Robust Inverse Reinforcement Learning Through Bayesian Theory of Mind

Anonymous Author(s)
Affiliation
Address
email

Abstract: We consider the Bayesian theory of mind (BTOM) framework for learning from demonstrations via inverse reinforcement learning (IRL). The BTOM model consists of a joint representation of the agent's reward function and the agent's internal *subjective* model of the environment dynamics, which may be inaccurate. In this paper, we make use of a class of prior distributions that parameterize how accurate the agent's model of the environment is to develop efficient algorithms to estimate the agent's reward and subjective dynamics in high-dimensional settings. The BTOM framework departs from existing offline model-based IRL approaches by performing *simultaneous* estimation of reward and dynamics. Our analysis reveals a novel insight that the estimated policy exhibits robust performance when the agent is believed (a priori) to have a highly accurate model of the environment. We verify this observation in the MuJoCo environment and show that our algorithms outperform state-of-the-art offline IRL algorithms.

Keywords: Inverse Reinforcement Learning, Theory of Mind, Robust Control

1 Introduction

Inverse reinforcement learning (IRL) is the problem of extracting the reward function and policy of a value-maximizing agent from its behavior [1, 2]. IRL is an important tool in domains where manually specifying reward functions or policies is difficult, such as in autonomous driving [3], or when the extracted reward function can reveal novel insight about a target population and device interventions, such as in biology, economics, and human-robot interaction studies [4, 5, 6]. However, wider application of IRL faces two interrelated algorithmic challenges: 1) having access to the target deployment environment or an accurate simulator thereof and 2) robustness of the learned policy and reward function due to the covariate shift between the training and deployment environments or dataset [7, 8, 9].

In this paper, we focus on model-based offline IRL to address challenge 1). A notable class of model-based offline IRL methods estimate the dynamics and reward in a two-stage, *decoupled* fashion [10, 11, 12, 13]. In the first stage, a dynamics model is estimated from the fixed dataset. Then, parameters of the dynamics model are fixed while training the reward and policy in the second stage. To overcome covariate shift in the estimated dynamics, recent methods design density estimation-based "pessimistic" penalties to prevent the learner policy from entering uncertainty regions in the state-action space (i.e., space not covered in the demonstration dataset) [14, 13, 12].

We instead approach IRL from the Bayesian Theory of Mind perspective [15], where we *simultaneously* estimate the expert's reward function and their *internal* model of the environment dynamics. The core idea of BTOM is that expert decisions convey their beliefs about the environment [15] and thus should affect the update direction of the dynamics model. BTOM has mostly been used to understand human biases encoded in the internal dynamics in simple and highly constrained domains

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

[16, 17, 18, 19, 20, 21, 22]. In contrast to these works, we study how BTOM enables learning high-performance and robust policies.

We first propose a class of priors parameterizing how accurate we believe the expert's model of the environment is. We then show that if the expert is believed a priori to have a highly accurate model, robustness emerges naturally from BTOM's *simultaneous* estimation approach by planning against the worst-case dynamics outside the offline data distribution. We further analyze how varying the prior affects the performance of the learner agent and pair our analysis with an set of algorithms which extend prior simultaneous estimation approaches [18, 19] to high-dimensional continuous-control settings. We show that the proposed algorithms outperform state-of-the-art (SOTA) offline IRL methods in the MuJoCo environments without the need for designing pessimistic penalties.

2 Preliminaries

2.1 Markov Decision Process

We consider modeling agent behavior using infinite-horizon *entropy-regularized* Markov decision processes (MDP; [23]) defined by tuple $(\mathcal{S}, \mathcal{A}, P, R, \mu, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability distribution $P(s'|s,a) \in \Delta(\mathcal{S})$, reward function $R(s,a) \in \mathbb{R}$, initial state distribution $\mu(s_0) \in \Delta(\mathcal{S})$, and discount factor $\gamma \in (0,1)$. We denote the discounted occupancy measure as $\rho_P^\pi(s,a) = \mathbb{E}_{\mu,P,\pi}\left[\sum_{t=0}^{\infty} \gamma^t P(s_t = s, a_t = a)\right]$ and the marginal state-action distribution as $d_P^\pi(s,a) = (1-\gamma)\rho_P^\pi(s,a)$. We further denote the discounted occupancy measure starting from a specific state-action pair (s,a) with $\rho_P^\pi(\tilde{s}, \tilde{a}|s,a)$. The agent selects actions from an optimal policy $\pi(a|s) \in \Delta(\mathcal{A})$ that achieves the maximum expected discounted cumulative rewards and policy entropy $\mathcal{H}(\pi(a|s)) = -\sum_{\tilde{a}} \pi(\tilde{a}|s) \log \pi(\tilde{a}|s)$ in the MDP:

$$\max_\pi J_P(\pi) = \mathbb{E}_{\mu,P,\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t,a_t) + \mathcal{H}(\pi(a_t|s_t))\right)\right] \tag{1}$$

The optimal policy satisfies the following conditions [24]:

$$\begin{aligned} \pi(a|s) &\propto \exp\left(Q(s,a)\right) \\ Q(s,a) &= R(s,a) + \gamma \mathbb{E}_{P(s'|s,a)}\left[V(s')\right] \\ V(s) &= \log \sum_{a'} \exp\left(Q(s,a')\right) \end{aligned} \tag{2}$$

2.2 Inverse Reinforcement Learning

The majority of contemporary IRL approaches have converged on the Maximum Causal Entropy (MCE) IRL framework, which aims to find a reward function $R_\theta(s.a)$ with parameters θ such that the entropy-regularized learner policy $\hat{\pi}$ has matching state-action feature with the unknown expert policy π [25].

