Robust Inverse Reinforcement Learning Through Bayesian Theory of Mind

Ran Wei [1] **Siliang Zeng** [2] **Chenliang Li** [1] **Alfredo Garcia** [1] **Anthony McDonald** [3] **Mingyi Hong** [2]

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 parametrize 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 (expert) 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.

1. Introduction

Inverse reinforcement learning (IRL) is the problem of extracting the reward function and policy of a value-maximizing agent from its behavior (Ng et al., 2000; Osa et al., 2018). IRL is an important tool in domains where manually specifying reward functions or policies is difficult, such as in autonomous driving (Phan-Minh et al., 2022), or when the extracted reward function can reveal novel insight about a target population, such as in biology and economics (Yamaguchi et al., 2018; Rust, 1987). Furthermore, IRL has been argued as a central mechanism of human theory of mind (Jara-Ettinger, 2019) and one of the main approaches for building value-aligned artificial intelligence (Russell, 2019). 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 (Ross & Bagnell, 2010; Spencer et al., 2021; Kuefler et al., 2017).

To tackle the first challenge, recent IRL research has focused on the *offline* setting, where only a fixed dataset is provided as opposed to the target environment or an accurate simulator (Chan & van der Schaar, 2021; Garg et al., 2021; Kostrikov et al., 2019; Rafailov et al., 2021; Das et al., 2020). Model-free approaches to offline IRL attempt to directly estimate expert reward and policy without building an explicit model of the environment dynamics (Chan & van der Schaar, 2021; Garg et al., 2021; Kostrikov et al., 2019). In contrast, model-based offline IRL approaches estimate a dynamics model from the offline dataset (Das et al., 2020; Rafailov et al., 2021; Yue et al., 2023; Zeng et al., 2023). Both model-free and model-based offline IRL suffer from covariate shift due to error in either the policy or the dynamics model. However, model-based approaches, which will be our focus, hold more promise due to the ability to generate synthetic data and leverage model generalization.

A notable class of these model-based offline IRL methods estimate the dynamics and reward in a two-stage, *decoupled* fashion (Rafailov et al., 2021; Das et al., 2020; Yue et al., 2023; Zeng et al., 2023). 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) (Chang et al., 2021; Zeng et al., 2023; Yue et al., 2023).

In this paper, we instead approach IRL from the Bayesian

[*]Equal contribution [1]Department of Industrial and Systems Engineering, Texas A&M University, USA [2]Department of Electrical and Computer Engineering, University of Minnesota, USA [3]Department of Industrial and Systems Engineering, University of Wisconsin, USA. Correspondence to: Ran Wei <rw422@tamu.edu>, Siliang Zeng <zeng0176@umn.edu>, Chenliang li <chenliangli@tamu.edu>, Alfredo Garcia <alfredo.garcia@tamu.edu>, Anthony McDonald <admcdonald@wisc.edu>, Mingyi Hong <mhong@umn.edu>.

Proceedings of the First Workshop on Theory of Mind in Communicating Agents, Honolulu, Hawaii, USA. PMLR 202, 2023.

Theory of Mind perspective (Baker et al., 2011), 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 (Baker et al., 2011) and thus should affect the update direction of the dynamics model as opposed to it being fixed. BTOM has mostly been used to understand human biases encoded in the internal dynamics in simple and highly constrained domains (Jarrett et al., 2021; Reddy et al., 2018; Herman et al., 2016; Wu et al., 2018; Makino & Takeuchi, 2012; Schmitt et al., 2017; Gong & Zhang, 2020). In contrast to these works, we study how BTOM naturally enables learning high-performance and robust policies given a limited dataset.

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 a set of algorithms which extend prior simultaneous estimation approaches (Herman et al., 2016; Wu et al., 2018) to high-dimensional continuous-control settings. We show that the proposed algorithms outperform state-of-the-art (SOTA) offline IRL methods without the need for designing pessimistic penalties.

In summary, our contributions are the following:

- We show that BTOM under appropriate formulation of the prior is robust to inaccuracies in the estimated dynamics model.
- We propose a set of practical algorithms for simultaneous estimation of reward and dynamics in high-dimensional environments.
- We perform extensive experiments in the MuJoCo environment to confirm our analysis and show that the proposed algorithms outperform pessimistic approaches.

2. Preliminaries

2.1. Markov Decision Process

We consider modeling agent behavior using infinite-horizon *entropy-regularized* Markov decision processes (MDP; Neu et al., 2017) defined by tuple $(\mathcal{S}, \mathcal{A}, \mu, P, \gamma, R)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, initial state distribution $\mu(s_0) \in \Delta(\mathcal{S})$, transition probability distribution $P(s'|s,a) \in \Delta(\mathcal{S})$, discount factor $\gamma \in (0,1)$, and reward function $R(s,a) \in \mathbb{R}$. We denote the discounted occupancy measure as $\rho^\pi_P(s,a) = \mathbb{E}_{\mu,P,\pi}\left[\sum_{t=0}^\infty \gamma^t P(s_t = s, a_t = a)\right]$ and the marginal state-action distribution as $d^\pi_P(s,a) = (1-\gamma)\rho^\pi_P(s,a)$. We further denote the discounted occupancy measure starting from a specific state-action pair (s,a) with $\rho^\pi_P(\tilde{s},\tilde{a}|s,a)$. 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:

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

The optimal policy satisfies the following conditions (i.e., Boltzmann rationality; Haarnoja et al., 2018a):

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

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 (Ziebart, 2010).

