] uses a quadratic programming method to learn mappings from features to a cost function. In addition, [17] combines the principle of maximum entropy [18] to IRL so that the solution becomes as random as possible while still matching reward features the best. This guarantees avoiding the worst-case policy [19, 20]. Another stream of IRL is imitation learning [21, 22, 23, 24]. Typical IRL approaches use a two-step process: first learn the expert's

reward function first, and then train the policy with the learned reward. This could be slow, [21] directly extracts a policy from data. Across all of these methods, there is no a complete inverse solution that can learn how an agent models rewards, dynamics, and uncertainty in a partially observable task with continuous nonlinear dynamics and continuous controls.

Meta reinforcement learning (Meta RL). The fundamental objective of Meta RL is to learn new skills or adapt to a new environment rapidly with a few training experiences. In order to efficiently adapt to the new tasks or environments, some Meta RL works try to infer tasks or meta parameters that govern the general task. For example, optimization-based meta RL works such as [25, 26, 27, 28] include a so-called 'outer loop' which optimizes the meta-parameters. In this sense, the meta RL is related to our goal since both work aim to infer the task parameters, although we use this parameterization to explain the actions of an agent. However, there are few studies to consider the partially observable setting of the agent. [29] includes both POMDP frameworks and meta-learning, but the partially observable information is the task information not the state information. [30] also considers a Bayesian approach with meta-learning, but it also uses Bayesian reasoning to infer the unseen tasks and learn quickly. Therefore, our paper differs from other Meta RL works in its task structure and goal. We allow partial observability about the world state, as occurs naturally in the animal's decision-making process. More fundamentally, the goal of our work is not to find smarter agents, but rather to infer the internal model of an existing agent and to explain its behaviors.

Neuroscience and cognitive science Neuroscientists aim to answer how the brain selects actions based on noisy sensory information and incomplete knowledge of the world. The hypothesis of the Bayesian brain [31] has been proposed to explain the brain's functionalities with Bayesian inference and probabilistic representations of the animal's internal state. Several studies propose mechanisms by which neurons could implement optimal behaviors despite pervasive uncertainty [11, 32, 33]. Despite the utility of having behavioral benchmarks based on optimality, animals often appear to behave suboptimally. Such suboptimality might come from the wrong internal model [7, 34] that is induced by a subjective prior belief of the animal [35, 36] and suboptimal inference [37]. The main goal of this paper is to infer the internal model of suboptimal agents using state-of-the-art deep reinforcement learning techniques and to provide a theoretical tool to interpret behavior and neural data obtained from ongoing neuroscience research.

For this reason, we test our approach by simulating an existing neuroscience task called 'catching fireflies' [38, 39], which is complex enough to require a sophisticated internal model, while being restrictive enough that animals can learn it and one can adequately constrain models of this behavior using feasible data volumes. Ultimately we will apply our approach to understand the internal models of behaving animals, where we do not know the ground truth. Before doing that, it is important to use simulated agents that allow us to validate the method when we do know the ground truth. Recently, a similar effort to build AI-relevant testbed for animal cognition and behavior is presented in [40, 41].

3 Bayesian Optimal Control Ensembles

Our method can be viewed as a search over an ensemble of agents, each optimally trained on different POMDP tasks, to find which of these agents best explain the experimentally observed behaviors of a target agent. The experimentalist is an external observer who has information about the world states and agent actions, but not about the agent's internal model, noisy sensory observations, or beliefs.

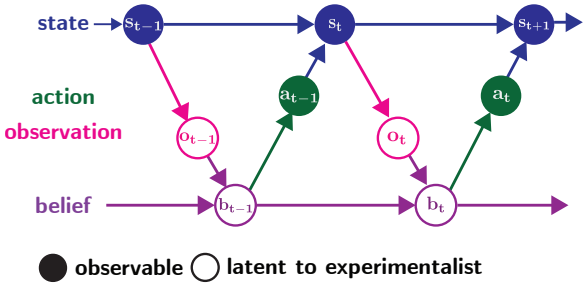

Figure 1: Graphical model of a POMDP. Solid circles denote observable variables to an experimentalist, and empty circles denote latent variables.

3.1 Belief Markov Decision Process and optimal control

A POMDP is defined as a tuple $M = (\mathcal{S}, \mathcal{A}, \Omega, R, T, O, \gamma)$ that includes states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, observations $o_t \in \Omega$, reward functions $R(s_t, a_t, s_{t+1}; \theta)$, state transition probabilities $T(s_{t+1}|s_t, a_t; \theta)$, observation probabilities $O(o_t|s_t; \theta)$ at time t, and a temporal discount factor γ. Here, $\theta \in \Theta$ denotes a vector of model parameters defining the rewards, state transitions and

observations, and the state space $\mathcal{S}$ and action space $\mathcal{A}$ are considered to occupy a continuous space. Thus, θ parameterizes a POMDP family. A graphical model of a POMDP is presented in Figure 1.