A related formulation casts IRL as maximum *discounted* likelihood (ML) estimation [26, 27, 28], subject to the constraint that the policy is entropy-regularized. Given a dataset of N expert trajectories each of length T: $\mathcal{D} = \{\tau_i\}_{i=1}^N, \tau = (s_{1:T}, a_{1:T})$ sampled from the expert policy in environment P with occupancy measure $\rho_\mathcal{D} := \rho_P^\pi$, ML-IRL aims to solve the following optimization problem:

$$\max_\theta \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\sum_{t=0}^{\infty} \gamma^t \log \hat{\pi}_\theta(a_t|s_t)\right], \quad \text{s.t. } \hat{\pi}_\theta(a|s) = \arg\max_{\hat{\pi}\in\Pi} \mathbb{E}_{\rho_P^{\hat{\pi}}}\left[R_\theta(s,a) + \mathcal{H}(\hat{\pi}(\cdot|s)\right] \tag{3}$$

where the policy is implicitly parameterized by the reward parameters θ.

It can be shown that MCE-IRL and ML-IRL are equivalent under linear reward parameterization [26, 27], however (3) permits non-linear reward parameterization through the following surrogate optimization problem:

$$\max_\theta \mathbb{E}_{\rho_\mathcal{D}}\left[R_\theta(s,a)\right] - \mathbb{E}_{\rho_P^{\hat{\pi}}}\left[R_\theta(s,a)\right], \quad \text{s.t. } \hat{\pi}_\theta(a|s) = \arg\max_{\hat{\pi}\in\Pi} \mathbb{E}_{\rho_P^{\hat{\pi}}}\left[R_\theta(s,a) + \mathcal{H}(\hat{\pi}(\cdot|s)\right] \tag{4}$$

(4) can be efficiently solved via alternating training of the learner policy and the reward function, similar to Generative Adversarial Network (GAN)-based algorithms [29, 30, 31, 32, 33, 34]. However, these methods all require access to the ground truth environment dynamics or a high quality simulator in order to compute or sample from the learner occupancy measure ρ_P^π.

2.3 Offline Model-Based IRL & RL

Existing offline model-based IRL algorithms such as [10, 11] adapt (4) using a two-step process. First, an estimate $\hat{P}$ of the environment dynamics is obtained from the offline dataset, e.g., using maximum likelihood estimation. Then, $\hat{P}$ is fixed and used in place of P to compute $\rho_{\hat{P}}^\pi$ while optimizing (4). However, this simple replacement incurs a gap between (4) and (3) which scales with the dynamics model error and the estimated value [13]. This puts a high demand on the accuracy of the estimated dynamics.

A related challenge is to prevent the policy from exploiting inaccuracies in the estimated dynamics, which can lead to erroneously high estimated value. This has been extensively studied in both online and offline model-based RL literature [35, 36, 37, 38]. The majority of recent offline model-based RL methods combat model-exploitation via a notion of "pessimism", which penalizes the learner policy from visiting states where the model is likely to be incorrect [35]. These pessimistic penalties are often designed based on quantifying uncertainty about transition dynamics through the estimated model [39, 40]. Drawing on these advances, recent offline IRL methods also incorporate pessimistic penalties into their RL subroutine [13, 12, 14]. However, it should be noted that designing pessimistic penalties involves nontrivial decisions to ensure that they can accurately capture out-of-distribution samples [41].

An orthogonal approach to avoid model-exploitation is to perform policy training against the worst-case dynamics in out-of-distribution states [42], similar to robust MDP [43, 44]. Rigter et al. [45] implemented this idea in the RAMBO algorithm and showed that it is competitive with pessimistic penalty-based approaches while requiring significantly less tuning. We will show that robust MDP corresponds to a sub-problem of IRL under the BTOM formulation.

We list additional related work in Appendix A.

3 Bayesian Theory of Mind

We consider IRL under the Bayesian Theory of Mind framework, where the observed expert decisions are the results of the unknown reward function $R_{\theta_1}(s,a)$ and their *internal* model of the environment dynamics $\hat{P}_{\theta_2}(s'|s,a)$. Importantly, we do not assume the expert internal parameters θ_2 are equal to the true environment parameters. We denote the concatenated parameters with $\theta = \{\theta_1, \theta_2\}$ and condition the policy on θ as $\hat{\pi}(a|s;\theta)$ to emphasize that the expert configuration is determined by both the reward and dynamics parameters.