A related formulation casts IRL as maximum *discounted* likelihood (ML) estimation (Gleave & Toyer, 2022; Zeng et al., 2022a;b), 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 $\rho_\mathcal{D} := \rho^\pi_P$, ML-IRL aims to solve the following optimization problem:

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

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 (Gleave & Toyer, 2022; Zeng et al., 2022a), however (3) permits non-linear reward parameterization through the following surrogate optimization problem:

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

(4) can be efficiently solved via alternating training of the learner policy and the reward function, similar to Generative Adversarial Network (GAN)-based algorithms (Ghasemipour et al., 2020; Ho & Ermon, 2016; Ke et al., 2021; Finn et al., 2016a;b; Fu et al., 2017). 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^{π}_{P}.

2.3. Offline Model-Based IRL & RL

Existing offline model-based IRL algorithms such as Rafailov et al. (2021) and Das et al. (2020) 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 $\rho^{\pi}_{\hat{P}}$ 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 (Zeng et al., 2023). 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 (Levine et al., 2020; Chua et al., 2018; Janner et al., 2019; Jafferjee et al., 2020). 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 (Levine et al., 2020). These pessimistic penalties are often designed based on quantifying uncertainty about transition dynamics through the estimated model (Yu et al., 2020; Kidambi et al., 2020). Drawing on these advances, recent offline IRL methods also incorporate pessimistic penalties into their RL subroutine (Zeng et al., 2023; Yue et al., 2023; Chang et al., 2021). However, it should be noted that designing pessimistic penalties involves nontrivial decisions to ensure that they can accurately capture out-of-distribution samples (Lu et al., 2021).

An orthogonal approach to avoid model-exploitation is to perform policy training against the worst-case dynamics in out-of-distribution states (Uehara & Sun, 2021), similar to robust MDP (Nilim & El Ghaoui, 2005; Iyengar, 2005). Rigter et al. (2022) 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.

3. Bayesian Theory of Mind

We consider IRL under the Bayesian Theory of Mind framework, where the observed expert decisions are the results of an unknown reward function $R_{\theta_1}(s,a)$ and their *internal* model of the environment dynamics $\hat{P}_{\theta_2}(s'|s,a)$. 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. We make no additional assumption about the expert other than that their policy is Boltzmann rational (2) with respect to their internal reward and dynamics. This means that their internal dynamics can potentially deviate from the true environment dynamics.

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 $\mathbb{P}(\theta)$:

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

where we have omitted the true environment transition probabilities $\prod_{i=1}^{N}\prod_{t=1}^{T} P(s_{i,t+1}|s_{i,t},a_{i,t})$ from the likelihood because they do 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 the expert's model of the environment is.

Let $\mathcal{L}(\theta) := \frac{1}{NT}\log \mathbb{P}(\theta|\mathcal{D})$ be the log-posterior (normalized by the data size). 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:

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

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; Kingma & Welling, 2013; Welling & Teh, 2011) 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 proposed by Herman et al. (2016) and Wu et al. (2018) 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 A for all proofs and derivations):

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

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 $\tilde{L}(\theta)$ with the same gradient as the original MAP estimation problem (7):

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

where

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

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

Optimizing (11) is now the same as optimizing (7) but tractable.

An interesting consequence of maximizing the first line of (11) alone 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 (Armstrong & Mindermann, 2018; Reddy et al., 2018; Shah et al., 2019). 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 the IRL learner exhibits robust performance as a natural consequence of 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 it 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 (Zeng et al., 2023) as follows:

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

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 under the *expert* data distribution:

Lemma 3.1. *Let* $R_{max} = \max_{s,a} |R_\theta(s,a)| + \log |\mathcal{A}|$ *and* $\epsilon = \mathbb{E}_{(s,a)\sim P(\tau)} D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a))$*, it holds that*

$$|\textbf{\textit{T1}}| \leq \frac{\gamma R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon} \tag{15}$$

Thus, if $\mathbb{E}_{(s,a)\sim P(\tau)} D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a)) \leq \epsilon$ holds for sufficiently small ϵ, for example by setting a large λ, **T1** can be dropped from (14) and the discounted likelihood reduces to $\ell(\theta)$.

$\ell(\theta)$ highlights the reason why the proposed BTOM approach can be robust to a limited dataset. It poses the offline

Algorithm 1 Deep Bayesian Theory of Mind (BTOM)

Require: Dataset $\mathcal{D} = \{\tau\}$, dynamics model $\hat{P}_{\theta_2}(s'|s,a)$, reward model $R_{\theta_1}(s,a)$, hyperparameters λ_1, λ_2
1: **for** $k = 1 : K$ **do**
2: Run MBPO to update learner policy $\hat{\pi}(a|s;\theta)$ and value function $Q_\theta(s,a)$ in dynamics $\hat{P}$
3: Sample real trajectory τ_{real} starting from $(s,a) \sim \mathcal{D}$ and following $\hat{P}$ and $\hat{\pi}$
4: Sample fake trajectory τ_{fake} starting from $s \sim \mathcal{D}$, $a_{\text{fake}} \sim \hat{\pi}(\cdot|s;\theta)$ and following $\hat{P}$ and $\hat{\pi}$
5: Evaluate (16) and take a gradient step
6: Evaluate (17) and take a few gradient steps.
7: **end for**

Algorithm 2 Robust Theory of Mind (RTOM)

Require: Dataset $\mathcal{D} = \{\tau\}$, dynamics model $\hat{P}_{\theta_2}(s'|s,a)$, reward model $R_{\theta_1}(s,a)$, hyperparameters λ_1, λ_2
1: **for** $k = 1 : K$ **do**
2: Run MBPO to update learner policy $\hat{\pi}(a|s;\theta)$ and value function $Q_\theta(s,a)$ in dynamics $\hat{P}$
3: Sample fake trajectory τ_{fake} starting from $s \sim \mathcal{D}$ and following $\hat{P}$ and $\hat{\pi}$
4: Evaluate (19) and take a gradient step
5: Evaluate (20) and take a few gradient steps
6: **end for**

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, pessimistic* dynamics, which is trained adversarially outside the data distribution. This connects BTOM to the robust MDP approach to offline model-based RL (Uehara & Sun, 2021; Rigter et al., 2022).