The state s_t is defined as the representation of the environment which can live in high dimensional spaces. It may be fully accessible by the experimentalist but not by the agent. The agent gets an observation o_t of the environment with state s_t, which is partial and noisy version of state s_t. Because of the partial observability, the dimension of o_t could be lower than the dimension of s_t. The observation process is modeled by the observation function $O(o_t|s_t;\theta)$. Note that any noise added from state to observation is the internal noise of the agent, *i.e.*, the noise within the nervous system of the agent. Because of this noise, the observation is only known to the agent and the experimentalist can never access it directly.

Based on its observations and actions up to time t, a rational agent builds a posterior distribution $B(s_t|o_{1:t}, a_{1:t-1};\theta)$ over the world state given the history of observations and actions, and it bases its actions upon that posterior. In practice this posterior is summarized by a *belief* b_t, defined as sufficient statistics for the posterior distribution over states, *i.e.*, $B(s_t|b_t) = B(s_t|o_{1:t}, a_{1:t-1};\theta)$. In principle, a belief b_t over a general continuous state could be infinite-dimensional, but we assume that the belief is continuous but finite-dimensional. Let $B(s_t|b_t)$ be the probability that the environment is in the state s_t when the agent's belief is b_t. By the Markov property, b_t is determined by b_{t-1}, a_{t-1}, o_t such that $B(s_t|b_t)$ can be calculated as follows.

$$B(s_t|b_t) = B(s_t|b_{t-1}, a_{t-1}, o_t;\theta) \tag{1}$$

$$= \frac{1}{Z} O(o_t|s_t;\theta) \int ds_{t-1} T(s_t|s_{t-1}, a_{t-1};\theta) B(s_{t-1}|b_{t-1}) \tag{2}$$

where $Z = \int ds_t\, O(o_t|s_t;\theta) \int ds_{t-1} T(s_t|s_{t-1}, a_{t-1};\theta) B(s_{t-1}|b_{t-1})$ is a normalizing constraint. In general this recursion is intractable, so we approximate it under tractable model assumptions, as we do in our application below. By replacing the state of the environment by the belief of the agent, the POMDP problem can be reformulated as a belief MDP problem, and the optimal policy can be found based on well-known MDP solvers [42, 43, 44, 45] applied to the fully observed belief state.

The optimal policy $\pi^*(a_t|b_t;\theta)$ defines how the agent chooses an action a_t^* that maximizes the temporally discounted total expected future reward, given the current belief b_t and internal model θ. This defines the Q-value $Q(b_t, a_t;\theta)$ as a belief-action value:

$$Q(b_t, a_t;\theta) = \int db_{t+1} \overline{T}(b_{t+1}|b_t, a_t;\theta) \left(\overline{R}(b_t, a_t, b_{t+1};\theta) + \gamma \max_a Q(b_{t+1}, a;\theta) \right) \tag{3}$$

where $\overline{T}(b_{t+1}|b_t, a_t;\theta)$ is the belief transition probability and $\overline{R}(b_t, a_t, b_{t+1};\theta)$ is the reward as a function of belief, defined as follows.

$$\overline{T}(b_{t+1}|b_t, a_t;\theta) = \iiint ds_t\, ds_{t+1}\, do_{t+1}\, B(s_t|b_t) T(s_{t+1}|s_t, a_t;\theta) O(o_{t+1}|s_{t+1};\theta) p(b_{t+1}|b_t, a_t, o_{t+1};\theta) \tag{4}$$

$$\overline{R}(b_t, a_t, b_{t+1};\theta) = \iint ds_t\, ds_{t+1} B(s_t|b_t) B(s_{t+1}|b_{t+1}) R(s_t, a_t, s_{t+1};\theta)$$

In (4), the belief update is expressed in a generalized form $p(b_{t+1}|b_t, a_t, o_{t+1};\theta)$ that allows either deterministic optimal belief updates, or could even account for other constraints on the inference process, including stochasticity.

The optimal action from a belief state will be also defined by a deterministic policy $\pi^*(a_t|b_t;\theta) = \delta(a_t^* = \arg\max_a Q(b_t, a;\theta))$. In case of continuous belief and action spaces, it is hard both to compute an optimal Q-function and to maximize it. Thus, we will approximate both using neural networks.

3.2 Training Bayesian optimal control ensembles with partial noisy observations

To successfully design and train an ensemble of agents, we identify three major challenges and provide solutions.

First, how can we construct the *optimal control ensembles* that can solve a family of tasks? As discussed, the task can be parameterized by the model parameter $\theta \in \Theta$ such that the family of tasks

shares the model structure but has different model parameters. We use this model parameter as an external input to flexible function approximators (neural networks) to estimate values and policies (Critic and Actor). Thus, the agent can be trained over parameter spaces. As presented in Figure 2, Critic and Actor both take parameter vector θ as an input, and respectively calculate the Q-value and best action for the task with that θ.