Upon observing a finite set of expert demonstrations $\mathcal{D}$, BTOM aims to compute the posterior distribution $\mathbb{P}(\theta|\mathcal{D})$ given a choice of a prior distribution $\mathbb{P}(\theta)$:

$$\mathbb{P}(\theta|\mathcal{D}) \propto \mathbb{P}(\mathcal{D}|\theta)\mathbb{P}(\theta) = \prod_{i=1}^{N}\prod_{t=1}^{T} \hat{\pi}(a_{i,t}|s_{i,t};\theta)\mathbb{P}(\theta) \tag{5}$$

where we have omitted $\prod_{i=1}^{N}\prod_{t=1}^{T} P(s_{i,t+1}|s_{i,t},a_{i,t})$ from the likelihood because it does not depend on θ.

We consider a class of prior distributions of the form:

$$\mathbb{P}(\theta) \propto \exp\left(\lambda \sum_{i=1}^{N}\sum_{t=1}^{T} \log \hat{P}_{\theta_2}(s_{i,t+1}|s_{i,t},a_{i,t})\right) \tag{6}$$

where the prior precision hyperparameter λ represents how accurate we believe is the expert's model of the environment.

Let $\mathcal{L}(\theta) := \frac{1}{NT} \log \mathbb{P}(\theta|\mathcal{D})$. It can be easily verified that

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\log \hat{\pi}(a|s;\theta) + \lambda \log \hat{P}_{\theta_2}(s'|s,a)\right]$$

In this paper, we consider finding a Maximum A Posteriori (MAP) estimate of the BTOM model by solving the following bi-level optimization problem:

$$\max_{\theta} \mathcal{L}(\theta), \quad \text{s.t. } \hat{\pi}(a|s;\theta) = \arg\max_{\hat{\pi}\in\Pi} \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}} \left[R_\theta(s,a) + \mathcal{H}(\hat{\pi}(\cdot|s))\right] \tag{7}$$

Note that this formulation differs from (3) and the decoupled approaches because it includes log likelihood of the dynamics in the objective (weighted by λ).

It should be noted that obtaining the full posterior distribution (or an approximation) is feasible using popular approximate inference methods (e.g., stochastic variational inference or Langevin dynamics [46, 47]) and does not significantly alter the proposed estimation principles and algorithms.

3.1 Naive Solution

We start by presenting a naive solution to (7) which can be seen as an extension of the tabular simultaneous reward-dynamics estimation algorithms [18, 19] to the high-dimensional setting.

Solving (7) requires: 1) computing the optimal policy with respect to θ, and 2) computing the gradient $\nabla_\theta \log \hat{\pi}(a|s;\theta)$ which requires inverting the policy optimization process itself. Both operations can be done exactly in the tabular setting as in prior works but are intractable in high-dimensional settings. We propose to overcome the intractability using sample-based approximation.

In this section, we focus on approximating the gradient of the policy $\nabla_\theta \log \hat{\pi}(a|s;\theta)$, which is less obvious. We can show that the $\nabla_\theta \log \hat{\pi}(a|s;\theta)$ has the following form (see Appendix B for all proofs and derivations):

$$\begin{aligned}\nabla_\theta \log \hat{\pi}(a|s;\theta) &= \nabla_\theta Q_\theta(s,a) - \nabla_\theta V_\theta(s) \\ &= \nabla_\theta Q_\theta(s,a) - \mathbb{E}_{\tilde{a}\sim\hat{\pi}(\cdot|s;\theta)}[\nabla_\theta Q_\theta(s,\tilde{a})]\end{aligned} \tag{8}$$

where $\nabla_\theta Q_\theta(s,a) = [\nabla_{\theta_1} Q_\theta(s,a), \nabla_{\theta_2} Q_\theta(s,a)]$ is the concatenation of reward and dynamics gradients defined as:

$$\nabla_{\theta_1} Q_\theta(s,a) = \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)} \left[\nabla_{\theta_1} R_{\theta_1}(\tilde{s},\tilde{a})\right] \tag{9}$$

$$\nabla_{\theta_2} Q_\theta(s,a) = \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)} \left[\gamma \sum_{s'} V_\theta(s') \nabla_{\theta_2} \hat{P}_{\theta_2}(s'|\tilde{s},\tilde{a})\right] \tag{10}$$

Given (9) and (10) are tractable to compute using sample-based approximation of expectations, we construct the following surrogate objective $\tilde{\mathcal{L}}(\theta)$ with the same gradient as the original MAP estimation problem (7):

$$\tilde{\mathcal{L}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\mathcal{E}_\theta(s,a)] - \mathbb{E}_{s\sim\mathcal{D},a\sim\hat{\pi}}[\mathcal{E}_\theta(s,a)] + \lambda \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[\log \hat{P}_{\theta_2}(s'|s,a)] \tag{11}$$

where

$$\mathcal{E}_\theta(s,a) = \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)} \left[R_\theta(\tilde{s},\tilde{a}) + \gamma EV_\theta(\tilde{s},\tilde{a})\right] \tag{12}$$

$$EV_\theta(s,a) = \sum_{s'} \hat{P}_{\theta_2}(s'|s,a) V_\theta(s') \tag{13}$$

Optimizing (11) (subject to the same policy constraint) is now the same as optimizing (7) but tractable.

