Robust Inverse Reinforcement Learning under Transition Dynamics Mismatch

Luca Viano
LIONS, EPFL

Yu-Ting Huang
EPFL

Parameswaran Kamalaruban*
The Alan Turing Institute

Adrian Weller
University of Cambridge
& The Alan Turing Institute

Volkan Cevher
LIONS, EPFL

Abstract

We study the inverse reinforcement learning (IRL) problem under a transition dynamics mismatch between the expert and the learner. Specifically, we consider the Maximum Causal Entropy (MCE) IRL learner model and provide a tight upper bound on the learner's performance degradation based on the ℓ_1-distance between the transition dynamics of the expert and the learner. Leveraging insights from the Robust RL literature, we propose a robust MCE IRL algorithm, which is a principled approach to help with this mismatch. Finally, we empirically demonstrate the stable performance of our algorithm compared to the standard MCE IRL algorithm under transition dynamics mismatches in both finite and continuous MDP problems.

1 Introduction

Recent advances in Reinforcement Learning (RL) [1, 2, 3, 4] have demonstrated impressive performance in games [5, 6], continuous control [7], and robotics [8]. Despite these successes, a broader application of RL in real-world domains is hindered by the difficulty of designing a proper reward function. Inverse Reinforcement Learning (IRL) addresses this issue by inferring a reward function from a given set of demonstrations of the desired behavior [9, 10]. IRL has been extensively studied, and many algorithms have already been proposed [11, 12, 13, 14, 15, 16].

Almost all IRL algorithms assume that the expert demonstrations are collected from the same environment as the one in which the IRL agent is trained. However, this assumption rarely holds in real world because of many possible factors identified by [17]. For example, consider an autonomous car that should learn by observing expert demonstrations performed on another car with possibly different technical characteristics. There is often a mismatch between the learner and the expert's transition dynamics, resulting in poor performance that are critical in healthcare [18] or autonomous driving [19]. Indeed, the performance degradation of an IRL agent due to transition dynamics mismatch has been noted empirically [20, 21, 22, 23], but without theoretical guidance.

To this end, our work first provides a theoretical study on the effect of such mismatch in the context of the infinite horizon Maximum Causal Entropy (MCE) IRL framework [24, 25, 26]. Specifically, we bound the potential decrease in the IRL learner's performance as a function of the ℓ_1-distance between the expert and the learner's transition dynamics. We then propose a robust variant of the MCE IRL algorithm to effectively recover a reward function under transition dynamics mismatch, mitigating degradation. There is precedence to our robust IRL approach, such as [27] that employs

*Correspondence to: Parameswaran Kamalaruban <kparameswaran@turing.ac.uk>

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

an adversarial training method to learn a robust policy against adversarial changes in the learner's environment. The novel idea of our work is to incorporate this method within our IRL context, by viewing the expert's transition dynamics as a perturbed version of the learner's one.

Our robust MCE IRL algorithm leverages techniques from the robust RL literature [28, 29, 30, 27]. A few recent works [20, 21, 31] attempt to infer the expert's transition dynamics from the demonstration set or via additional information, and then apply the standard IRL method to recover the reward function based on the learned dynamics. Still, the transition dynamics can be estimated only up to a certain accuracy, i.e., a mismatch between the learner's belief and the dynamics of the expert's environment remains. Our robust IRL approach can be incorporated into this research vein to further improve the IRL agent's performance.

To our knowledge, this is the first work that rigorously reconciles model-mismatch in IRL with only one shot access to the expert environment. We highlight the following contributions:

1. We provide a tight upper bound for the suboptimality of an IRL learner that receives expert demonstrations from an MDP with different transition dynamics compared to a learner that receives demonstrations from an MDP with the same transition dynamics (Section 3.1).
2. We find suitable conditions under which a solution exists to the MCE IRL optimization problem with model mismatch (Section 3.2).
3. We propose a robust variant of the MCE IRL algorithm to learn a policy from expert demonstrations under transition dynamics mismatch (Section 4).
4. We demonstrate our method's robust performance compared to the standard MCE IRL in a broad set of experiments under both linear and non-linear reward settings (Section 5).
5. We extend our robust IRL method to the high dimensional continuous MDP setting with appropriate practical relaxations, and empirically demonstrate its effectiveness (Section 6).

2 Problem Setup

This section formalizes the IRL problem with an emphasis on the learner and expert environments. We use bold notation to represent vectors. A glossary of notation is given in Appendix C.

2.1 Environment and Reward

We formally represent the environment by a Markov decision process (MDP) $M_{\boldsymbol{\theta}} := \{\mathcal{S}, \mathcal{A}, T, \gamma, P_0, R_{\boldsymbol{\theta}}\}$, parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$. The state and action spaces are denoted as $\mathcal{S}$ and $\mathcal{A}$, respectively. We assume that $|\mathcal{S}|, |\mathcal{A}| < \infty$. $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ represents the transition dynamics, i.e., $T(s'|s, a)$ is the probability of transitioning to state s' by taking action a from state s. The discount factor is given by $\gamma \in (0, 1)$, and P_0 is the initial state distribution. We consider a linear reward function $R_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ of the form $R_{\boldsymbol{\theta}}(s) = \langle \boldsymbol{\theta}, \boldsymbol{\phi}(s) \rangle$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is the reward parameter, and $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^d$ is a feature map. We use a one-hot feature map $\boldsymbol{\phi} : \mathcal{S} \to \{0, 1\}^{|\mathcal{S}|}$, where the sth element of $\boldsymbol{\phi}(s)$ is 1 and 0 elsewhere. Our results can be extended to any general feature map (see empirical evidence in Fig. 6), but we use this particular choice as a running example for concreteness.

We focus on the state-only reward function since the state-action reward function is not that useful in the robustness context. Indeed, as [22] pointed out, the actions to achieve a specific goal under different transition dynamics will not necessarily be the same and, consequently, should not be imitated. Analogously, in the IRL context, the reward for taking a particular action should not be recovered since the quality of that action depends on the transition dynamics. We denote an MDP without a reward function by $M = M_{\boldsymbol{\theta}} \backslash R_{\boldsymbol{\theta}} = \{\mathcal{S}, \mathcal{A}, T, \gamma, P_0\}$.

2.2 Policy and Performance

A policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ is a mapping from a state to a probability distribution over actions. The set of all valid stochastic policies is denoted by $\Pi := \{\pi : \sum_a \pi(a|s) = 1, \forall s \in \mathcal{S}; \pi(a|s) \geq 0, \forall (s, a) \in \mathcal{S} \times \mathcal{A}\}$. We are interested in two different performance measures of any policy π acting in the MDP $M_{\boldsymbol{\theta}}$: (i) the expected discounted return $V^{\pi}_{M_{\boldsymbol{\theta}}} := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_{\boldsymbol{\theta}}(s_t) \mid \pi, M\right]$, and (ii) its entropy regularized variant $V^{\pi,\text{soft}}_{M_{\boldsymbol{\theta}}} := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left\{R_{\boldsymbol{\theta}}(s_t) - \log \pi(a_t|s_t)\right\} \mid \pi, M\right]$. The state occupancy measure of a policy π in the

MDP M is defined as $\rho_M^\pi(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}\left[s_t = s \mid \pi, M\right]$, where $\mathbb{P}\left[s_t = s \mid \pi, M\right]$ denotes the probability of visiting the state s after t steps by following the policy π in M. Note that $\rho_M^\pi(s)$ does not depend on the reward function. Let $\boldsymbol{\rho}_M^\pi \in \mathbb{R}^{|\mathcal{S}|}$ be a vector whose s^{th} element is $\rho_M^\pi(s)$. For the one-hot feature map $\boldsymbol{\phi}$, we have that $V_{M_{\boldsymbol{\theta}}}^\pi = \frac{1}{1-\gamma}\sum_s \rho_M^\pi(s) R_{\boldsymbol{\theta}}(s) = \frac{1}{1-\gamma}\langle \boldsymbol{\theta}, \boldsymbol{\rho}_M^\pi \rangle$. A policy π is *optimal* for the MDP $M_{\boldsymbol{\theta}}$ if $\pi \in \arg\max_{\pi'} V_{M_{\boldsymbol{\theta}}}^{\pi'}$, in which case we denote it by $\pi_{M_{\boldsymbol{\theta}}}^*$. Similarly, the *soft-optimal* policy (always unique [32]) in $M_{\boldsymbol{\theta}}$ is defined as $\pi_{M_{\boldsymbol{\theta}}}^{\text{soft}} := \arg\max_{\pi'} V_{M_{\boldsymbol{\theta}}}^{\pi',\text{soft}}$ (see Appendix D for a parametric form of this policy).