Second, how should we represent and update the agent's belief? For our concrete example application below, we use an extended Kalman filter [46] to provide a tractable Gaussian approximation for the belief state and its nonlinear dynamics. The resultant belief update is deterministic, $p(b_{t+1}|b_t, a_t, o_{t+1}; \theta) = \delta\left(b_{t+1} = f(b_t, a_t, o_{t+1}; \theta)\right)$. Tests with more flexible particle filters showed that this approximation is reasonable in our target application. For other applications, different belief representations and dynamics may be more accurate [47], and in principle a family of agents could use representational learning [48].

Lastly, how should we train a rational model agent ensemble with continuous belief and action spaces? Here we use the model-free deep reinforcement learning algorithm called Deep Deterministic Policy Gradient (DDPG) [49]. This method is able to approximate the value function over continuous belief states, actions, and task parameters, all using one neural network (the Critic), and uses it to train a policy network (the Actor) which also receives inputs about the current belief and task parameters. Viable alternatives for continuous control in the deep reinforcement learning literature include [50, 51, 52, 53].

The training process for optimal control ensembles is summarized in Algorithm 1, and a block diagram is provided in Figure 2. The agent is trained on simulated experiences. Given the belief b_t and parameters θ, the Actor returns the best action a_t. As the agent performs the action a_t, it changes the world state to s_{t+1} following the state transition probabilities T. The reward from the world R is given to the agent and fed back to the Critic to get a better estimation of the Q-value, which then improves the selection of the action in the Actor. From the new state s_{t+1}, the agent gets a partial and noisy observation o_{t+1} with the observation probabilities O. Then, the Gaussian belief state is updated using the extended Kalman filter f, $b_{t+1} = f(b_t, a_t, o_{t+1}; \theta)$. A new action a_{t+1} is selected by the Actor, and these processes are iterated until the neural networks are fully trained. During this training, we sample new model parameters θ every episode so the agent can experience the entire space of tasks, and thus generalize better over that space.

Algorithm 1: Train Bayesian optimal control ensembles

Initialization: Initialize Actor and Critic
repeat
  t = 0, Reset s_0, b_0
  Sample model parameter $\theta \sim$ prior $\mathcal{P}(\Theta)$
  repeat
    Select action $a_t \leftarrow$ Actor$(b_t; \theta)$
    Sample new state $s_{t+1} \sim T(s_{t+1}|s_t, a_t; \theta)$
    Get reward $r = R(s_t, a_t, s_{t+1}; \theta)$
    Train Critic by back-propagating r
    Calculate Q-value $q \leftarrow$ Critic$(b_t, a_t; \theta)$
    Train Actor by back-propagating q
    Sample new observation $o_{t+1} \sim O(o_{t+1}|s_{t+1}; \theta)$
    Update belief $b_{t+1} \leftarrow f(b_t, a_t, o_{t+1}; \theta)$ using the extended Kalman filter
    $t \leftarrow t + 1$
  until *episode ends*;
until *Actor and Critic are fully trained*;

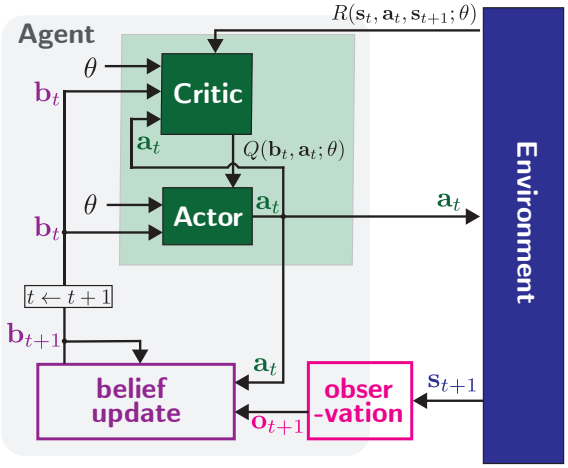

Figure 2: A block diagram of Algorithm 1.

4 Inverse Rational Control with Maximum Likelihood Estimation

Once an agent ensemble is fully trained over the entire parameter space, we can use this ensemble to find the internal model parameters of the best-fitting rational agent in that model family. We solve

the continuous Inverse Rational Control problem by finding the parameters θ that have the highest likelihood for explaining an agent's measured behavior.

4.1 Discrepancy between the true world and internal model

Recall that our core hypothesis is that animals have their own internal model of the world which may not be always correct, but they still behave *rationally*, choosing actions with the highest expected subjective reward according to their internal model. We must therefore distinguish between the two kinds of model parameters: the true ones ϕ which determine the world dynamics and are known to the experimentalist, and the agent's internal model parameters θ which are latent for the experimentalist but governs all cognitive processes of the agent (Figure 3). The world parameters ϕ govern the world dynamics such as state transition probability $T(s_{t+1}|s_t, a_t; \phi)$ and reward function $R(s_t, a_t, s_{t+1}; \phi)$. On the other hand, the internal parameters θ govern the agent's internal process such as the observation probability $O(o_t|s_t; \theta)$, the belief transition probability $\overline{T}(b_{t+1}|b_t, a_t; \theta)$, and the subjective reward as a function of belief $\overline{R}(b_t, a_t, b_{t+1}; \theta)$, leading to a subjective belief update probability $p(b_{t+1}|b_t, a_t, o_{t+1}; \theta)$ and rational policy $\pi(a_t|b_t; \theta)$.