An interesting consequence of maximizing the first two terms of (11) alone (excluding the prior) is that we both increase the reward and modify the internal dynamics to generate states with higher expected value (EV) upon taking expert actions then following the learner policy $\hat{\pi}$, and we do the opposite when taking learner actions. Intuitively, reward and dynamics play complementary roles in determining the value of actions and thus should be regularized [48, 17, 49]. Otherwise, one cannot disentangle the effect of truly high reward and falsely optimistic dynamics. Our prior (6) alleviates this unidentifiability to some extent.

3.2 A Robust BTOM Model

We now present our main observation that robustness emerges from the BTOM formulation under the dynamics accuracy prior (6).

We start by analyzing a discounted, full-trajectory version of the BTOM likelihood (7). Note that discounting does not change the optimal solution to (7) under expressive reward and dynamics model class; nor does (3) require infinite data because we can truncate the summation at $T = \text{int}\left(\frac{1}{1-\gamma}\right)$ and obtain nearly the same estimator as with infinite sequence length. We restate a decomposition of the discounted likelihood in [13] as follows (see Appendix B.2):

$$\begin{aligned}
&\mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \log \hat{\pi}(a_t|s_t;\theta)\right] = \mathbb{E}_{P(\tau)}\left[\sum_{t=0}^{\infty}\gamma^t \left(Q_\theta(s_t,a_t) - V_\theta(s_t)\right)\right] \\
&= \underbrace{\mathbb{E}_{\rho_P^\pi}\left[R_{\theta_1}(s_t,a_t)\right] - \mathbb{E}_\mu\left[V_\theta(s_0)\right]}_{\ell(\theta)} + \underbrace{\gamma\mathbb{E}_{\rho_P^\pi}\left[\mathbb{E}_{s'\sim\hat{P}_{\theta_2}(\cdot|s_t,a_t)}V_\theta(s') - \mathbb{E}_{s''\sim P(\cdot|s_t,a_t)}V_\theta(s'')\right]}_{\textbf{T1}}
\end{aligned} \tag{14}$$

where **T1** corresponds to the value difference under the real and estimated dynamics. We can show that **T1** is negligible if the estimated dynamics is accurate, e.g. $\mathbb{E}_{(s,a)\sim P(\tau)}D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a)) \leq \epsilon$ for sufficiently small ϵ, under the *expert* data distribution, which can be achieved by setting a large λ. Then, **T1** can be dropped and the discounted likelihood reduces to $\ell(\theta)$.

$\ell(\theta)$ highlights the reason why the proposed BTOM approach can be robust to limited dataset. It poses the offline IRL problem as maximizing the cumulative reward of expert trajectories in the real environment, and minimizing the cumulative reward generated by the learner in the estimated dynamics with respect to *both* reward and dynamics. In other words, it aims to find performance-matching reward and policy under the *worst-case* dynamics, which is trained adversarially outside the data distribution. This connects BTOM to the robust MDP approach to offline model-based RL [42, 45].

3.3 Proposed Algorithms

We now propose two scalable Deep Bayesian Theory of Mind algorithms to find the MAP solution to (7) via the surrogate objective (11). The first algorithm (**BTOM**; 1) applies the naive solution, while the second algorithm (**RTOM**; 2) exploits the observation in section 3.2 to derive a more efficient algorithm for high λ.

The estimation problem (7) has an inherently nested structure where, for each update of parameters θ (the outer problem), we have to solve for the optimal policy $\hat{\pi}(a|s;\theta)$ (the inner problem). Following recent ML-IRL approaches [27, 13], we perform the nested optimization using *two-timescale* stochastic approximation [50, 51], where the inner problem is solved via stochastic gradient updates on a faster time scale than the outer problem. For both algorithms, we solve the inner problem using Model-Based Policy Optimization (MBPO; [37]) which uses Soft Actor-Critic (SAC; [24]) in a dynamics model ensemble.