2.3 Learner and Expert

Our setting has two entities: a learner implementing the MCE IRL algorithm, and an expert. We consider two MDPs, $M_{\boldsymbol{\theta}}^L = \left\{\mathcal{S}, \mathcal{A}, T^L, \gamma, P_0, R_{\boldsymbol{\theta}}\right\}$ and $M_{\boldsymbol{\theta}}^E = \left\{\mathcal{S}, \mathcal{A}, T^E, \gamma, P_0, R_{\boldsymbol{\theta}}\right\}$, that differ only in the transition dynamics. The true reward parameter $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ is known only to the expert. The expert provides demonstrations to the learner: (i) by following policy $\pi_{M_{\boldsymbol{\theta}^*}^E}^*$ in M^E when there is a *transition dynamics mismatch* between the learner and the expert, or (ii) by following policy $\pi_{M_{\boldsymbol{\theta}^*}^L}^*$ in M^L otherwise. The learner always operates in the MDP M^L and is not aware of the true reward parameter and of the expert dynamics T^E[2], i.e., it only has access to $M_{\boldsymbol{\theta}^*}^L \backslash R_{\boldsymbol{\theta}^*}$. It learns a reward parameter $\boldsymbol{\theta}$ and the corresponding soft-optimal policy $\pi_{M_{\boldsymbol{\theta}}^L}^{\text{soft}}$, based on the state occupancy measure $\boldsymbol{\rho}$ received from the expert. Here, $\boldsymbol{\rho}$ is either $\boldsymbol{\rho}_{M^E}^{\pi_{M^E_{\boldsymbol{\theta}^*}}^*}$ or $\boldsymbol{\rho}_{M^L}^{\pi_{M^L_{\boldsymbol{\theta}^*}}^*}$ depending on the case. Our results can be extended to the stochastic estimate of $\boldsymbol{\rho}$ using concentration inequalities [11].

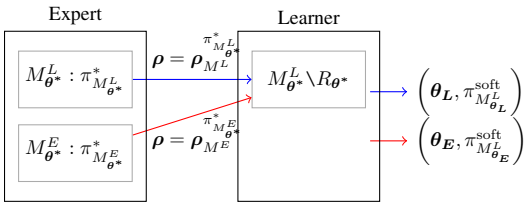


Figure 1: An illustration of the IRL problem under transition dynamics mismatch: See Section 2.

Our learner model builds on the MCE IRL [24, 25, 26] framework that matches the expert's state occupancy measure $\boldsymbol{\rho}$. In particular, the learner policy is obtained by maximizing its causal entropy while matching the expert's state occupancy:

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi(a_t|s_t) \;\middle|\; \pi, M^L\right] \quad \text{subject to} \quad \boldsymbol{\rho}_{M^L}^\pi = \boldsymbol{\rho}. \tag{1}$$

Note that this optimization problem only requires access to $M_{\boldsymbol{\theta}}^L \backslash R_{\boldsymbol{\theta}}$. The constraint in (1) follows from our choice of the one-hot feature map. We denote the optimal solution of the above problem by $\pi_{M_{\boldsymbol{\theta}}^L}^{\text{soft}}$ with a corresponding reward parameter: (i) $\boldsymbol{\theta} = \boldsymbol{\theta_E}$, when we use $\boldsymbol{\rho}_{M^E}^{\pi_{M^E_{\boldsymbol{\theta}^*}}^*}$ as $\boldsymbol{\rho}$, or (ii) $\boldsymbol{\theta} = \boldsymbol{\theta_L}$, when we use $\boldsymbol{\rho}_{M^L}^{\pi_{M^L_{\boldsymbol{\theta}^*}}^*}$ as $\boldsymbol{\rho}$. Here, the parameters $\boldsymbol{\theta_E}$ and $\boldsymbol{\theta_L}$ are obtained by solving the corresponding dual problems of (1). Finally, we are interested in the performance of the learner policy $\pi_{M_{\boldsymbol{\theta}}^L}^{\text{soft}}$ in the MDP $M_{\boldsymbol{\theta}^*}^L$. Our problem setup is illustrated in Figure 1.

3 MCE IRL under Transition Dynamics Mismatch

This section analyses the MCE IRL learner's suboptimality when there is a transition dynamics mismatch between the expert and the learner, as opposed to an ideal learner without this mismatch. The proofs of the theoretical statements of this section can be found in Appendix E.

3.1 Upper bound on the Performance Gap

First, we introduce an auxiliary lemma to be used later in our analysis. We define the distance between the two transition dynamics T and T', and the distance between the two poli-

[2]The setting with T^E known to the learner has been studied under the name of *imitation learning across embodiments* [33].

cies π and π' as follows, respectively: $d_{\text{dyn}}(T, T') := \max_{s,a} \|T(\cdot \mid s, a) - T'(\cdot \mid s, a)\|_1$, and $d_{\text{pol}}(\pi, \pi') := \max_s \|\pi(\cdot|s) - \pi'(\cdot|s)\|_1$. Consider the two MDPs $M_{\boldsymbol{\theta}} = \{\mathcal{S}, \mathcal{A}, T, \gamma, P_0, R_{\boldsymbol{\theta}}\}$ and $M'_{\boldsymbol{\theta}} = \{\mathcal{S}, \mathcal{A}, T', \gamma, P_0, R_{\boldsymbol{\theta}}\}$. We assume that the reward function is bounded, i.e., $R_{\boldsymbol{\theta}}(s) \in \left[R^{\min}_{\boldsymbol{\theta}}, R^{\max}_{\boldsymbol{\theta}}\right], \forall s \in \mathcal{S}$. Also, we define the following two constants: $\kappa_{\boldsymbol{\theta}} := \sqrt{\gamma \cdot \max\left\{R^{\max}_{\boldsymbol{\theta}} + \log|\mathcal{A}|, -\log|\mathcal{A}| - R^{\min}_{\boldsymbol{\theta}}\right\}}$ and $|R_{\boldsymbol{\theta}}|^{\max} := \max\left\{\left|R^{\min}_{\boldsymbol{\theta}}\right|, \left|R^{\max}_{\boldsymbol{\theta}}\right|\right\}$.

Lemma 1. *Let $\pi := \pi^{\text{soft}}_{M_{\boldsymbol{\theta}}}$ and $\pi' := \pi^{\text{soft}}_{M'_{\boldsymbol{\theta}}}$ be the soft optimal policies for the MDPs $M_{\boldsymbol{\theta}}$ and $M'_{\boldsymbol{\theta}}$ respectively. Then, the distance between π and π' is bounded as follows: $d_{\text{pol}}(\pi', \pi) \leq 2 \min\left\{\frac{\kappa_{\boldsymbol{\theta}}\sqrt{d_{\text{dyn}}(T',T)}}{(1-\gamma)}, \frac{\kappa^2_{\boldsymbol{\theta}} d_{\text{dyn}}(T',T)}{(1-\gamma)^2}\right\}$.*

The above result is obtained by bounding the KL divergence between the two soft optimal policies, and involves a non-standard derivation compared to the well-established performance difference theorems in the literature (see Appendix E.1). The lemma above bounds the maximum total variation distance between two soft optimal policies obtained by optimizing the same reward under different transition dynamics. It serves as a prerequisite result for our later theorems (Theorem 1 for soft optimal experts and Theorem 6). In addition, it may be a result of independent interest for entropy regularized MDP.