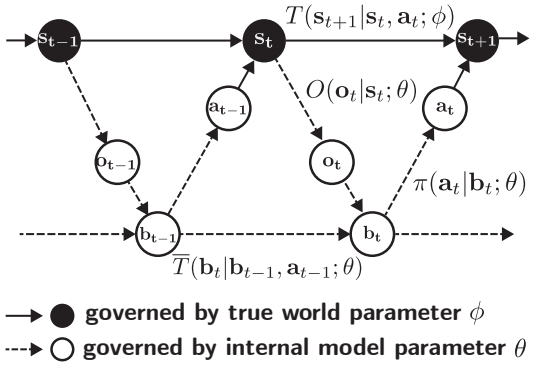

Figure 3: An illustrative explanation of model discrepancy. The solid lines and circles are governed by the true world parameter ϕ which is known to the experimentalist. The dashed lines and empty circles are governed by internal model parameter θ which is latent to the experimentalist, and may differ from ϕ.

4.2 Inferring internal model parameter θ

To find the internal model parameters θ that maximize the log-likelihood of the experimentally observable data $(s, a)_{1:T}$, $\hat{\theta} = \arg\max_\theta \ln p(s_{1:T}, a_{1:T}|\phi, \theta)$we use the Monte Carlo Expectation Maximization (MCEM) algorithm [54] to marginalize the complete data log-likelihood over latent observations $o_{1:T}$ and beliefs $b_{1:T}$. This yields an iterative algorithm, which repeatedly maximizes

$$\hat{\theta}_{k+1} = \arg\max_\theta \int do_{1:T}\, db_{1:T}\, p(o_{1:T}\, b_{1:T}|s_{1:T}, a_{1:T}; \theta_k) \ln p(s_{1:T}, o_{1:T}, b_{1:T}, a_{1:T}|\phi, \theta) \tag{5}$$

$$\approx \arg\max_\theta \frac{1}{L} \sum_{l=1}^{L} \ln p(s_{1:T}, o^{(l)}_{1:T}, b^{(l)}_{1:T}, a_{1:T}|\phi, \theta) \tag{6}$$

where the sum is over samples $(o^{(l)}, b^{(l)})_{1:T}$ drawn from a posterior distribution $p(o_{1:T}\, db_{1:T}|s_{1:T}, a_{1:T}; \theta_k)$ determined by parameters θ_k from previous iterations.

The log-likelihood of the complete data (including the l-th samples of observations and beliefs based on parameter θ_k) can be decomposed using the Markov property into

$$\ln p(s_{1:T}, o^{(l)}_{1:T}, b^{(l)}_{1:T}, a_{1:T}|\phi, \theta) \tag{7}$$

$$= \ln p(s_0, o^{(l)}_0, b^{(l)}_0, a_0) + \sum_{t=1}^{T} \Big(\ln T(s_t|s_{t-1}, a_{t-1}; \phi) + \ln O(o^{(l)}_t|s_t; \theta) + \ln p(b^{(l)}_t|b^{(l)}_{t-1}, a_{t-1}, o^{(l)}_t; \theta) + \ln \pi(a_t|b^{(l)}_t; \theta) \Big). \tag{8}$$

Note that the only terms depending on the agent's parameters θ are the latent observations probabilities, belief dynamics, and policy. So when we optimize over θ, all other terms vanish. Moreover, since we use deterministic belief updates, the belief update term in (8) is also independent of θ when evaluated on sampled beliefs. The only terms that survive are

$$\hat{\theta} = \arg\max_\theta \sum_{l=1}^{L} \sum_{t=1}^{T} \Big(\ln O(o^{(l)}_t|s_t; \theta) + \ln \pi(a_t|b^{(l)}_t; \theta) \Big). \tag{9}$$

To optimize (9), we use gradient ascent over parameter space, $\theta \leftarrow \theta + \alpha \nabla_\theta \mathcal{L}$ with learning rate α. This is explained in Algorithm 2.