3.3. Proposed Algorithms

Using the insights from the previous sections, we propose two scalable Deep Bayesian Theory of Mind algorithms to find the MAP solution to (7). The first algorithm (**BTOM**; 1) applies the naive solution with surrogate objective (11), while the second algorithm (**RTOM**; 2) exploits the observation in section 3.2 to derive a more efficient algorithm for high λ via surrogate objective $\ell(\theta)$.

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 (Zeng et al., 2022a; 2023), we perform the nested optimization using *two-timescale* stochastic approximation (Borkar, 1997; Hong et al., 2020), 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; Janner et al., 2019) which uses Soft Actor-Critic (SAC; Haarnoja et al., 2018a) 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:

$$\max_{\theta_1} \quad \mathbb{E}_{(s,a)\sim\mathcal{D},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[R_{\theta_1}(\tilde{s},\tilde{a})\right] - \mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[R_{\theta_1}(\tilde{s},\tilde{a})\right] \tag{16}$$

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 (Rigter et al., 2022):

$$\begin{aligned}\max_{\theta_2} \quad & \lambda_1\mathbb{E}_{(s,a)\sim\mathcal{D},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] \\ & - \lambda_1\mathbb{E}_{s\sim\mathcal{D},a_{\text{fake}}\sim\hat{\pi},\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] \\ & + \lambda_2\mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log\hat{P}_{\theta_2}(s'|s,a)\right]\end{aligned} \tag{17}$$

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:

$$\begin{aligned}&\nabla_{\theta_2} EV_\theta(s,a) \\ &= \sum_{s'} V_\theta(s')\nabla_{\theta_2}\hat{P}_{\theta_2}(s'|s,a) \\ &= \mathbb{E}_{s'\sim\hat{P}(\cdot|s,a)}\left[\left(V_\theta(s') - b(s,a)\right)\nabla_{\theta_2}\log\hat{P}_{\theta_2}(s'|s,a)\right]\end{aligned} \tag{18}$$

Following Rigter et al. (2022), we set the baseline to $b(s,a) = Q_\theta(s,a) - R_{\theta_1}(s,a)$ to reduce gradient variance and further normalize $V_\theta(s') - b(s,a)$ across the mini-batch to stabilize training. In the continuous-control setting, the value function can be estimated as $V_\theta(s) = \mathbb{E}_{a\sim\hat{\pi}_\theta}[Q_\theta(s,a) - \log\hat{\pi}(a|s;\theta)]$ with a single sample.

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:

$$\max_{\theta_1} \mathbb{E}_{\rho_{\mathcal{D}}}\left[R_{\theta_1}(s,a)\right] - \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}}\left[R_{\theta_1}(s,a)\right] \tag{19}$$

We then update the dynamics by dropping the first term in (17):

$$\begin{aligned}\max_{\theta_2} \quad & -\lambda_1 \mathbb{E}_{s\sim\mathcal{D}, a_{\text{fake}}\sim\hat{\pi}, \rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a_{\text{fake}})}\left[EV_{\theta_2}(\tilde{s},\tilde{a})\right] \\ & + \lambda_2 \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\log \hat{P}_{\theta_2}(s'|s,a)\right]\end{aligned} \tag{20}$$

We provide additional details about the proposed algorithms in Appendix B.

3.4. Performance Guarantees

In this section, we study how policy and dynamics estimation error affect learner performance in the real environment. Vemula et al. (2023) provided the following result relating expert-learner performance gap in the real and estimated environment in the context of model-based RL:

Lemma 3.2. *(Performance difference via advantage in model; Lemma 4.1 in (Vemula et al., 2023)) Let d^{π}_{P} denote the marginal state-action distribution following policy π in environment P. The following relationship holds:*

$$\mathbb{E}_{(s,a)\sim d^{\pi}_{P}}\left[\log \hat{\pi}_{\hat{P}}(a|s)\right] \tag{21}$$

$$= \mathbb{E}_{s\sim d^{\pi}_{P}}\left[\mathbb{E}_{a\sim\pi} Q^{\hat{\pi}}_{\hat{P}}(s,a) - V^{\hat{\pi}}_{\hat{P}}(s)\right] \tag{22}$$

$$= \underbrace{(1-\gamma)\mathbb{E}_{s\sim\mu}\left[V^{\pi}_{P}(s) - V^{\hat{\pi}}_{P}(s)\right]}_{\textit{Performance difference in real environment}} \tag{23}$$

$$+ \underbrace{\gamma \mathbb{E}_{(s,a)\sim d^{\hat{\pi}}_{P}}\left[\mathbb{E}_{s'\sim P} V^{\hat{\pi}}_{\hat{P}}(s') - \mathbb{E}_{s''\sim\hat{P}} V^{\hat{\pi}}_{\hat{P}}(s'')\right]}_{\textit{Model (dis)advantage under learner distribution}} \tag{24}$$

$$+ \underbrace{\gamma \mathbb{E}_{(s,a)\sim d^{\pi}_{P}}\left[\mathbb{E}_{s'\sim\hat{P}} V^{\hat{\pi}}_{\hat{P}}(s') - \mathbb{E}_{s''\sim P} V^{\hat{\pi}}_{\hat{P}}(s'')\right]}_{\textit{Model advantage under expert distribution}} \tag{25}$$