Now, we turn to our objective. Let $\pi_1 := \pi^{\text{soft}}_{M^L_{\boldsymbol{\theta_L}}}$ be the policy returned by the MCE IRL algorithm when there is no transition dynamics mismatch. Similarly, let $\pi_2 := \pi^{\text{soft}}_{M^L_{\boldsymbol{\theta_E}}}$ be the policy returned by the MCE IRL algorithm when there is a mismatch. Note that π_1 and π_2 are the corresponding solutions to the optimization problem (1), when $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}^{\pi^*_{M^L_{\boldsymbol{\theta}^*}}}_{M^L}$ and $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}^{\pi^*_{M^E_{\boldsymbol{\theta}^*}}}_{M^E}$, respectively. The following theorem bounds the performance degradation of the policy π_2 compared to the policy π_1 in the MDP $M^L_{\boldsymbol{\theta}^*}$, where the learner operates on:

Theorem 1. *The performance gap between the policies π_1 and π_2 on the MDP $M^L_{\boldsymbol{\theta}^*}$ is bounded as follows: $\left|V^{\pi_1}_{M^L_{\boldsymbol{\theta}^*}} - V^{\pi_2}_{M^L_{\boldsymbol{\theta}^*}}\right| \leq \frac{\gamma \cdot |R_{\boldsymbol{\theta}^*}|^{\max}}{(1-\gamma)^2} \cdot d_{\text{dyn}}\left(T^L, T^E\right)$.*

The above result is obtained from the optimality conditions of the problem (1), and using Theorem 7 from [34]. In Section 4.4, we show that the above bound is indeed tight. When the expert policy is soft-optimal, we can use Lemma 1 and Simulation Lemma [35, 36] to obtain an upper bound on the performance gap (see Appendix E.2). For an application of Theorem 1, consider an IRL learner that first learns a simulator of the expert environment, and then matches the expert behavior in the simulator. In this case, our upper bound provides an estimate (sufficient condition) of the accuracy required for the simulator.

3.2 Existence of Solution under Mismatch

The proof of the existence of a unique solution to the optimization problem (1), presented in [37], relies on the fact that both expert and learner environments are the same. This assumption implies that the expert policy is in the feasible set that is consequently non-empty. Theorem 2 presented in this section poses a condition under which we can ensure that the feasible set is non-empty when the expert and learner environments are not the same.

Given M^L and $\boldsymbol{\rho}$, we define the following quantities useful for stating our theorem. We define, for each state $s \in \mathcal{S}$, the probability flow matrix $\boldsymbol{F}(s) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ as follows: $[\boldsymbol{F}(s)]_{i,j} := \rho(s) T^L_{s_i,s,a_j}$, where $T^L_{s_i,s,a_j} := T^L(s_i|s, a_j)$ for $i = 1, \ldots, |\mathcal{S}|$ and $j = 1, \ldots, |\mathcal{A}|$. Let $\boldsymbol{B}(s) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ be a row matrix that contains only ones in row s and zero elsewhere. Then, we define the matrix $\boldsymbol{T} \in \mathbb{R}^{2|\mathcal{S}|\times|\mathcal{S}||\mathcal{A}|}$ by stacking the probability flow and the row matrices as follows: $\boldsymbol{T} := \begin{bmatrix} \boldsymbol{F}(s_1) & \boldsymbol{F}(s_2) & \ldots & \boldsymbol{F}(s_{|\mathcal{S}|}) \\ \boldsymbol{B}(s_1) & \boldsymbol{B}(s_2) & \ldots & \boldsymbol{B}(s_{|\mathcal{S}|}) \end{bmatrix}$. In addition, we define the vector $\boldsymbol{v} \in \mathbb{R}^{2|\mathcal{S}|}$ as follows: $\boldsymbol{v}_i = \rho(s_i) - (1-\gamma)P_0(s_i)$ if $i \leq |\mathcal{S}|$, and 1 otherwise.

Theorem 2. *The feasible set of the optimization problem* (1) *is non-empty iff the rank of the matrix $\boldsymbol{T}$ is equal to the rank of the augmented matrix $(\boldsymbol{T}|\boldsymbol{v})$.*

The proof of the above theorem leverages the fact that the Bellman flow constraints [15] must hold for any policy in an MDP. This requirement leads to the formulation of a linear system whose solutions

set corresponds to the feasible set of (1). The Rouché-Capelli theorem [38][Theorem 2.38] states that the solutions set is non-empty if and only if the condition in Theorem 2 holds. We note that the construction of the matrix $\boldsymbol{T}$ does not assume any restriction on the MDP structure since it leverages only on the Bellman flow constraints. Theorem 2 allows us to develop a robust MCE IRL scheme in Section 4 by ensuring the absence of duality gap. To this end, the following corollary provides a simple sufficient condition for the existence of a solution under transition dynamics mismatch.

Corollary 1. *Let $|\mathcal{A}| > 1$. Then, a sufficient condition for the non-emptiness of the feasible set of the optimization problem* (1) *is given by $\boldsymbol{T}$ being full rank.*

3.3 Reward Transfer under Mismatch

Consider a class $\mathcal{M}$ of MDPs such that it contains both the learner and the expert environments, i.e., $M^L, M^E \in \mathcal{M}$ (see Figure 2). We are given the expert's state occupancy measure $\boldsymbol{\rho} = \boldsymbol{\rho}_{M^E}^{\pi^*_{M^E_{\boldsymbol{\theta}^*}}}$; but the expert's policy $\pi^*_{M^E_{\boldsymbol{\theta}^*}}$ and the MDP M^E are unknown. Further, we assume that every MDP $M \in \mathcal{M}$ satisfies the condition in Theorem 2.

We aim to find a policy π^L that performs well in the MDP $M^L_{\boldsymbol{\theta}^*}$, i.e., $V^{\pi^L}_{M^L_{\boldsymbol{\theta}^*}}$ is high. To this end, we can choose any MDP $M^{\text{train}} \in \mathcal{M}$, and solve the MCE IRL problem (1) with the constraint given by $\boldsymbol{\rho} = \boldsymbol{\rho}^{\pi}_{M^{\text{train}}}$. Then, we always obtain a reward parameter $\boldsymbol{\theta}^{\text{train}}$ s.t. $\boldsymbol{\rho} = \boldsymbol{\rho}_{M^{\text{train}}}^{\pi^{\text{soft}}_{M^{\text{train}}_{\boldsymbol{\theta}^{\text{train}}}}}$, since M^{train} satisfies the condition in Theorem 2. We can use this reward parameter $\boldsymbol{\theta}^{\text{train}}$ to learn a good policy π^L in the MDP $M^L_{\boldsymbol{\theta}^{\text{train}}}$, i.e., $\pi^L := \pi^*_{M^L_{\boldsymbol{\theta}^{\text{train}}}}$ or $\pi^L := \pi^{\text{soft}}_{M^L_{\boldsymbol{\theta}^{\text{train}}}}$. Using Lemma 1, we obtain a bound on the performance gap between π^L and $\pi_1 := \pi^{\text{soft}}_{M^L_{\boldsymbol{\theta}^L}}$ (see Theorem 6 in Appendix E.4).