BTOM For the BTOM outer problem, we estimate the expectations in (11) and (12) via sampling and perform coordinate-ascent optimization. Specifically, for each update step, we first sample a mini-batch of state-action pairs $(s,a) \sim \mathcal{D}$ and a mini-batch of (fake) actions $a_{\text{fake}} \sim \hat{\pi}(\cdot|s;\theta)$ and simulate both (s,a) and (s,a_{fake}) forward in the estimated dynamics $\hat{P}$ to get the real and fake trajectories $\tau_{\text{real}}, \tau_{\text{fake}}$. We then optimize the reward function first by taking a single gradient step to optimize the following objective function:

$$\max_{\theta_1} \quad \mathbb{E}_{(s,a)\sim\mathcal{D},\rho_{\hat{P}}^{\hat{\pi}}(\tilde{s},\tilde{a}|s,a)}\left[R_{\theta_1}(\tilde{s},\tilde{a})\right] - \mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi}(\cdot|s;\theta),\rho_{\hat{P}}^{\hat{\pi}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[R_{\theta_1}(\tilde{s},\tilde{a})\right] \tag{15}$$

Lastly, we optimize the dynamics model by taking a few gradient steps (a hyperparameter) to optimize the following objective function using on-policy rollouts branched from mini-batches of

expert state-actions as in RAMBO [45]:

$$\begin{aligned}\max_{\theta_2} \quad & \lambda_1 \mathbb{E}_{(s,a)\sim\mathcal{D},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] - \lambda_1 \mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi}(\cdot|s,;\theta),\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] \\ & + \lambda_2 \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log \hat{P}_{\theta_2}(s'|s,a)\right]\end{aligned} \tag{16}$$

where we have introduced weighting coefficients λ_1 and λ_2 to facilitate tuning the prior precision λ and dynamics model learning rate. We estimate the dynamics gradient using the REINFORCE method with baseline.

RTOM We adapt the BTOM algorithm slightly for the RTOM outer problem, where we only simulate a single trajectory for each state in the mini-batch and update the reward using the following objective:

$$\max_{\theta_1} \mathbb{E}_{\rho_{\mathcal{D}}}\left[R_{\theta_1}(s,a)\right] - \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}}\left[R_{\theta_1}(s,a)\right] \tag{17}$$

We then update the dynamics by dropping the first term in (16):

$$\max_{\theta_2} \quad -\lambda_1 \mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi}(\cdot|s;\theta),\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] + \lambda_2 \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log \hat{P}_{\theta_2}(s'|s,a)\right] \tag{18}$$

We provide additional details and pseudo code for the proposed algorithms in Appendix C.

3.4 Performance Guarantees

In this section, we study how policy and dynamics estimation error affect learner performance in the real environment. Using lemma 4.1 from [52] which decomposes the real environment performance gap between the expert and the learner into their policy and model advantages in the estimated dynamics, we arrive at the follow performance bound (see Appendix B.3):

Theorem 3.1. *Let* $\epsilon_{\hat{\pi}} = -\mathbb{E}_{(s,a)\sim d^{\pi}_{P}}[\log \hat{\pi}_{\hat{P}}(a|s)]$ *be the policy estimation error and* $\epsilon_{\hat{P}} = \mathbb{E}_{(s,a)\sim d^{\pi}_{P}} D_{KL}[P(\cdot|s,a)||\hat{P}(\cdot|s,a)]$ *be the dynamics estimation error. Let* $R_{max} = \max_{s,a}|R_\theta(s,a)| + \log|\mathcal{A}|$*. Assuming bounded expert-learner marginal state-action density ratio* $\left\|\frac{d^{\hat{\pi}}_{P}(s,a)}{d^{\pi}_{P}(s,a)}\right\|_\infty \leq C$*, we have the following (absolute) performance bound for the IRL agent:*

$$|J_P(\hat{\pi}) - J_P(\pi)| \leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon_{\hat{P}}} \tag{19}$$

This bound highlights the connection between IRL and behavior cloning and the Bayesian nature of IRL: by incorporating the dynamics and Bellman-optimality as regularizations, we can achieve better generalizations than behavior cloning. We believe a tighter bound can be obtained by further analyzing the density ratio C given that the BTOM policy will act conservatively as a result of planning against worst-case dynamics. We leave this to future work.

4 Experiments

We aim to answer the following questions with our experiments:

1. How does the dynamics accuracy prior affect agent behavior?
2. How well does BTOM and RTOM perform compared to SOTA offline IRL algorithms?

We investigate Q1 using a Gridworld environment. We investigate Q2 using the standard D4RL dataset on MuJoCo continuous control benchmarks.

4.1 Gridworld Example

We use a 5x5 gridworld environment to understand the behavior of the BTOM algorithm. The environment has deterministic transitions conditioned on the following set of actions: up, down, left,

right, and stay. Any actions pointing in the direction of the boundary when the agent is already in a boundary cell will keep the agent in the same cell. The expert agent, who knows the true transition dynamics and plans using a discount factor of $\gamma = 0.7$, starts in the lower left corner and receives a reward when reaching the upper right corner. We represent the reward function as the log probability of the target state: $\log \tilde{P}(s)$, where the upper right corner has a target probability of 1.

Using 100 expert trajectories of length 50, we trained 3 BTOM agents with $\lambda \in \{0.001, 0.5, 10\}$. As a comparison, we also trained a decoupled agent whose dynamics model is fixed after an initial maximum likelihood pretraining step and its reward is estimated using the same gradient update rule as BTOM in (15).