Intuitively, maximizing the policy likelihood (21) w.r.t. $\hat{P}$ (including the reward) increases the performance gap (23) between the expert and the learner, increases model advantage under the expert data distribution, and decreases model advantage under the (unknown) learner data distribution. The performance gap is then to be closed by the learner during the inner optimization problem.

Using this result, we arrive at the follow performance bound:

Theorem 3.3. *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. 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:*

$$|J_P(\hat{\pi}) - J_P(\pi)| \leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon_{\hat{P}}} \tag{26}$$

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 BTOM 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 transition likelihood penalty λ of 0.001, 0.5, and 10, respectively. 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 (9).

Given that the environment is simple and both the policy, reward, and dynamics models are well-specified, all agents

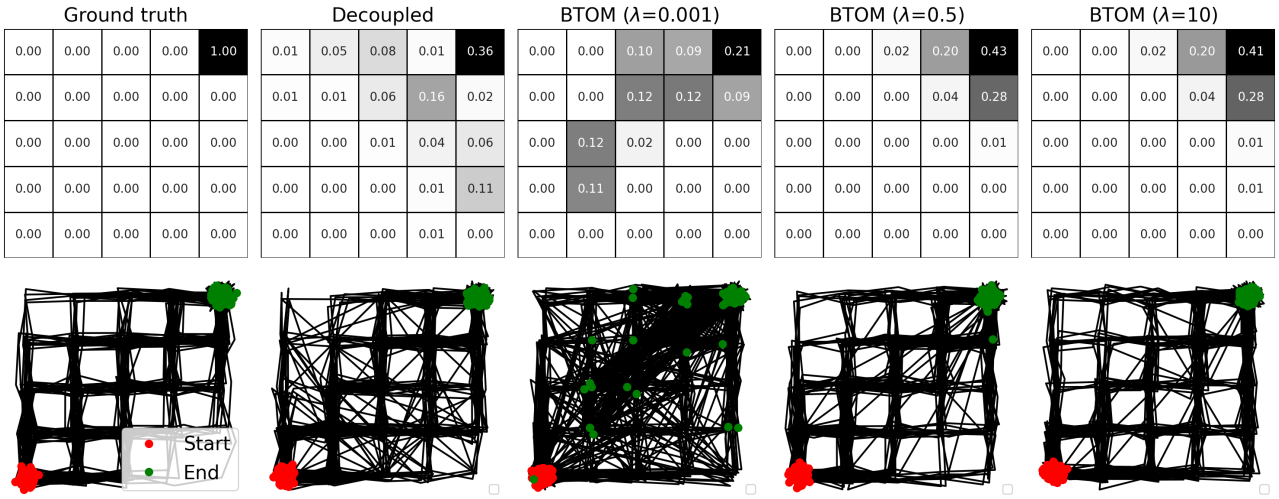

Figure 1. Gridworld experiment results. **(Row 1)** Ground truth and estimated target state distributions (softmax of reward) for agent using decoupled estimation and BTOM agents with $\lambda = [0.001, 0.5, 10]$. BTOM agents with higher λ obtain more accurate reward estimates. **(Row 2)** Sample paths 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

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.

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 agents 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 optimizes model advantage 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 (Todorov et al., 2012) using the D4RL dataset (Fu et al., 2020). We use ML-IRL (Zeng et al., 2023), an offline model-based IRL algorithm based on MOPO (Yu et al., 2020), 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 train 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 B.

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 optimizing only $\ell(\theta)$ in (14) as **T1** would be large in this case. 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.

Finally, we remark that BTOM has 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 (Asadi et al., 2018).

5. Related Work and Discussions

Bayesian IRL. Ramachandran & Amir (2007) first proposed a Bayesian formulation of IRL to solve the reward ambiguity problem. A MAP inference approach was proposed in (Choi & Kim, 2011) and a variational inference approach was proposed in (Chan & van der Schaar, 2021). Their formulations consider non-entropy-regularized policies and the dynamics model is fixed during reward inference. In contrast, simultaneous estimation of reward and dynamics can potentially infer the demonstrator's biased beliefs about the environment, which is desirable for psychology and human-robot interaction studies (Baker et al., 2011; Wu et al., 2018; Reddy et al., 2018). Despite the attractiveness, simultaneous estimation is challenging because of the need to invert the agent's planning process, especially in continuous domains. Reddy et al. (2018) avoids this by representing agent discrete choice policies using neural network-parameterized Q functions and regularizing the Bellman error to be small over the entire state-action space. This method however cannot be straightforwardly adapted to the continuous action case. Kwon et al. (2020) avoids this by first training a task-conditioned policy on a distribution of environments with known parameters using meta reinforcement learning and then use the meta-trained policy to guide inference. This precludes the method from being used in general settings with unknown task distributions. To our knowledge, our proposed algorithms are the first to address simultaneous estimation in general environments.