However, there are two problems with this approach: (i) it requires access to multiple environments M^{train}, and (ii) unless M^{train} happened to be closer to the expert's MDP M^E, we cannot recover the true intention of the expert. Since the MDP M^E is unknown, one cannot compare the different reward parameters $\boldsymbol{\theta}^{\text{train}}$'s obtained with different MDPs M^{train}'s. Thus, with $\boldsymbol{\theta}^{\text{train}}$, it is impossible to ensure that the performance of π^L is high in the MDP $M^L_{\boldsymbol{\theta}^*}$. Instead, we try to learn a robust policy π^L over the class $\mathcal{M}$, while aligning with the expert's occupancy measure $\boldsymbol{\rho}$, and acting only in M^L. By doing this, we ensure that π^L performs reasonably well on any MDP $M_{\boldsymbol{\theta}^*} \in \mathcal{M}$ including $M^L_{\boldsymbol{\theta}^*}$. We further build upon this idea in the next section.

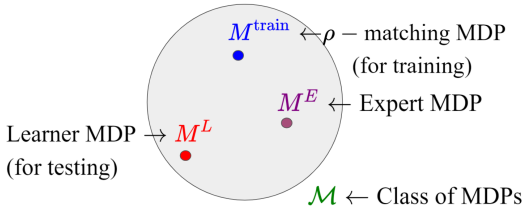


Figure 2: Illustrative example of learning a policy π^L to act in one MDP M^L, given the expert occupancy measure $\boldsymbol{\rho}$.

4 Robust MCE IRL via Two-Player Markov Game

4.1 Robust MCE IRL Formulation

This section focuses on recovering a learner policy via MCE IRL framework in a robust manner, under transition dynamics mismatch, i.e., $\boldsymbol{\rho} = \boldsymbol{\rho}_{M^E}^{\pi^{\text{soft}}_{M^E_{\boldsymbol{\theta}^*}}}$ in Eq. (1). In particular, our learner policy matches the expert state occupancy measure $\boldsymbol{\rho}$ under the most adversarial transition dynamics belonging to a set described as follows for a given $\alpha > 0$: $\mathcal{T}^{L,\alpha} := \left\{\alpha T^L + (1-\alpha)\bar{T}, \forall \bar{T} \in \Delta_{\mathcal{S}|\mathcal{S},\mathcal{A}}\right\}$, where $\Delta_{\mathcal{S}|\mathcal{S},\mathcal{A}}$ is the set of all the possible transition dynamics $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0,1]$. Note that the set $\mathcal{T}^{L,\alpha}$ is equivalent to the (s,a)-rectangular uncertainty set [28] centered around T^L, i.e., $\mathcal{T}^{L,\alpha} = \left\{T : d_{\text{dyn}}\left(T, T^L\right) \leq 2(1-\alpha)\right\}$. We need this set $\mathcal{T}^{L,\alpha}$ for establishing the equivalence between robust MDP and action-robust MDP formulations. The action-robust MDP formulation allows us to learn a robust policy while accessing only the MDP M^L.

We define a class of MDPs as follows: $\mathcal{M}^{L,\alpha} := \left\{\left\{\mathcal{S}, \mathcal{A}, T^{L,\alpha}, \gamma, P_0\right\}, \forall T^{L,\alpha} \in \mathcal{T}^{L,\alpha}\right\}$. Then, based on the discussions in Section 3.3, we propose the following robust MCE IRL problem:

$$\max_{\pi^{\mathrm{pl}} \in \Pi} \min_{M \in \mathcal{M}^{L,\alpha}} \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi^{\mathrm{pl}}(a_t|s_t) \,\middle|\, \pi^{\mathrm{pl}}, M\right] \text{ subject to } \boldsymbol{\rho}_M^{\pi^{\mathrm{pl}}} = \boldsymbol{\rho} \tag{2}$$

The corresponding dual problem is given by:

$$\min_{\boldsymbol{\theta}} \max_{\pi^{\mathrm{pl}} \in \Pi} \min_{M \in \mathcal{M}^{L,\alpha}} \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi^{\mathrm{pl}}(a_t|s_t) \,\middle|\, \pi^{\mathrm{pl}}, M\right] + \boldsymbol{\theta}^\top \left(\boldsymbol{\rho}_M^{\pi^{\mathrm{pl}}} - \boldsymbol{\rho}\right) \tag{3}$$

In the dual problem, for any $\boldsymbol{\theta}$, we attempt to learn a robust policy over the class $\mathcal{M}^{L,\alpha}$ with respect to the entropy regularized reward function. The parameter $\boldsymbol{\theta}$ plays the role of aligning the learner's policy with the expert's occupancy measure via constraint satisfaction.

4.2 Existence of Solution

We start by formulating the IRL problem for any MDP $M^{L,\alpha} \in \mathcal{M}^{L,\alpha}$, with transition dynamics $T^{L,\alpha} = \alpha T^L + (1-\alpha)\bar{T} \in \mathcal{T}^{L,\alpha}$, as follows:

$$\max_{\pi^{\mathrm{pl}} \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi^{\mathrm{pl}}(a_t|s_t) \,\middle|\, \pi^{\mathrm{pl}}, M^{L,\alpha}\right] \text{ subject to } \boldsymbol{\rho}_{M^{L,\alpha}}^{\pi^{\mathrm{pl}}} = \boldsymbol{\rho} \tag{4}$$

By introducing the Lagrangian vector $\boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{S}|}$, we get:

$$\max_{\pi^{\mathrm{pl}} \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi^{\mathrm{pl}}(a_t|s_t) \,\middle|\, \pi^{\mathrm{pl}}, M^{L,\alpha}\right] + \boldsymbol{\theta}^\top \left(\boldsymbol{\rho}_{M^{L,\alpha}}^{\pi^{\mathrm{pl}}} - \boldsymbol{\rho}\right) \tag{5}$$

For any fixed $\boldsymbol{\theta}$, the problem (5) is feasible since Π is a closed and bounded set. We define $U(\boldsymbol{\theta})$ as the value of the program (5) for a given $\boldsymbol{\theta}$. By weak duality, $U(\boldsymbol{\theta})$ provides an upper bound on the optimization problem (4). Consequently, we introduce the dual problem aiming to find the value of $\boldsymbol{\theta}$ corresponding to the lowest upper bound, which can be written as

$$\min_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) := \max_{\pi^{\mathrm{pl}} \in \Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi^{\mathrm{pl}}(a_t|s_t) \,\middle|\, \pi^{\mathrm{pl}}, M^{L,\alpha}\right] + \boldsymbol{\theta}^\top \left(\boldsymbol{\rho}_{M^{L,\alpha}}^{\pi^{\mathrm{pl}}} - \boldsymbol{\rho}\right). \tag{6}$$

Given $\boldsymbol{\theta}$, we define $\pi^{\mathrm{pl},*} := \pi^{\mathrm{soft}}_{M^{L,\alpha}_{\boldsymbol{\theta}}}$. Due to [32][Theorem 1], for any fixed $M^{L,\alpha}_{\boldsymbol{\theta}}$, the policy $\pi^{\mathrm{pl},*}$ exists and it is unique. We can compute the gradient[3] $\nabla_{\boldsymbol{\theta}} U = \boldsymbol{\rho}_{M^{L,\alpha}}^{\pi^{\mathrm{pl},*}} - \boldsymbol{\rho}$, and update the parameter via gradient descent: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} U$. Note that, if the condition in Theorem 2 holds, the feasible set of (4) is non-empty. Then, according to [37][Lemma 2], there is no duality gap between the programs (4) and (6). Based on these observations, we argue that the program (2) is well-posed and admits a unique solution.