Given that the environment is simple and both the policy, reward, and dynamics models are well-specified, all agents recover the ground-truth policy in state-actions pairs visited by the expert. The ground truth and estimated target state probabilities are shown in the first row of Figure 1. All agents correctly estimated that the upper right corner has the highest reward, although not with the same precision as the ground truth sparse reward. BTOM agents with $\lambda = 0.5$ and $\lambda = 10$ are able to assign high reward only to states close to the true goal state, where as the BTOM agent with $\lambda = 0.001$ and the decoupled agent assigned high rewards to state much further away from the true goal state.

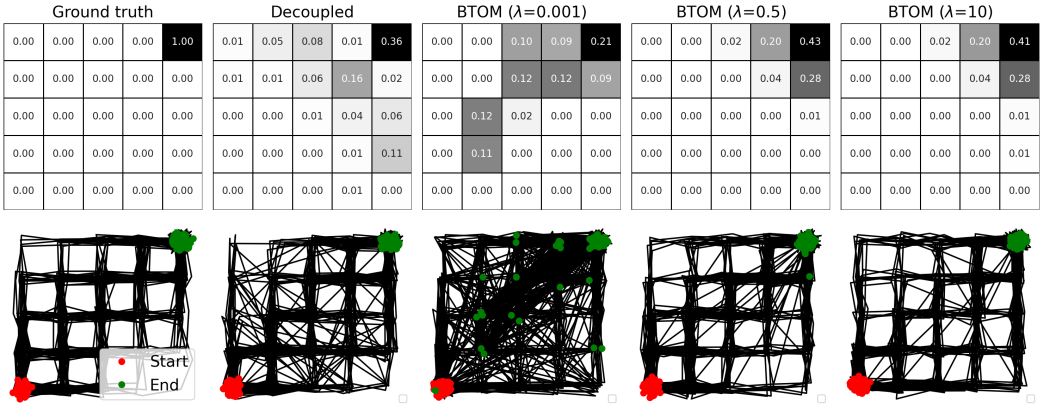

Figure 1: Gridworld experiment results. **(Row 1)** Ground truth and estimated target state distributions (softmax of reward) for agent using decoupled estimation and BTOM agents with $\lambda = [0.001, 0.5, 10]$. BTOM agents with higher λ obtain more accurate reward estimates. **(Row 2)** Sample paths generated by the ground truth agent, decoupled, and BTOM agents. BTOM agents with higher λ generate fewer illegal (diagonal) transitions. Illegal transitions generated by BTOM agents have a strong tendency to point towards the goal state.

Table 1: MuJoCo benchmark performance using 10 expert trajectories from the D4RL dataset. Each row reports the mean and standard deviation of performance over 5 random seeds.

Environment	Dataset	BTOM (ours)	RTOM (ours)	ML-IRL	Expert
HalfCheetah	Medium	$\mathbf{8813.35 \pm 997.49}$	8085.18 ± 597.86	7706.43 ± 159.39	12156.16 ± 88.01
HalfCheetah	Medium-replay	7508.65 ± 190.75	6961.28 ± 130.61	$\mathbf{9383.34 \pm 358.67}$	12156.16 ± 88.01
HalfCheetah	Medium-expert	$\mathbf{11519.98 \pm 149.69}$	11289.09 ± 258.70	11276.09 ± 551.94	12156.16 ± 88.01
Hopper	Medium	2243.15 ± 922.75	$\mathbf{3306.59 \pm 473.60}$	2461.45 ± 705.70	3512.64 ± 17.10
Hopper	Medium-replay	$\mathbf{3520.69 \pm 29.50}$	3307.11 ± 471.38	2889.73 ± 542.65	3512.64 ± 17.10
Hopper	Medium-expert	3209.91 ± 731.66	$\mathbf{3550.25 \pm 28.85}$	3350.79 ± 264.96	3512.64 ± 17.10
Walker2D	Medium	$\mathbf{4307.99 \pm 855.55}$	4035.21 ± 247.23	4195.36 ± 352.86	5365.62 ± 55.79
Walker2D	Medium-replay	3960.70 ± 1521.52	3880.54 ± 713.29	$\mathbf{4092.58 \pm 308.71}$	5365.62 ± 55.79
Walker2D	Medium-expert	4862.66 ± 100.37	$\mathbf{4941.10 \pm 38.99}$	4363.54 ± 729.60	5365.62 ± 55.79

We visualize the estimated dynamics models by sampling 100 imagined rollouts using the estimated policies in the second row of Figure 1. This figure shows that the BTOM($\lambda = 0.001$) and the decoupled agent would take significantly more illegal transitions (i.e., diagonal transitions) than BTOM agent with higher λ. Comparing among BTOM agents, we see that increasing λ decreases

the number of illegal transitions. In contrast to the decoupled agent whose illegal transitions are rather random, the illegal transitions generated by BTOM agents with lower λ have a strong tendency to point towards the goal state. This corroborates with our analysis that BTOM learns optimistic dynamics under the expert distribution.