Algorithm 2: Estimate θ that explains externally observable data the best

Data: Collected data by the experimentalist: $s_{0:T}, a_{0:T}$
T: the length of a trajectory
L: the number of samples
$b_0 = \mathcal{N}(o_0, 10^{-6})$
Initialization: Initialize θ with a random sample from the prior $\theta \sim \mathcal{P}(\Theta)$
repeat
  $\mathcal{L}(\theta) = 0$
  for $l = 1 : L$ **do**
    for $t = 1 : T$ **do**
      Sample $o_t^{(l)} \sim O(o_t^{(l)} | s_t; \theta)$
      Belief update using extended Kalman filter $b_t^{(l)} \leftarrow f(b_{t-1}^{(l)}, a_{t-1}, o_t^{(l)}; \theta)$
      $\mathcal{L}(\theta) \leftarrow \mathcal{L}(\theta) + \ln O(o_t^{(l)} | s_t; \theta) + \ln \pi(a_t | b_t^{(l)}; \theta)$
    end
  end
  Update θ using gradient ascent step: $\theta \leftarrow \theta + \alpha \bigtriangledown_\theta \mathcal{L}$
until $\mathcal{L}(\theta)$ *converges*;

5 Demonstration task: 'Catching fireflies'

To verify the proposed method, we carefully select a relevant task. Our application focus on continuous world states, actions, and beliefs makes standard RL testbeds (*e.g.* Nintendo, MuJoCo) more difficult. Common tasks like gridworld or tiger do not exhibit continuous properties and remain excessively small toy problems. Standard continuous control tasks do not use partially observability; tasks that do would likely generate beliefs that would be substantially harder to interpret. Additionally, there is a ready application to existing neuroscience experiments based on 'catching fireflies' in virtual reality [55, 39], which is complex enough to be interesting to animals, requires a continuous representation of uncertainty and continuous control, and yet remains tractable enough that we can assess the fidelity of recovered beliefs.

In our task, an agent must navigate through a virtual environment to reach a transiently visible target, called the 'firefly' (Figure 4A). At the beginning of each trial, a firefly blinks briefly at a random location on the ground plane. The agent is able to control its forward and angular velocities to freely navigate the world. If the agent stops sufficiently close to the invisible target, it receives a reward. As the agent moves, a sparse ground plane texture generates an optic flow pattern, a vector field of local image motion. This allows the agent to estimate its speed up to some perceptual uncertainty. However, there is no direct access to information about its current location because the ground plane texture is transient and does not provide spatial landmarks. Thus, the agent must integrate optic flow to estimate its current position relative to the firefly target, as well as its uncertainty about that position.

We demonstrate the efficacy of our approach using a simulated agent for which ground truth is known. Thus, we verify our method by showing the successful recovery of the internal model parameters since we know the ground truth. Note that there are no comparisons to alternative methods because no other algorithms exist that solve the IRC problem in continuous state and action spaces. Figure 4B shows a two-dimensional contour plot of the approximate log-likelihood of observable data $\mathcal{L}(\theta)$. Recall that the model parameters θ are high dimensional, so here we plot only two dimensions of θ. The red line shows an example trajectory of parameters θ as IRC Algorithm 2 converges. Our approach estimates θ that maximizes the log-likelihood of the observable data $\mathcal{L}(\theta)$. Figure 4C shows that the estimated parameters recovered by our algorithm closely match the agent's true parameters.