4.3 Solution via Markov Game

In the following, we outline a method (see Algorithm 1) to solve the robust MCE IRL dual problem (3). To this end, for any given $\boldsymbol{\theta}$, we need to solve the inner max-min problem of (3). First, we express the entropy term $\mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \log \pi^{\mathrm{pl}}(a_t|s_t) \middle| \pi^{\mathrm{pl}}, M\right]$ as follows:

$$\sum_{s \in \mathcal{S}} \rho_M^{\pi^{\mathrm{pl}}}(s) \sum_{a \in \mathcal{A}} \left\{-\pi^{\mathrm{pl}}(a|s) \log \pi^{\mathrm{pl}}(a|s)\right\} = \sum_{s \in \mathcal{S}} \rho_M^{\pi^{\mathrm{pl}}}(s) H^{\pi^{\mathrm{pl}}}(A \mid S = s) = \left(\boldsymbol{H}^{\pi^{\mathrm{pl}}}\right)^\top \boldsymbol{\rho}_M^{\pi^{\mathrm{pl}}},$$

where $\boldsymbol{H}^{\pi^{\mathrm{pl}}} \in \mathbb{R}^{|\mathcal{S}|}$ a vector whose s^{th} element is the entropy of the player policy given the state s. Since the quantity $\boldsymbol{H}^{\pi^{\mathrm{pl}}} + \boldsymbol{\theta}$ depends only on the states, to solve the dual problem, we can utilize the equivalence between the *robust MDP* [28, 29] formulation and the *action-robust MDP* [30, 27, 40] formulation shown in [27]. We can interpret the minimization over the environment class as the

[3] In Appendix F.2, we proved that this is indeed the gradient update under the transition dynamics mismatch.

Algorithm 1 Robust MCE IRL via Markov Game

Input: opponent strength $1-\alpha$
Initialize: player policy π^{pl}, opponent policy π^{op}, and parameter $\boldsymbol{\theta}$
while not converged **do**
  compute $\boldsymbol{\rho}_{M^L}^{\alpha\pi^{\mathrm{pl}}+(1-\alpha)\pi^{\mathrm{op}}}$ by dynamic programming [37][Section V.C].
  update $\boldsymbol{\theta}$ with Adam [39] using the gradient $\left(\boldsymbol{\rho}_{M^L}^{\alpha\pi^{\mathrm{pl}}+(1-\alpha)\pi^{\mathrm{op}}} - \boldsymbol{\rho}\right)$.
  use Algorithm 2 with $R = R_{\boldsymbol{\theta}}$ to update π^{pl} and π^{op} s.t. they solve the problem (9).
end while
Output: player policy π^{pl}

minimization over a set of opponent policies that with probability $1-\alpha$ take control of the agent and perform the worst possible move from the current agent state. Indeed, interpreting $\left(\boldsymbol{H}^{\pi^{\mathrm{pl}}}+\boldsymbol{\theta}\right)^\top \boldsymbol{\rho}_M^{\pi^{\mathrm{pl}}}$ as an entropy regularized value function, i.e., $\boldsymbol{\theta}$ as a reward parameter, we can write:

$$\max_{\pi^{\mathrm{pl}}\in\Pi}\min_{M\in\mathcal{M}^{L,\alpha}}\left(\boldsymbol{H}^{\pi^{\mathrm{pl}}}+\boldsymbol{\theta}\right)^\top \boldsymbol{\rho}_M^{\pi^{\mathrm{pl}}} = \max_{\pi^{\mathrm{pl}}\in\Pi}\min_{\bar{T}}\mathbb{E}\left[G \mid \pi^{\mathrm{pl}}, P_0, \alpha T^L + (1-\alpha)\bar{T}\right] \tag{7}$$

$$\leq \max_{\pi^{\mathrm{pl}}\in\Pi}\min_{\pi^{\mathrm{op}}\in\Pi}\mathbb{E}\left[G \mid \alpha\pi^{\mathrm{pl}} + (1-\alpha)\pi^{\mathrm{op}}, M^L\right], \tag{8}$$

where $G := \sum_{t=0}^{\infty}\gamma^t\left\{R_{\boldsymbol{\theta}}(s_t) + H^{\pi^{\mathrm{pl}}}\left(A \mid S=s_t\right)\right\}$. The above inequality holds due to the derivation in section 3.1 of [27]. Further details are in Appendix F.1.

Finally, we can formulate the problem (8) as a two-player zero-sum Markov game [41] with transition dynamics given by $T^{\mathrm{two},L,\alpha}(s'|s,a^{\mathrm{pl}},a^{\mathrm{op}}) = \alpha T^L(s'|s,a^{\mathrm{pl}}) + (1-\alpha)T^L(s'|s,a^{\mathrm{op}})$, where a^{pl} is an action chosen according to the player policy and a^{op} according to the opponent policy. Note that the opponent is restricted to take the worst possible action from the state of the player, i.e., there is no additional state variable for the opponent. As a result, we reach a two-player Markov game with a regularization term for the player as follows:

$$\arg\max_{\pi^{\mathrm{pl}}\in\Pi}\min_{\pi^{\mathrm{op}}\in\Pi}\mathbb{E}\left[G \mid \pi^{\mathrm{pl}}, \pi^{\mathrm{op}}, M^{\mathrm{two},L,\alpha}\right], \tag{9}$$

where $M^{\mathrm{two},L,\alpha} = \left\{\mathcal{S},\mathcal{A},\mathcal{A},T^{\mathrm{two},L,\alpha},\gamma,P_0,R_{\boldsymbol{\theta}}\right\}$ is the two-player MDP associated with the above game. The repetition of the action space $\mathcal{A}$ denotes the fact that player and adversary share the same action space. Inspired from [42], we propose a dynamic programming approach to find the player and opponent policies (see Algorithm 2 in Appendix F.3).

4.4 Performance Gap of Robust MCE IRL

Let π^{pl} be the policy returned by our Algorithm 1 when there is a transition dynamics mismatch. Recall that $\pi_1 := \pi^{\mathrm{soft}}_{M^L_{\boldsymbol{\theta}_L}}$ is the policy recovered without this mismatch. Then, we obtain the following upper-bound[4] for the performance gap of our algorithm via the triangle inequality:

Theorem 3. *The performance gap between the policies π_1 and π^{pl} on the MDP $M^L_{\boldsymbol{\theta}^*}$ is bounded as follows:* $\left|V^{\pi_1}_{M^L_{\boldsymbol{\theta}^*}} - V^{\pi^{\mathrm{pl}}}_{M^L_{\boldsymbol{\theta}^*}}\right| \leq \frac{|R_{\boldsymbol{\theta}^*}|^{\max}}{(1-\gamma)^2}\cdot\left\{\gamma\cdot d_{\mathrm{dyn}}\left(T^L,T^E\right) + 2\cdot(1-\alpha)\right\}$.

However, we now provide a constructive example, in which, by choosing the appropriate value for α, the performance gap of our Algorithm 1 vanishes. In contrast, the performance gap of the standard MCE IRL is proportional to the mismatch. Note that our Algorithm 1 with $\alpha = 1$ corresponds to the standard MCE-IRL algorithm.

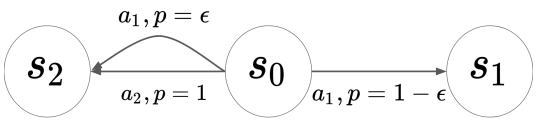


Figure 3: Constructive example to study the performance gap of Algorithm 1 and the MCE IRL.

[4]This bound is worst than the one given in Theorem 1. When the condition in Theorem 2 does not hold, the robust MCE IRL achieves a tighter bound than the MCE IRL for a proper choice of α (see Appendix F.5).