4.2 MuJoCo Benchmarks

In this section, we compare the performance of BTOM and RTOM with SOTA offline IRL algorithms in the MuJoCo continuous control environments [53] using the D4RL dataset [54]. We use ML-IRL [13], an offline model-based IRL algorithms based on MOPO [39], as our comparison.

We use the following MuJoCo environments: HalfCheetah, Hopper, and Walker2D. For each environment, D4RL offers 4 types of datasets: medium, medium-replay, medium-expert, and expert. Following prior IRL evaluation protocols, our agents maintain two datasets: 1) a *transition dataset* is used to trained the dynamics model and the actor-critic networks and 2) an *expert dataset* is used to train the reward function. The transition dataset is selected from one of the first three types of D4RL datasets and is not sub-sampled. The expert dataset contains 10 randomly sampled D4RL expert trajectories. For both BTOM and RTOM, we set the model objective weighting terms to $\lambda_1 = 0.01, \lambda_2 = 1$ to encourage an accurate model under the data distribution. For each environment and transition dataset, we train our algorithms for a fixed number of epochs and repeat this process for 5 random seeds. After the final epoch, we evaluate the agent for 10 episodes in the MuJoCo environments. We provide additional implementation and hyperparameter details in Appendix D.

Table 1 reports the mean and standard deviation of the evaluation performance across different seeds for each setting. For ML-IRL, we list the results reported in the original publication. Our algorithms outperform the benchmark in almost all settings. On the medium-expert dataset, which has the best coverage of expert trajectories, our algorithms perform near optimally and overall have smaller variance than ML-IRL.

Between the two proposed algorithms, BTOM and RTOM perform comparably on the medium-expert datasets. However, BTOM outperforms RTOM on the medium and medium-replay datasets in the Halfcheetah and Walker2D environments. Training the dynamics model on these datasets corresponds to violating the dynamics accuracy assumption for $\ell(\theta)$ in (14) as **T1** would be large. For BTOM, this is not a problem because the dynamics log likelihood only serves as a prior and the surrogate objective (11) is not affected. However, for RTOM, relaxing the dynamics accuracy assumption causes $\ell(\theta)$ to deviate from the true objective.

5 Limitations

A limitation of the proposed algorithms is that BTOM can have less stable training dynamics than RTOM where its evaluation performance may alternate between periods of near optimal performance and periods of medium performance (thus the larger variance in Table 1). While stability is a known issue for training energy-based models using contrastive divergence objectives (i.e., objective (11); Du et al., 2020), we believe the current issue is related to BTOM's two-sample path method having weaker and noisier learning signal. Another source of instability is likely introduced by simultaneously training the dynamics model, which may be improved in future work by adding Lipschitz regularizations [56].

6 Conclusion

We showed that inverse reinforcement learning under the Bayesian Theory of Mind framework gives rise to robust policies. This yielded a set of novel offline model-based IRL algorithms achieving SOTA performance in the MuJoCo continuous control benchmarks without ad hoc pessimistic penalty design. An interesting future direction is to identify appropriate priors to robustly infer reward and internal dynamics from sub-optimal and biased human demonstrators.

Acknowledgements

References

[1] A. Y. Ng, S. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[2] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[3] T. Phan-Minh, F. Howington, T.-S. Chu, S. U. Lee, M. S. Tomov, N. Li, C. Dicle, S. Findler, F. Suarez-Ruiz, R. Beaudoin, et al. Driving in real life with inverse reinforcement learning. *arXiv preprint arXiv:2206.03004*, 2022.

[4] S. Yamaguchi, H. Naoki, M. Ikeda, Y. Tsukada, S. Nakano, I. Mori, and S. Ishii. Identification of animal behavioral strategies by inverse reinforcement learning. *PLoS computational biology*, 14(5):e1006122, 2018.

[5] J. Rust. Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica: Journal of the Econometric Society*, pages 999–1033, 1987.

[6] D. Sadigh, N. Landolfi, S. S. Sastry, S. A. Seshia, and A. D. Dragan. Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots*, 42:1405–1426, 2018.

[7] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668. JMLR Workshop and Conference Proceedings, 2010.

[8] J. Spencer, S. Choudhury, A. Venkatraman, B. Ziebart, and J. A. Bagnell. Feedback in imitation learning: The three regimes of covariate shift. *arXiv preprint arXiv:2102.02872*, 2021.

[9] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer. Imitating driver behavior with generative adversarial networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE, 2017.

[10] R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn. Visual adversarial imitation learning using variational models. *Advances in Neural Information Processing Systems*, 34:3016–3028, 2021.

[11] N. Das, S. Bechtle, T. Davchev, D. Jayaraman, A. Rai, and F. Meier. Model-based inverse reinforcement learning from visual demonstrations. *arXiv preprint arXiv:2010.09034*, 2020.

[12] S. Yue, G. Wang, W. Shao, Z. Zhang, S. Lin, J. Ren, and J. Zhang. Clare: Conservative model-based reward learning for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.04782*, 2023.