Decision-aware model learning. Decision-aware model learning aims to solve the objective mismatch problem in model-based RL (Lambert et al., 2020). Many proposed methods in this class use value-targeted regression similar to our model loss in (17) (Grimm et al., 2020; Farahmand et al., 2017). Our analysis and that of Vemula et al. (2023) suggest that value-targeted model objectives may be related to robust objectives. Furthermore, since the set of value-equivalent models only shrink for an increasingly larger set of policies and values (Grimm et al., 2020), using value-aware model objectives alone may not be optimal and additional prediction-based regularizations may be needed.

Theory of Mind. Theory of Mind inference is known to be unidentifiable in general. Many researchers believe that reliable inference in human theory of mind relies on highly structured priors and normative assumptions (Jara-Ettinger, 2019; Armstrong & Mindermann, 2018; Langley et al., 2022). We took a small step in understanding the relationship between a type of structured prior, i.e., the dynamics accuracy prior (6), and the inference outcome. Different from prior works which also use accuracy-based regularizations but assume known ground truth dynamics (Reddy et al., 2018; Shah et al., 2019), our prior is more general and flexible since it is estimated partially from data. While our goal in this work has been to understand BTOM inference of expert demonstrators, an interesting future direction is to identify appropriate priors to reliably infer reward and internal dynamics from sub-optimal and biased human

demonstrators.

Our observation of the robustness of BTOM also has interesting cognitive science implications. It suggests that inference of (Boltzmann) rational agents naturally gives rise to a form of "pessimism in the face of uncertainty", which provides a testable hypothesis of Boltzmann rationality as a model of human theory of mind. Furthermore, this knowledge can potentially be applied in machine teaching and multi-agent coordination settings to design more efficient and human-like communicative actions (Ho et al., 2016; Foerster et al., 2019; Mirsky et al., 2022).

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.

Acknowledgements

AG would like to acknowledge partial support from the Army Research Office grant W911NF-22-1-0213. RW would like to acknowledge Marc Rigter for answering questions about the RAMBO algorithm.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

Foerster, J., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., and Bowling, M. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1942–1951. PMLR, 2019.

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

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

Garg, D., Chakraborty, S., Cundy, C., Song, J., and Ermon, S. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34: 4028–4039, 2021.

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

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

Gong, Z. and Zhang, Y. 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, pp. 2485–2492, 2020.

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

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

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

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

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

Ho, M. K., Littman, M., MacGlashan, J., Cushman, F., and Austerweil, J. L. Showing versus doing: Teaching by demonstration. *Advances in neural information processing systems*, 29, 2016.

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

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

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

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

Jara-Ettinger, J. Theory of mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29: 105–110, 2019.

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

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

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

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

Kostrikov, I., Nachum, O., and Tompson, J. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*, 2019.

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

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

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

Langley, C., Cirstea, B. I., Cuzzolin, F., and Sahakian, B. J. Theory of mind and preference learning at the interface of cognitive science, neuroscience, and ai: A review. *Frontiers in Artificial Intelligence*, pp. 62, 2022.

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

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

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

Mirsky, R., Carlucho, I., Rahman, A., Fosong, E., Macke, W., Sridharan, M., Stone, P., and Albrecht, S. V. A survey of ad hoc teamwork research. In *Multi-Agent Systems: 19th European Conference, EUMAS 2022, Düsseldorf, Germany, September 14–16, 2022, Proceedings*, pp. 275–293. Springer, 2022.

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

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

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

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

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

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

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

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

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

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

Russell, S. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

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

Schmitt, F., Bieg, H.-J., Herman, M., and Rothkopf, C. A. 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.

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

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

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

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

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

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

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

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

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

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

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

Zeng, S., Li, C., Garcia, A., and Hong, M. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 35:10122–10135, 2022b.

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

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

A. Proofs for section 3

Derivation of BTOM Gradients (section 3.1). Recall the definition of the optimal entropy-regularized policy and value functions:

$$\begin{aligned}\hat{\pi}(a|s;\theta) &= \frac{\exp(Q_\theta(s,a))}{\sum_{\tilde{a}}\exp(Q_\theta(s,\tilde{a}))}\\ Q_\theta(s,a) &= R_{\theta_1}(s,a) + \gamma\mathbb{E}_{\hat{P}_{\theta_2}(\cdot|s,a)}[V_\theta(s')]\\ V_\theta(s) &= \log\sum_{\tilde{a}}\exp(Q_\theta(s,\tilde{a}))\end{aligned} \tag{27}$$

The gradient of the policy log likelihood in terms of the Q function gradient is obtained as follow:

$$\begin{aligned}\nabla_\theta \log\hat{\pi}(a|s;\theta) &= \nabla_\theta Q_\theta(s,a) - \nabla_\theta V_\theta(s)\\ &= \nabla_\theta Q_\theta(s,a) - \frac{1}{Z_\theta}\nabla_\theta\sum_{\tilde{a}}\exp(Q_\theta(s,\tilde{a}))\\ &= \nabla_\theta Q_\theta(s,a) - \frac{1}{Z_\theta}\sum_{\tilde{a}}\exp(\nabla_\theta Q_\theta(s,\tilde{a}))\\ &= \nabla_\theta Q_\theta(s,a) - \mathbb{E}_{\tilde{a}\sim\hat{\pi}}[\nabla_\theta Q_\theta(s,\tilde{a})]\end{aligned} \tag{28}$$

where $Z_\theta = \sum_{a'}\exp(Q_\theta(s,a'))$ is the normalizer.