Consider a reference MDP $M^{(\epsilon)} = \left\{\mathcal{S}, \mathcal{A}, T^{(\epsilon)}, \gamma, P_0\right\}$ with variable ϵ (see Figure 3). The state space is $\mathcal{S} = \{s_0, s_1, s_1\}$, where s_1 and s_2 are absorbing states. The action space is $\mathcal{A} = \{a_1, a_2\}$ and the initial state distribution is $P_0\left(s_0\right) = 1$. The transition dynamics is defined as: $T^{(\epsilon)}(s_1|s_0, a_1) = 1-\epsilon$, $T^{(\epsilon)}(s_2|s_0, a_1) = \epsilon$, $T^{(\epsilon)}(s_1|s_0, a_2) = 0$, and $T^{(\epsilon)}(s_2|s_0, a_2) = 1$. The true reward function is given by: $R_{\boldsymbol{\theta}^*}\left(s_0\right) = 0$, $R_{\boldsymbol{\theta}^*}\left(s_1\right) = 1$, and $R_{\boldsymbol{\theta}^*}\left(s_2\right) = -1$. We define the learner and the expert environment as: $M^L := M^{(0)}$ and $M^L := M^{(\epsilon_E)}$. Note that the distance between the two transition dynamics is $d_{\text{dyn}}\left(T^L, T^E\right) = 2\epsilon_E$. Let π^{pl} and $\pi_2 := \pi^{\text{soft}}_{M^L_{\boldsymbol{\theta_E}}}$ be the policies returned by Algorithm 1 and the MCE IRL algorithm, under the above mismatch. Recall that π_1 is the policy recovered by the MCE IRL algorithm without this mismatch. Then, the following holds:

Theorem 4. *For this example, the performance gap of Algorithm 1 vanishes by choosing* $\alpha = 1 - \frac{d_{\text{dyn}}(T^L, T^E)}{2}$*, i.e.,* $\left|V^{\pi_1}_{M^L_{\boldsymbol{\theta}^*}} - V^{\pi^{\text{pl}}}_{M^L_{\boldsymbol{\theta}^*}}\right| = 0$*. Whereas, the performance gap of the standard MCE IRL is given by:* $\left|V^{\pi_1}_{M^L_{\boldsymbol{\theta}^*}} - V^{\pi_2}_{M^L_{\boldsymbol{\theta}^*}}\right| = \frac{\gamma}{1-\gamma} \cdot d_{\text{dyn}}(T^L, T^E)$.

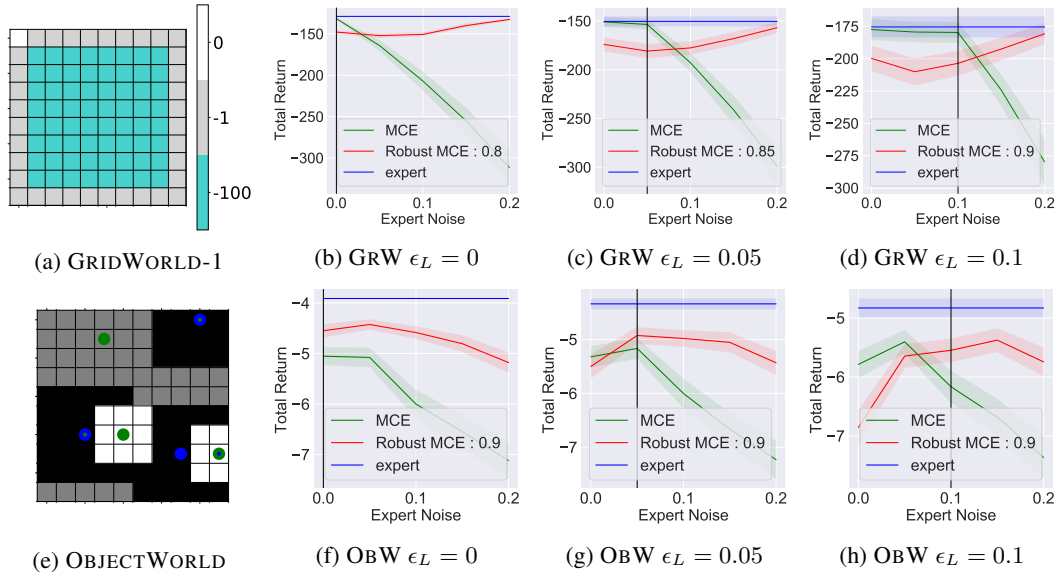
5 Experiments

This section demonstrates the superior performance of our Algorithm 1 compared to the standard MCE IRL algorithm, when there is a transition dynamics mismatch between the expert and the learner. All the missing figures and hyper-parameter details are reported in Appendix G.

Setup. Let $M^{\text{ref}}_{\boldsymbol{\theta}^*} = \left(\mathcal{S}, \mathcal{A}, T^{\text{ref}}, \gamma, P_0, R_{\boldsymbol{\theta}^*}\right)$ be a reference MDP. Given a *learner noise* $\epsilon_L \in [0, 1]$, we introduce a learner MDP without reward function as $M^{L,\epsilon_L} = \left(\mathcal{S}, \mathcal{A}, T^{L,\epsilon_L}, \gamma, P_0\right)$, where $T^{L,\epsilon_L} \in \Delta_{\mathcal{S}|\mathcal{S},\mathcal{A}}$ is defined as $T^{L,\epsilon_L} := (1-\epsilon_L)T^{\text{ref}} + \epsilon_L \bar{T}$ with $\bar{T} \in \Delta_{\mathcal{S}|\mathcal{S},\mathcal{A}}$. Similarly, given an *expert noise* $\epsilon_E \in [0, 1]$, we define an expert MDP $M^{E,\epsilon_E}_{\boldsymbol{\theta}^*} = \left(\mathcal{S}, \mathcal{A}, T^{E,\epsilon_E}, \gamma, P_0, R_{\boldsymbol{\theta}^*}\right)$, where $T^{E,\epsilon_E} \in \Delta_{\mathcal{S}|\mathcal{S},\mathcal{A}}$ is defined as $T^{E,\epsilon_E} := (1-\epsilon_E)T^{\text{ref}} + \epsilon_E \bar{T}$ with $\bar{T} \in \Delta_{\mathcal{S}|\mathcal{S},\mathcal{A}}$. Note that a pair (ϵ_E, ϵ_L) corresponds to an IRL problem under dynamics mismatch, where the expert acts in the MDP $M^{E,\epsilon_E}_{\boldsymbol{\theta}^*}$ and the learner in M^{L,ϵ_L}. In our experiments, we set T^{ref} to be deterministic, and $\bar{T}$ to be uniform. Then, one can easily show that $d_{\text{dyn}}\left(T^{L,\epsilon_L}, T^{E,\epsilon_E}\right) = 2\left(1 - \frac{1}{|\mathcal{S}|}\right)|\epsilon_L - \epsilon_E|$. The learned policies are evaluated in the MDP $M^{L,\epsilon_L}_{\boldsymbol{\theta}^*}$, i.e., M^{L,ϵ_L} endowed with the true reward function $R_{\boldsymbol{\theta}^*}$.

Baselines. We are not aware of any comparable prior IRL work that exactly matches our setting: (i) only one shot access to the expert environment, and (ii) do not explicitly model the expert environment. Note that Algorithm 2 in [33] requires online access to T^E (or the expert environment) to empirically estimate the gradient for every (time step) adversarial expert policy $\check{\pi}^*$, whereas we do not access the expert environment after obtaining a batch of demonstrations, i.e., $\boldsymbol{\rho}$. Thus, for each pair (ϵ_E, ϵ_L), we compare the performance of the following: (i) our robust MCE IRL algorithm with different values of $\alpha \in \{0.8, 0.85, 0.9, 0.95\}$, (ii) the standard MCE IRL algorithm, and (iii) the ideal baseline that utilizes the knowledge of the true reward function, i.e, $\pi^*_{M^{L,\epsilon_L}_{\boldsymbol{\theta}^*}}$.