[13] S. Zeng, C. Li, A. Garcia, and M. Hong. Understanding expertise through demonstrations: A maximum likelihood framework for offline inverse reinforcement learning. *arXiv preprint arXiv:2302.07457*, 2023.

[14] J. Chang, M. Uehara, D. Sreenivas, R. Kidambi, and W. Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34:965–979, 2021.

[15] C. Baker, R. Saxe, and J. Tenenbaum. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[16] D. Jarrett, A. Hüyük, and M. Van Der Schaar. Inverse decision modeling: Learning interpretable representations of behavior. In *International Conference on Machine Learning*, pages 4755–4771. PMLR, 2021.

[17] S. Reddy, A. D. Dragan, and S. Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. *arXiv preprint arXiv:1805.08010*, 2018.

[18] M. Herman, T. Gindele, J. Wagner, F. Schmitt, and W. Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial Intelligence and Statistics*, pages 102–110. PMLR, 2016.

[19] Z. Wu, P. Schrater, and X. Pitkow. Inverse rational control: Inferring what you think from how you forage. *arXiv preprint arXiv:1805.09864*, 2018.

[20] T. Makino and J. Takeuchi. Apprenticeship learning for model parameters of partially observable environments. *arXiv preprint arXiv:1206.6484*, 2012.

[21] F. Schmitt, H.-J. Bieg, M. Herman, and C. A. Rothkopf. I see what you see: Inferring sensor and policy models of human real-world motor behavior. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] Z. Gong and Y. Zhang. What is it you really want of me? generalized reward learning with biased beliefs about domain dynamics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2485–2492, 2020.

[23] G. Neu, A. Jonsson, and V. Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[25] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[26] A. Gleave and S. Toyer. A primer on maximum causal entropy inverse reinforcement learning. *arXiv preprint arXiv:2203.11409*, 2022.

[27] S. Zeng, M. Hong, and A. Garcia. Structural estimation of markov decision processes in high-dimensional state space with finite-time guarantees. *arXiv preprint arXiv:2210.01282*, 2022.

[28] S. Zeng, C. Li, A. Garcia, and M. Hong. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. *arXiv preprint arXiv:2210.01808*, 2022.

[29] S. K. S. Ghasemipour, R. Zemel, and S. Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pages 1259–1277. PMLR, 2020.

[30] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[31] L. Ke, S. Choudhury, M. Barnes, W. Sun, G. Lee, and S. Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 313–329. Springer, 2021.

[32] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. *arXiv preprint arXiv:1611.03852*, 2016.

[33] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58. PMLR, 2016.

[34] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[35] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[36] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[37] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[38] T. Jafferjee, E. Imani, E. Talvitie, M. White, and M. Bowling. Hallucinating value: A pitfall of dyna-style planning with imperfect environment models. *arXiv preprint arXiv:2006.04363*, 2020.

[39] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Y. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142, 2020.

[40] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020.

[41] C. Lu, P. Ball, J. Parker-Holder, M. Osborne, and S. J. Roberts. Revisiting design choices in offline model based reinforcement learning. In *International Conference on Learning Representations*, 2021.

[42] M. Uehara and W. Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

[43] A. Nilim and L. El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.

[44] G. N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

[45] M. Rigter, B. Lacerda, and N. Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *arXiv preprint arXiv:2204.12581*, 2022.

[46] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[47] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[48] S. Armstrong and S. Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. *Advances in neural information processing systems*, 31, 2018.

[49] R. Shah, N. Gundotra, P. Abbeel, and A. Dragan. On the feasibility of learning, rather than assuming, human biases for reward inference. In *International Conference on Machine Learning*, pages 5670–5679. PMLR, 2019.

[50] V. S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5): 291–294, 1997.

[51] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

[52] A. Vemula, Y. Song, A. Singh, J. A. Bagnell, and S. Choudhury. The virtues of laziness in model-based rl: A unified objective and algorithms. *arXiv preprint arXiv:2303.00694*, 2023.

[53] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[54] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[55] Y. Du, S. Li, J. Tenenbaum, and I. Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.

[56] K. Asadi, D. Misra, and M. Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

[57] D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

[58] J. Choi and K.-E. Kim. Map inference for bayesian inverse reinforcement learning. *Advances in neural information processing systems*, 24, 2011.

[59] A. J. Chan and M. van der Schaar. Scalable bayesian inverse reinforcement learning. *arXiv preprint arXiv:2102.06483*, 2021.

[60] M. Kwon, S. Daptardar, P. R. Schrater, and X. Pitkow. Inverse rational control with partially observable continuous nonlinear dynamics. *Advances in neural information processing systems*, 33:7898–7909, 2020.

[61] N. Lambert, B. Amos, O. Yadan, and R. Calandra. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

[62] C. Grimm, A. Barreto, S. Singh, and D. Silver. The value equivalence principle for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:5541–5552, 2020.

[63] A.-m. Farahmand, A. Barreto, and D. Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

[64] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.