Recall $\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)$ is the discounted state-action occupancy measure starting from pair (s,a). We define for any function $f(s,a)$:

$$\mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}[f(s,a)] = \mathbb{E}_{\tau\sim(\hat{P},\hat{\pi})}\left[\sum_{t=0}^{\infty}\gamma^t f(s,a)\middle| s_0=s, a_0=a\right] \tag{29}$$

We now derive Q function gradients with respect to the reward parameters θ_1 and dynamics parameters θ_2, respectively.

$$\begin{aligned}\nabla_{\theta_1}Q_\theta(s,a) &= \nabla_{\theta_1}R_{\theta_1}(s,a) + \gamma\mathbb{E}_{s'\sim\hat{P}_{\theta_2}(\cdot|s,a)}[\nabla_{\theta_1}V_\theta(s')]\\ &= \nabla_{\theta_1}R_{\theta_1}(s,a) + \gamma\mathbb{E}_{s'\sim\hat{P}_{\theta_2}(\cdot|s,a),a'\sim\hat{\pi}(\cdot|s';\theta)}[\nabla_{\theta_1}Q_\theta(s',a')]\\ &= \nabla_{\theta_1}R_{\theta_1}(s,a) + \gamma\mathbb{E}_{s'\sim\hat{P}_{\theta_2}(\cdot|s,a),a'\sim\hat{\pi}(\cdot|s';\theta)}\Big[\\ &\quad \nabla_{\theta_1}R_{\theta_1}(s',a') + \gamma\mathbb{E}_{s''\sim\hat{P}_{\theta_2}(\cdot|s',a'),a''\sim\hat{\pi}(\cdot|s'';\theta)}[\nabla_{\theta_1}Q_\theta(s'',a'')]\Big]\\ &= \nabla_{\theta_1}R_{\theta_1}(s,a) + \mathbb{E}_{\tau\sim(\hat{P},\hat{\pi})}\left[\sum_{h=1}^{\infty}\gamma^h\nabla_{\theta_1}R_{\theta_1}(s_h,a_h)\middle| s_0=s, a_0=a\right]\\ &= \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)}\left[\nabla_{\theta_1}R_{\theta_1}(\tilde{s},\tilde{a})\right]\end{aligned} \tag{30}$$

In line two we used the result that $\nabla_\phi V_\phi(s)$ for both $\phi=\theta_1$ and $\phi=\theta_2$ corresponds to the second term in (28) .

$$
\begin{aligned}
\nabla_{\theta_2} Q_\theta(s,a) &= \nabla_{\theta_2} R_{\theta_1}(s,a) + \nabla_{\theta_2} \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a)}[V_\theta(s')] \\
&= \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a), a' \sim \hat{\pi}(\cdot|s';\theta)}[\nabla_{\theta_2} Q_\theta(s',a')] \\
&= \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s,a) + \gamma \mathbb{E}_{s' \sim \hat{P}_{\theta_2}(\cdot|s,a), a' \sim \hat{\pi}(\cdot|s';\theta)} \Bigg[\\
&\quad \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s',a') + \gamma \mathbb{E}_{s'' \sim \hat{P}_{\theta_2}(\cdot|s',a'), a'' \sim \hat{\pi}(\cdot|s'';\theta)}[\nabla_{\theta_2} Q_\theta(s'',a'')] \Bigg] \\
&= \gamma \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s,a) + \mathbb{E}_{\tau \sim (\hat{P},\hat{\pi})} \left[\sum_{h=1}^{\infty} \gamma^{h+1} \sum_{\tilde{s}} V_\theta(\tilde{s}) \nabla_{\theta_2} \hat{P}_{\theta_2}(\tilde{s}|s_h,a_h) \middle| s_0 = s, a_0 = a \right] \\
&= \mathbb{E}_{\rho^{\hat{\pi}}_{\hat{P}}(\tilde{s},\tilde{a}|s,a)} \left[\gamma \sum_{s'} V_\theta(s') \nabla_{\theta_2} \hat{P}_{\theta_2}(s'|\tilde{s},\tilde{a}) \right]
\end{aligned}
\tag{31}
$$

We make a quick remark on the identifiability of simultaneous estimation.

Remark A.1. Simultaneous reward-dynamics estimation of the form (5) without specific assumptions on the prior $P(\theta)$ is in general unidentifiable.

Proof. Let $\mathbf{R} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $\mathbf{P} \in \mathbb{R}_+^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}, \sum_{s'} \mathbf{P}^a_{ss'} = 1$, $\mathbf{Q} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $\mathbf{V} \in \mathbb{R}^{|\mathcal{S}|}$ be a set of Bellman-consistent reward, dynamics, and value functions in matrix form. Let $\mathbf{P}' \neq \mathbf{P}$ be an alternative dynamics model. We can always find an alternative reward $\mathbf{R}' = \mathbf{R} + \Delta\mathbf{R}$, where:

$$
\begin{aligned}
\Delta\mathbf{R} &= (\mathbf{Q} - \mathbf{Q}) - \gamma(\mathbf{P}'\mathbf{V} - \mathbf{P}\mathbf{V}) \\
&= -\gamma \Delta\mathbf{P}\mathbf{V}
\end{aligned}
\tag{32}
$$

without changing the value functions and optimal entropy-regularized policy. □

Remark A.1 implies that existing simultaneous estimation approaches which do not use explicit or implicit regularizations, such as the SERD algorithm by (Herman et al., 2016), cannot in general accurately estimate expert reward. Paired with theorem 3.3, it shows that these algorithms cannot in general achieve good performance.