Environments. We consider four GRIDWORLD environments and an OBJECTWORLD [43] environment. All of them are $N \times N$ grid, where a cell represents a state. There are four actions per state, corresponding to steps in one of the four cardinal directions; T^{ref} is defined accordingly. GRIDWORLD environments are endowed with a linear reward function $R_{\boldsymbol{\theta}^*}(s) = \langle \boldsymbol{\theta}^*, \boldsymbol{\phi}(s)\rangle$, where $\boldsymbol{\phi}$ is a one-hot feature map. The entries $\boldsymbol{\theta}^*_{\boldsymbol{s}}$ of the parameter $\boldsymbol{\theta}^*$ for each state $s \in \mathcal{S}$ are shown in Figures 4a, 10e, 10i, and 10m. OBJECTWORLD is endowed with a non-linear reward function, determined by the distance of the agent to the objects that are randomly placed in the environment. Each object has an outer and an inner color; however, only the former plays a role in determining the reward while the latter serves as a distractor. The reward is -2 in positions within three cells to an outer blue object (black areas of Figure 4e), 0 if they are also within two cells from an outer green object (white areas), and -1 otherwise (gray areas). We shift the rewards originally proposed by [43] to non-positive values, and we randomly placed the goal state in a white area. We also modify the reward features by augmenting them with binary features indicating whether the goal state has been reached. These changes simplify the application of the MCE IRL algorithm in the infinite horizon setting. For this non-linear reward setting, we used the deep MCE IRL algorithm from [44], where the reward function is parameterized by a neural network.



(a) GRIDWORLD-1 (b) GRW $\epsilon_L = 0$ (c) GRW $\epsilon_L = 0.05$ (d) GRW $\epsilon_L = 0.1$

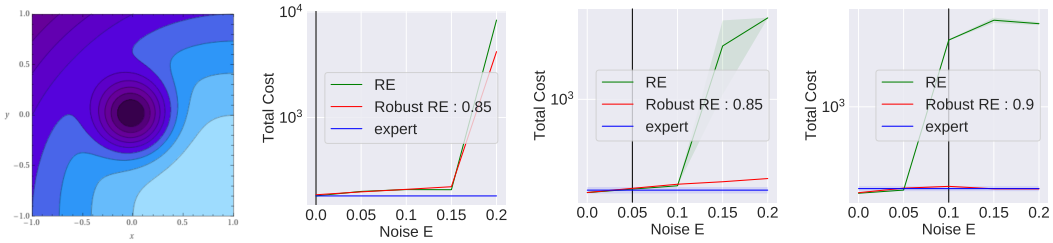
(e) OBJECTWORLD (f) OBW $\epsilon_L = 0$ (g) OBW $\epsilon_L = 0.05$ (h) OBW $\epsilon_L = 0.1$

Figure 4: Comparison of the performance our Algorithm 1 against the baselines, under different levels of mismatch: $(\epsilon_E, \epsilon_L) \in \{0.0, 0.05, 0.1, 0.15, 0.2\} \times \{0.0, 0.05, 0.1\}$. Each plot corresponds to a fixed leaner environment M^{L,ϵ_L} with $\epsilon_L \in \{0.0, 0.05, 0.1\}$. The values of α used for Algorithm 1 are reported in the legend. The vertical line indicates the position of the learner environment in the x-axis. We abbreviated the environment names as GRW, and OBW. Note that our Robust MCE IRL outperforms standard MCE IRL when the expert noise increases along the x-axis. At the same time, Robust MCE IRL might perform slightly worse in the low expert noise regime. This observation aligns with the overly conservative nature of robust training methods.

Results. In Figure 4, we have presented the results for two of the environments, and the complete results can be found in Figure 10. Also, in Figure 4, we have reported the results of our algorithm with the best performing value of α; and the performance of our algorithm with different values of α are presented in Figure 11. In all the plots, every point in the x-axis corresponds to a pair (ϵ_E, ϵ_L). For example, consider Figure 4b, for a fixed learner environment M^{L,ϵ_L} with $\epsilon_L = 0$, and different expert environments M^{E,ϵ_E} by varying ϵ_E along the x-axis. Note that, in this figure, the distance $d_{\text{dyn}}\left(T^{L,\epsilon_L}, T^{E,\epsilon_E}\right) \propto |\epsilon_L - \epsilon_E|$ increases along the x-axis. For each pair (ϵ_E, ϵ_L), in the y-axis, we present the performance of the learned polices in the MDP $M_{\boldsymbol{\theta}^*}^{L,\epsilon_L}$, i.e., $V^{\pi}_{M_{\boldsymbol{\theta}^*}^{L,\epsilon_L}}$. In alignment with our theory, the performance of the standard MCE IRL algorithm degrades along the x-axis. Whereas, our Algorithm 1 resulted in robust performance (even closer to the ideal baseline) across different levels of mismatch. These results confirm the efficacy of our method under mismatch. However, one has to carefully choose the value of $1-\alpha$ (s.t. $T^{E,\epsilon_E} \in \mathcal{T}^{L,\alpha}$): (i) underestimating it would lead to a linear decay in the performance, similar to the MCE IRL, (ii) overestimating it would also slightly hinder the performance, and (iii) given a rough estimate $\widehat{T}^E$ of the expert dynamics, choosing $1-\alpha \approx \frac{d_{\text{dyn}}(T^L, \widehat{T}^E)}{2}$ would lead to better performance in practice. The potential drop in the performance of our Robust MCE IRL method under the low expert noise regime (see Figures 4c, 4d, and 4h) can be related to the overly conservative nature of robust training. See Appendix G.3 for more discussion on the choice of $1-\alpha$. In addition, we have tested our method on a setting with low-dimensional feature mapping $\boldsymbol{\phi}$, where we observed significant improvement over the standard MCE IRL (see Appendix G.2).

6 Extension to Continuous MDP Setting

In this section, we extend our ideas to the continuous MDP setting, i.e., the environments with continuous state and action spaces. In particular, we implement a robust variant of the Relative Entropy IRL (RE IRL) [15] algorithm (see Algorithm 3 in Appendix H). We cannot use the dynamic programming approach to find the player and opponent policies in the continuous MDP setting. Therefore, we solve the two-player Markov game in a model-free manner using the policy gradient methods (see Algorithm 4 in Appendix H).



(a) GAUSSIANGRID (b) M^{L,ϵ_L} with $\epsilon_L = 0$ (c) M^{L,ϵ_L} with $\epsilon_L = 0.05$ (d) M^{L,ϵ_L} with $\epsilon_L = 0.1$

Figure 5: Comparison of the performance our Robust RE IRL (Algorithm 3) against the standard RE IRL, under different levels of mismatch: $(\epsilon_E, \epsilon_L) \in \{0.0, 0.05, 0.1, 0.15, 0.2\} \times \{0.0, 0.05, 0.1\}$. Each plot corresponds to a fixed leaner environment M^{L,ϵ_L} with $\epsilon_L \in \{0.0, 0.05, 0.1\}$. The values of α used for Algorithm 3 are reported in the legend. The vertical line indicates the position of the learner environment in the x-axis. The results are averaged across 5 seeds.

We evaluate the performance of our Robust RE IRL method on a continuous gridworld environment that we called GAUSSIANGRID. The details of the environment and the experimental setup are given in Appendix H. The results are reported in Figure 5, where we notice that our Robust RE IRL method outperforms standard RE IRL.

7 Related Work

In the context of forward RL, there are works that build on the robust MDP framework [28, 29, 45], for example, [46, 47, 48]. However, our work is closer to the line of work that leverages on the equivalence between action-robust and robust MDPs [49, 50, 30, 27, 40]. To our knowledge, this is the first work to adapt the robust RL methods in the IRL context. Other works study the IRL problem under a mismatch between the learner and the expert's worldviews [51, 52]. However, these works do not consider the dynamics mismatch.

Generative Adversarial Imitation Learning (GAIL) [53] and its variants are IRL methods that use a GAN-based reward to align the distribution of the state-action pairs between the expert and the learner. When there is a transition dynamics mismatch, the expert's actions are not quite useful for imitation. [54, 55] have considered state only distribution matching when the expert actions are not observable. Building on these works, [22, 23] have studied the imitation learning problem under transition dynamics mismatch. These works propose model-alignment based imitation learning algorithms in the high dimensional settings to address the dynamics mismatch. Finally, our work has the following important differences with AIRL [56]. In AIRL, the learner has access to the expert environment during the training phase, i.e., there is no transition dynamics mismatch during the training phase but only at test time. In contrast, we consider a different setting where the learner can not access the expert environment during the training phase. In addition, AIRL requires input demonstrations containing both states and actions, while our algorithm requires state-only demonstrations.