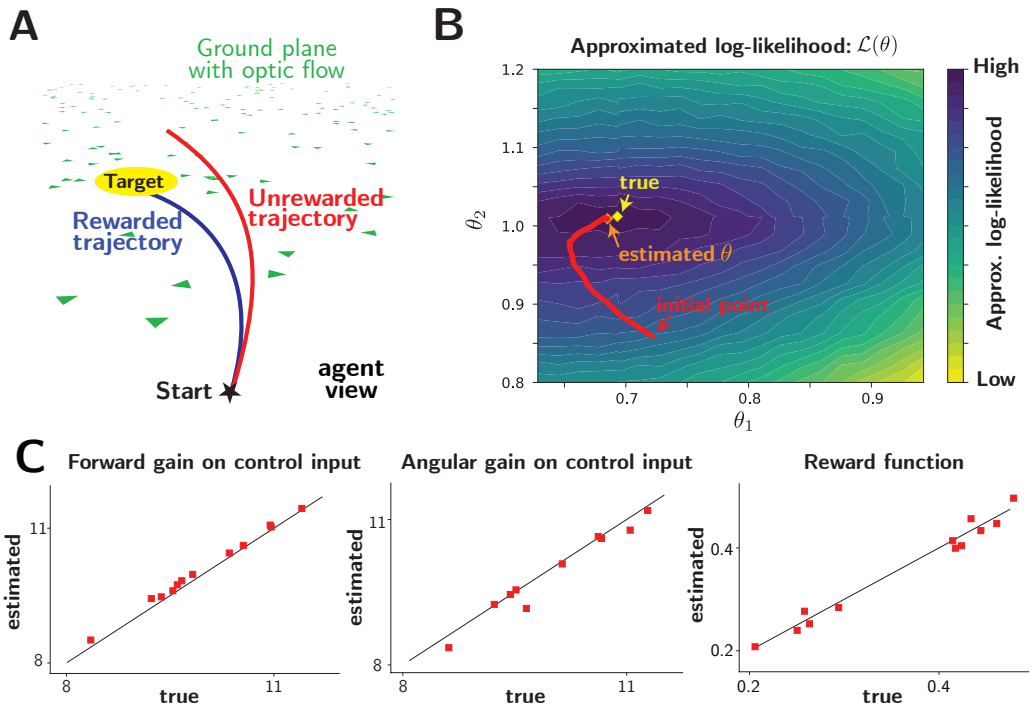

Figure 4: **A**. An illustration of the 'catching fireflies' task from the agent's point of view. To reach the transiently visible firefly target, an agent must navigate by optic flow over a dynamically textured ground plane. The agent is rewarded if it stops close enough to the target location. **B**. Converging trajectory of IRC estimates of the agent's parameters θ. We use gradient ascent to find θ that maximizes approximated log likelihood $\mathcal{L}(\theta)$ in Algorithm 2. **C**. Successful recovery of individual agent parameters. The black line is the identity, meaning that true values and estimated values are equal. Across all parameter spaces, the proposed approach accurately recovers the agent's internal model parameters given limited data.

6 Conclusion

This paper introduces a novel framework to infer the internal model of agents from their behaviors. We infer not only the subjective reward function of the agent, but we also simultaneously infer the task dynamics that the agent assumes. To accomplish this, we first train Bayesian optimal control ensembles that generalize over the space of task parameters. Since the target agent is only exposed to partial information about the world state, the agent chooses the best action based on its belief about the world and its assumptions about the task. By using this optimally trained agent ensemble, our approach to Inverse Rational Control with continuous state and action spaces can infer the internal model parameters that best explain the collected behavioral data. With a simulated agent where we know the ground truth, we confirm that our approach successfully recovers the internal models. This success encourages us to apply this method to real behavioral data as well as to new tasks and applications.

Broader Impact

We have implemented IRC for neuroscience applications, but the core principles have value in other fields as well. We can view IRC as a form of Theory of Mind, whereby one agent (a neuroscientist) creates a model of another agent's mind (for a behaving animal). Theory of Mind is a prominent component of human social interactions, and imputing rational motivations to actions provides a useful description of how people think [56, 57, 58]. Using IRC methods to provide a better understanding of people's motivations could yield important insights for understanding and improving social and political interactions, as well as raising possible ethical concerns if used for manipulation. The design

of agents interacting with humans would also benefit from being able to attribute rational strategies to others. For example, recent work uses a related approach to impute purpose to a neural network [16]. One important practical example is self-driving cars, which currently struggle to handle the perceived unpredictability of humans. While humans do indeed behave unpredictably, some of this may stem from ignorance of the rational computation that drives their actions. The IRC provides a framework for interpreting agents, and serves as a valuable tool for greater understanding of unifying principles of control.

References

[1] Scott E Fahlman, Geoffrey E Hinton, and Terrence J Sejnowski. Massively parallel architectures for al: Netl, thistle, and boltzmann machines. In *National Conference on Artificial Intelligence, AAAI*, 1983.

[2] Stuart Russell. Learning agents for uncertain environments. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 101–103. ACM, 1998.

[3] Jaedeug Choi and Kee-Eung Kim. Inverse reinforcement learning in partially observable environments. *Journal of Machine Learning Research*, 12(Mar):691–730, 2011.

[4] Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 897–904, 2011.

[5] Krishnamurthy Dvijotham and Emanuel Todorov. Inverse optimal control with linearly-solvable mdps. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 335–342, 2010.

[6] Felix Schmitt, Hans-Joachim Bieg, Michael Herman, and Constantin A Rothkopf. I see what you see: Inferring sensor and policy models of human real-world motor behavior. In *AAAI*, pages 3797–3803, 2017.

[7] Z. Wu, M. Kwon, S. Daptardar, P. Schrater, and X. Pitkow. Rational thoughts in neural codes. *Proceedings of the National Academy of Sciences of the United States of America*, 2020.

[8] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*. MIT press, 1998.

[9] Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

[10] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[11] Rajesh PN Rao. Decision making under uncertainty: a neural model based on partially observable markov decision processes. *Frontiers in computational neuroscience*, 4:146, 2010.

[12] Herbert A Simon. Theories of bounded rationality. *Decision and organization*, 1(1):161–176, 1972.

[13] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[14] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[15] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736, 2006.

[16] Matthew Chalk, Gašper Tkačik, and Olivier Marre. Inferring the function performed by a recurrent neural network. *bioRxiv*, page 598086, 2019.

[17] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[18] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

[19] Marcell Vazquez-Chanlatte, Susmit Jha, Ashish Tiwari, Mark K Ho, and Sanjit Seshia. Learning task specifications from demonstrations. In *Advances in Neural Information Processing Systems*, pages 5367–5377, 2018.