Lemma A.2. *(Restate of lemma 3.1) Let* $R_{max} = \max_{s,a} |R_\theta(s,a)| + \log|\mathcal{A}|$ *and* $\epsilon = \mathbb{E}_{(s,a)\sim P(\tau)} D_{KL}(P(\cdot|s,a)||\hat{P}(\cdot|s,a))$*, it holds that*

$$
|\boldsymbol{T1}| \leq \frac{\gamma R_{max}}{(1-\gamma)^2} \sqrt{2\epsilon}
\tag{33}
$$

Proof.

$$
\begin{aligned}
|\mathbf{T1}| &= \left| \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E}_{(s_t,a_t)\sim P(\tau)} \left[\sum_{s'} V_\theta(s') \left(\hat{P}(s'|s_t,a_t) - P(s'|s_t,a_t) \right) \right] \right| \\
&\overset{(1)}{\leq} \sum_{t=0}^{\infty} \gamma^{t+1} \mathbb{E}_{(s_t,a_t)\sim P(\tau)} \left[\sum_{s'} |V_\theta(s')| \left| \hat{P}(s'|s_t,a_t) - P(s'|s_t,a_t) \right| \right] \\
&\overset{(2)}{\leq} \sum_{t=0}^{\infty} \gamma^{t+1} \|V_\theta(\cdot)\|_\infty \mathbb{E}_{(s_t,a_t)\sim P(\tau)} \left[\left\| \hat{P}(\cdot|s_t,a_t) - P(\cdot|s_t,a_t) \right\|_1 \right] \\
&\overset{(3)}{\leq} \sum_{t=0}^{\infty} \gamma^{t+1} \|V_\theta(\cdot)\|_\infty \sqrt{2 \mathbb{E}_{(s_t,a_t)\sim P(\tau)} D_{KL}(P||\hat{P})} \\
&= \frac{\gamma}{1-\gamma} \|V_\theta(\cdot)\|_\infty \sqrt{2\epsilon}
\end{aligned}
$$

where (1) follows from Jensen's inequality, (2) follows from Holder's inequality, and (3) follows from Pinsker's inequality. Finally, given $\mathcal{H}(\pi(a|s)) = -\sum_a \pi(a|s)\log\pi(a|s) \leq -\sum_a \pi(a|s)\log\frac{1}{|\mathcal{A}|} = \log|\mathcal{A}|$, we have $\|V_\theta(\cdot)\|_\infty \leq \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\left(\max_{s,a}|R_\theta(s,a)| + \log|\mathcal{A}|\right)\right] = \frac{R_{\max}}{1-\gamma}$.

□

Theorem A.3. *(Restate of theorem 3.3) Let $\epsilon_{\hat{\pi}} = -\mathbb{E}_{(s,a)\sim d_P^{\pi}}[\log\hat{\pi}_{\hat{P}}(a|s)]$ be the policy estimation error and $\epsilon_{\hat{P}} = \mathbb{E}_{(s,a)\sim d_P^{\pi}} D_{KL}[P(\cdot|s,a)||\hat{P}(\cdot|s,a)]$ be the dynamics estimation error. Assuming bounded expert-learner marginal state-action density ratio $\left\|\frac{d_P^{\hat{\pi}}(s,a)}{d_P^{\pi}(s,a)}\right\|_\infty \leq C$, we have the following (absolute) performance bound for the IRL agent:*

$$|J_P(\hat{\pi}) - J_P(\pi)| \leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{max}}{(1-\gamma)^2}\sqrt{2\epsilon_{\hat{P}}} \tag{34}$$

Proof.

$$\begin{aligned}
&|J_P(\hat{\pi}) - J_P(\pi)| \\
&\leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} \\
&\quad + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left|\frac{d_P^{\hat{\pi}}(s,a)}{d_P^{\pi}(s,a)}\left(\mathbb{E}_{s'\sim P}V_{\hat{P}}^{\hat{\pi}}(s') - \mathbb{E}_{s''\sim\hat{P}}V_{\hat{P}}^{\hat{\pi}}(s'')\right)\right|\right] \\
&\quad + \frac{\gamma}{1-\gamma}\mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left|\mathbb{E}_{s'\sim\hat{P}}V_{\hat{P}}^{\hat{\pi}}(s') - \mathbb{E}_{s''\sim P}V_{\hat{P}}^{\hat{\pi}}(s'')\right|\right] \\
&\leq \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} \\
&\quad + \frac{\gamma}{1-\gamma}\left\|\frac{d_P^{\hat{\pi}}(\cdot,\cdot)}{d_P^{\pi}(\cdot,\cdot)}\right\|_\infty \left\|V_{\hat{P}}^{\hat{\pi}}(\cdot)\right\|_\infty \mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\right\|_1\right] \\
&\quad + \frac{\gamma}{1-\gamma}\left\|V_{\hat{P}}^{\hat{\pi}}(\cdot)\right\|_\infty \mathbb{E}_{(s,a)\sim d_P^{\pi}}\left[\left\|\hat{P}(\cdot|s,a) - P(\cdot|s,a)\right\|_1\right] \\
&= \frac{1}{1-\gamma}\epsilon_{\hat{\pi}} + \frac{\gamma(C+1)R_{\max}}{(1-\gamma)^2}\sqrt{2\epsilon_{\hat{P}}}
\end{aligned} \tag{35}$$

where the last line uses results from lemma 3.1. □

B. Implementation Details

Our implementation builds on top of the official RAMBO implementation[1] (Rigter et al., 2022).

B.1. MuJoCo Benchmarks

For the MuJoCo benchmarks described in section 4.2, we follow standard practices in model-based RL.

B.1.1. Dynamics Pre-training

We use an ensemble of $K = 7$ neural networks where each network outputs the mean and covariance parameters of a Gaussian distribution over the difference between the next state and the current state $\delta = s' - s$:

$$\hat{P}_{\theta_2}^{(k)}(\delta|s,a) = \mathcal{N}(\delta|\mu_{\theta_2}^{(k)}(s,a), \Sigma_{\theta_2}^{(k)}(s,a)) \tag{36}$$

Each network is a 4-layer feedforward network with 200 hidden units and Sigmoid linear unit (SiLU) activation function. For the initial pre-training step, we maximize the likelihood of dataset transitions using a batch size of 256 and early stop when all models stop improving for more than 1 percent. We then select the 5 best models in terms of mean-squared-error on a 10 % holdout validation set. During model rollouts, we randomly pick one of the 5 best models (elites) to sample the next state.

[1] https://github.com/marc-rigter/rambo

Table 2. Shared hyperparameters across different environments

	Hyparameter	BTOM	RTOM
SAC + MBPO	critic learning rate	3e-4	3e-4
	actor learning rate	3e-4	3e-4
	discount factor (γ)	0.99	0.99
	soft target update parameter (τ)	5e-3	5e-3
	target entropy	-dim(A)	-dim(A)
	minimum temperature (α)	0.1	0.001
	batch size	256	256
	real ratio	0.5	0.5
	model retain epochs	5	5
	training epochs	500	300
	steps per epoch	1000	1000
Dynamics	# model networks	7	7
	# elites	5	5
	adv. rollout batch size	1000	256
	adv. rollout steps	10	10
	adv. update steps	50	50
	adv. loss weighting (λ_1)	0.01	0.01
	supervised. loss weighting (λ_2)	1	1
	learning rate	1e-4	1e-4
	adv. update steps	50	50
Reward	max reward	10	10
	rollout batch size	1000	64
	rollout steps	40	100
	l2 penalty	1e-3	1e-3
	learning rate	1e-4	1e-4
	update steps	1	1

Table 3. Environment-specific hyperparameters

Environment	Hyperparameter	BTOM
Hopper	model rollout batch size	10000
	model rollout steps	40
	model rollout frequency	250
HalfCheetah	model rollout batch size	50000
	model rollout steps	5
	model rollout frequency	250
Walker2d	model rollout batch size	10000
	model rollout steps	40
	model rollout frequency	250

B.1.2. Policy Training

Our policy training process follows MBPO (Janner et al., 2019) which uses SAC with automatic temperature tuning (Haarnoja et al., 2018b). Shared hyperparameters across different environments are listed in Table 2 and environment-specific hyperparameters are listed in Table 3. For the actor and critic, we use feedforward neural networks with 2 hidden layers of 256 units and ReLU activation. We train the actor and critic networks using a combination of real and simulated samples. We use a real ratio of 0.5, which is standard practice in model-based RL and IRL. We found that BTOM requires a higher minimum temperature to stablize training, which is set to $\alpha = 0.1$.

We found that different MuJoCo environments require different model rollout hyperparameters, similar to what's reported in (Lu et al., 2021). Specifically, Hopper and Walker2d only work with significantly larger rollout steps. We decrease their rollout batch size to reduce computational overhead. HalfCheetah on the other hand works better with smaller rollout steps

and larger rollout batch size. In contrast to Lu et al. (2021), we did not use different rollout hyperparameters for different datasets.

B.1.3. Reward and Dynamics Training

We use 10 random trajectories from the D4RL MuJoCo expert dataset after removing all expert trajectories that resulted in terminal states.

We use the same network architecture as the actor-critic to parameterize the reward function. We further clip the reward function to a maximum range of ± 10 and apply l2 regularization on all weights and biases with a penalty of 0.001.

As described in the main text, we update the reward function by simulating sample trajectories and taking a single gradient step. For RTOM, we randomly sample expert trajectory segments of length "rollout steps" and use the first step as the start of our simulated sample paths.

We update the dynamics using on-policy rollouts branched from the dataset state-actions. We use the same batch size for reward and dynamics rollouts, which is 1000 for BTOM and 256 for RTOM. Because only the first steps in BTOM sample paths come from the dataset, it requires a larger batch size to iterate more data samples. We also train BTOM for more epochs than RTOM.

To compute the dynamics log likelihood in the REINFORCE gradient in (18), we treat the ensemble as a uniform mixture and compute the likelihood as:

$$\hat{P}_{\theta_2}(\delta|s,a) = \frac{1}{K}\sum_{k=1}^{K} \hat{P}_{\theta_2}^{(k)}(\delta|s,a) \tag{37}$$

We set the dynamics adversarial loss weighting to $\lambda_1 = 0.01$ for both BTOM and RTOM. We found this to work better than what's in the official RAMBO implementation, which is $\lambda_1 = 0.0768$. Note that the RAMBO author reported $\lambda_1 =$ 3e-4 in their paper but forget to average their REINFORCE loss over the mini-batch of size 256 in their implementation, which is instead treated as a sum by default by TensorFlow. We empirically found that small λ_1 leads to severe model exploitation.