[20] Dexter RR Scobee and S Shankar Sastry. Maximum likelihood constraint inference for inverse reinforcement learning. In *Proceedings of 8th International Conference on Learning Representations*, 2020.

[21] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[22] Wonseok Jeon, Seokin Seo, and Kee-Eung Kim. A bayesian approach to generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2018.

[23] Balaraman Ravindran and Sergey Levine. Adail: Adaptive adversarial imitation learning. In *NeurIPS Workshop on Learning Transferable Skills*, 2019.

[24] Tanmay Gangwani and Jian Peng. State-only imitation with transition dynamics mismatch. In *Proceedings of 8th International Conference on Learning Representations*, 2020.

[25] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135. JMLR. org, 2017.

[26] Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[27] Ferran Alet, Martin F Schneider, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Meta-learning curiosity algorithms. In *Proceedings of 8th International Conference on Learning Representations*, 2020.

[28] Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J Smola. Meta-q-learning. In *Proceedings of 8th International Conference on Learning Representations*, 2020.

[29] Jan Humplik, Alexandre Galashov, Leonard Hasenclever, Pedro A Ortega, Yee Whye Teh, and Nicolas Heess. Meta reinforcement learning as task inference. In *Proceedings of 7th International Conference on Learning Representations*, 2019.

[30] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018.

[31] Kenji Doya, Shin Ishii, Alexandre Pouget, and Rajesh PN Rao. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

[32] Peter Dayan and Nathaniel D Daw. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience*, 8(4):429–453, 2008.

[33] Yanping Huang and Rajesh PN Rao. Reward optimization in the primate brain: A probabilistic model of decision making under uncertainty. *PloS one*, 8(1), 2013.

[34] Neil MT Houlsby, Ferenc Huszár, Mohammad M Ghassemi, Gergő Orbán, Daniel M Wolpert, and Máté Lengyel. Cognitive tomography reveals complex, task-independent mental representations. *Current Biology*, 23(21):2169–2175, 2013.

[35] Jean Daunizeau, Hanneke EM Den Ouden, Matthias Pessiglione, Stefan J Kiebel, Klaas E Stephan, and Karl J Friston. Observing the observer (i): meta-bayesian models of learning and decision-making. *PloS one*, 5(12), 2010.

[36] Jean Daunizeau, Hanneke EM Den Ouden, Matthias Pessiglione, Stefan J Kiebel, Karl J Friston, and Klaas E Stephan. Observing the observer (ii): deciding when to decide. *PLoS one*, 5(12), 2010.

[37] Jeffrey M Beck, Wei Ji Ma, Xaq Pitkow, Peter E Latham, and Alexandre Pouget. Not noisy, just wrong: the role of suboptimal inference in behavioral variability. *Neuron*, 74(1):30–39, 2012.

[38] Kaushik J. Lakshminarasimhan, Marina Petsalis, Hyeshin Park, Gregory C. DeAngelis, Xaq Pitkow, and Dora E. Angelaki. A dynamic bayesian observer model reveals origins of bias in visual path integration. *Neuron*, 99(1):194 – 206.e5, 2018.

[39] Kaushik J Lakshminarasimhan, Eric Avila, Erin Neyhart, Gregory C DeAngelis, Xaq Pitkow, and Dora E Angelaki. Tracking the mind's eye: Primate gaze behavior during virtual visuomotor navigation reflects belief dynamics. *Neuron*, pages 662–674, May 2020.

[40] Matthew Crosby, Benjamin Beyret, and Marta Halina. The animal-ai olympics. *Nature Machine Intelligence*, 1(5):257–257, 2019.

[41] The animal-AI testbed. `http://animalaiolympics.com/AAI/`.

[42] Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

[43] Ronald A Howard. Dynamic programming and markov processes. 1960.

[44] JAEE Van Nunen. A set of successive approximation methods for discounted markovian decision problems. *Zeitschrift fuer operations research*, 20(5):203–208, 1976.

[45] Martin L Puterman and Moon Chirl Shin. Modified policy iteration algorithms for discounted markov decision problems. *Management Science*, 24(11):1127–1137, 1978.

[46] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[47] Eszter Vértes and Maneesh Sahani. Flexible and accurate inference and learning for deep generative models. In *Advances in Neural Information Processing Systems*, pages 4166–4175, 2018.

[48] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[49] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *ICLR*, 2016.

[50] Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Reinforcement learning in continuous action spaces through sequential monte carlo methods. In *Advances in neural information processing systems*, pages 833–840, 2008.

[51] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[52] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[53] Riley Simmons-Edler, Ben Eisner, Eric Mitchell, Sebastian Seung, and Daniel Lee. Q-learning for continuous actions with cross-entropy guided policies. *arXiv preprint arXiv:1903.10605*, 2019.

[54] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[55] Kaushik J Lakshminarasimhan, Marina Petsalis, Hyeshin Park, Gregory C DeAngelis, Xaq Pitkow, and Dora E Angelaki. A dynamic bayesian observer model reveals origins of bias in visual path integration. *Neuron*, 2018.

[56] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

[57] Anna N Rafferty, Michelle M LaMar, and Thomas L Griffiths. Inferring learners' knowledge from their actions. *Cognitive Science*, 39(3):584–618, 2015.

[58] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[59] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Appendix

A Derivation of Monte Carlo Expectation Maximization (MCEM)

A derivation from (5) to (6) is based on MCEM. We here provide more details of the MCEM.

Let x be the observable data, z be the latent variable and θ be the parameters that govern the process. The goal is to find θ that maximizes the log likelihood of the observable data.

$$\theta = \arg\max_{\theta} \ln p(x|\theta)$$

The log likelihood of the observable data can be reformulated as follows.

$$\begin{aligned}
\ln p(x|\theta) &= \int dz\, q(z) \ln p(x|\theta) \\
&= \int dz\, q(z) \big[\ln p(x,z|\theta) - \ln p(z|x,\theta) \big] \\
&= \int dz\, q(z) \big[\ln p(x,z|\theta) - \ln q(z) + \ln q(z) - \ln p(z|x,\theta) \big] \\
&= \int dz\, q(z) \Big[\ln \frac{p(x,z|\theta)}{q(z)} - \ln \frac{p(z|x,\theta)}{q(z)} \Big] \\
&= \int dz\, q(z) \ln \frac{p(x,z|\theta)}{q(z)} - \int dz\, q(z) \ln \frac{p(z|x,\theta)}{q(z)} \qquad (10) \\
&= \mathcal{L}(q,\theta) + KL(q||p) \qquad (11)
\end{aligned}$$

Since KL divergence is always non-negative value, $\mathcal{L}(q,\theta)$ is the lower bound of $\ln p(x|\theta)$. The complete data log likelihood $\ln p(x,z|\theta)$ is easier to handle than the observed data log likelihood $\ln p(x|\theta)$. Thus, instead of maximizing $\ln p(x|\theta)$, we aim to maximize its lower bound $\mathcal{L}(q,\theta) = \int dz\, q(z) \ln \frac{p(x,z|\theta)}{q(z)}$.

A.1 E-step

As $KL(q||p)$ gets smaller, we have a tighter lower bound. If $KL(q||p) = 0$, $\ln p(x|\theta) = \mathcal{L}(q,\theta)$. $KL(q||p) = 0$ is satisfied only if $q = p$. Thus $q(z) = p(z|x,\theta)$ from (10). This is the E-step of the EM algorithm [59]. Note that in this step, $q(z)$ is a function only of z, which means both x and θ are used as given variables. Thus, we denote θ_{old} as a fixed parameter that is used to specify $q(z)$. Once $q(z) = p(z|x,\theta_{\text{old}})$ is used in $\mathcal{L}(q,\theta)$ of (11), $\ln p(x|\theta)$ can be expressed as follows.

$$\begin{aligned}
\ln p(x|\theta) &= \mathcal{L}(q,\theta) \\
&= \int dz\, p(z|x,\theta_{\text{old}}) \ln \frac{p(x,z|\theta)}{p(z|x,\theta_{\text{old}})} \\
&= \int dz\, p(z|x,\theta_{\text{old}}) \ln p(x,z|\theta) - \int dz\, p(z|x,\theta_{\text{old}}) \ln p(z|x,\theta_{\text{old}}) \\
&= \int dz\, p(z|x,\theta_{\text{old}}) \ln p(x,z|\theta) + H(z|x,\theta_{\text{old}}) \\
&= \mathcal{Q}(\theta,\theta_{\text{old}}) + H(z|x,\theta_{\text{old}}) \qquad (12)
\end{aligned}$$

A.2 M-step

Next, we want to find θ that maximizes $\ln p(x|\theta)$. This is the M-step of the EM algorithm. Since $H(z|x,\theta_{\text{old}})$ is a constant (i.e., not a function of θ),

$$\begin{aligned}
\theta &= \arg\max_{\theta} \ln p(x|\theta) \\
&= \arg\max_{\theta} \mathcal{Q}(\theta,\theta_{\text{old}}) \\
&= \arg\max_{\theta} \int dz\, p(z|x,\theta_{\text{old}}) \ln p(x,z|\theta). \qquad (13)
\end{aligned}$$

If $p(z|x, \theta_{\text{old}})$ is hard to get analytically, (13) can be approximated by the Monte Carlo approach. The resultant optimization is called the MCEM algorithm.

$$\begin{aligned}\theta &= \arg\max_{\theta} \int dz\, p(z|x, \theta_{\text{old}}) \ln p(x, z|\theta) \\ &\approx \arg\max_{\theta} \frac{1}{L} \sum_{l=1}^{L} \ln p(x, z^{(l)}|\theta) \end{aligned} \tag{14}$$

where z^l is l-th particle for the latent variable z.