8 Conclusions

In this work, we theoretically analyze the MCE IRL algorithm under the transition dynamics mismatch: (i) we derive necessary and sufficient conditions for the existence of solution, and (ii) we provide a tight upper bound on the performance degradation. We propose a robust MCE IRL algorithm and empirically demonstrate its significant improvement over the standard MCE IRL under dynamics mismatch. Even though our Algorithm 1 is not essentially different from the standard robust RL methods, it poses additional theoretical challenges in the IRL context compared to the RL setup. In particular, we have proved: (i) the existence of solution for the robust MCE IRL formulation, and (ii) the performance gap improvement of our algorithm compared to the non-robust MCE IRL in a constructive example. We present empirical results for the settings not covered by our theory: MDPs with non-linear reward function and continuous state and action spaces.

Code Repository

`https://github.com/lviano/RobustMCE_IRL/tree/master/robustIRLcode`

Acknowledgments and Disclosure of Funding

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 725594 - time-data). Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-19-1-0404, by the Department of the Navy, Office of Naval Research (ONR) under a grant number N62909-17-1-2111 and by Hasler Foundation Program: Cyber Human Systems (project number 16066). This work has been supported by 2021 gift from the Schindler Group for research excellence in reinforcement learning.

Parameswaran Kamalaruban acknowledges support from The Alan Turing Institute. He carried out part of this work while at LIONS, EPFL.

Adrian Weller acknowledges support from a Turing AI Fellowship under grant EP/V025379/1, The Alan Turing Institute, and the Leverhulme Trust via CFI.

References

[1] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

[2] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2014.

[3] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2015.

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 2015.

[6] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 2017.

[7] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proc. Intl Conf. on Learning Representations (ICLR)*, 2016.

[8] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 2016.

[9] Stuart Russell. Learning agents for uncertain environments. In *Proc. Conf. on Learning Theory (COLT)*, 1998.

[10] Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2000.

[11] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2004.

[12] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2006.

[13] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Proc. AAAI Conference on Artificial Intelligence*, 2008.

[14] Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

[15] Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proc. Intl Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[16] T Osa, J Pajarinen, G Neumann, JA Bagnell, P Abbeel, and J Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 2018.

[17] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

[18] Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: A survey. *arXiv preprint arXiv:1908.08796*, 2019.

[19] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2002.00444*, 2020.

[20] Sid Reddy, Anca Dragan, and Sergey Levine. Where do you think you're going?: Inferring beliefs about dynamics from behavior. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[21] Ze Gong and Yu Zhang. What is it you really want of me? generalized reward learning with biased beliefs about domain dynamics. In *Proc. AAAI Conference on Artificial Intelligence*, 2020.

[22] Tanmay Gangwani and Jian Peng. State-only imitation with transition dynamics mismatch. In *Proc. Intl Conf. on Learning Representations (ICLR)*, 2020.

[23] Fangchen Liu, Zhan Ling, Tongzhou Mu, and Hao Su. State alignment-based imitation learning. In *Proc. Intl Conf. on Learning Representations (ICLR)*, 2020.

[24] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, Carnegie Mellon University, 2010.

[25] Brian D Ziebart, J Andrew Bagnell, and Anind K Dey. The principle of maximum causal entropy for estimating interacting processes. *IEEE Transactions on Information Theory*, 2013.

[26] Zhengyuan Zhou, Michael Bloem, and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. *IEEE Transactions on Automatic Control*, 2017.

[27] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2019.

[28] Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 2005.

[29] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 2005.

[30] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2017.

[31] Michael Herman, Tobias Gindele, Jörg Wagner, Felix Schmitt, and Wolfram Burgard. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Proc. Intl Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[32] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2019.

[33] Xiangli Chen, Mathew Monfort, Brian D Ziebart, and Peter Carr. Adversarial inverse optimal control for general imitation learning losses and embodiment transfer. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2016.

[34] Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Multi-task reinforcement learning as a hidden-parameter block mdp. *arXiv preprint arXiv:2007.07206*, 2020.

[35] Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynominal time. In *Proc. Intl Conf. on Machine Learning (ICML)*, 1998.

[36] Eyal Even-Dar and Yishay Mansour. Approximate equivalence of Markov decision processes. In *Learning Theory and Kernel Machines*. 2003.

[37] Michael Bloem and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *IEEE Conference on Decision and Control*, 2014.

[38] I.R. Shafarevich, A.O. Remizov, D.P. Kramer, and L. Nekludova. *Linear Algebra and Geometry*. Springer Berlin Heidelberg, 2014.

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Intl Conf. on Learning Representations (ICLR)*, 2015.

[40] Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[41] Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings*. 1994.

[42] Jordi Grau-Moya, Felix Leibfried, and Haitham Bou-Ammar. Balancing two-player stochastic games with soft q-learning. In *Proc. Intl Joint Conf. on Artificial Intelligence (IJCAI)*, 2018.

[43] Sergey Levine, Zoran Popovic, and Vladlen Koltun. Nonlinear inverse reinforcement learning with gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[44] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

[45] Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 2013.

[46] Shirli Di-Castro Shashua and Shie Mannor. Deep robust kalman filter. *arXiv preprint arXiv:1703.02310*, 2017.

[47] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE international conference on robotics and automation (ICRA)*, 2018.

[48] Daniel J Mankowitz, Nir Levine, Rae Jeong, Abbas Abdolmaleki, Jost Tobias Springenberg, Yuanyuan Shi, Jackie Kay, Todd Hester, Timothy Mann, and Martin Riedmiller. Robust reinforcement learning for continuous control with model misspecification. In *Proc. Intl Conf. on Learning Representations (ICLR)*, 2020.

[49] Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 2005.

[50] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.

[51] Luis Haug, Sebastian Tschiatschek, and Adish Singla. Teaching inverse reinforcement learners via features and demonstrations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[52] Sebastian Tschiatschek, Ahana Ghosh, Luis Haug, Rati Devidze, and Adish Singla. Learner-aware teaching: Inverse reinforcement learning with preferences and constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[53] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[54] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

[55] Wen Sun, Anirudh Vemula, Byron Boots, and J Andrew Bagnell. Provably efficient imitation learning from observation alone. In *Proc. Intl Conf. on Machine Learning (ICML)*, 2019.

[56] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *Proc. Intl Conf. on Learning Representations (ICLR)*, 2018.

[57] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement Learning: Theory and Algorithms, 2019.

[58] Abdeslam Boularias and Brahim Chaib-Draa. Bootstrapping apprenticeship learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[59] Wen Sun, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Dual policy iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[60] Parameswaran Kamalaruban, Rati Devidze, Volkan Cevher, and Adish Singla. Interactive teaching algorithms for inverse reinforcement learning. In *Proc. Intl Joint Conf. on Artificial Intelligence (IJCAI)*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The paper is organized exactly according to the contributions listed at the end of the introduction section.
   (b) Did you describe the limitations of your work? [Yes] Yes, we highlight the fact that the worst case guarantees of the robust MCE IRL can be worst than the standard MCE IRL, and describe how a good choice for α is crucial for better performance.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] This work presents a theoretical investigation of the imitation learning under transition dynamics mismatch (see Section 8). As such in the present form there are no direct negative societal impacts of our work. However, in future, when the proposed methods are applied on real-world systems, the practitioner has to be careful with the choice of α.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We confirm that our paper conforms with the ethics review guidelines.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] For example, see Theorem 2.
   (b) Did you include complete proofs of all theoretical results? [Yes] Complete proofs of all the theoretical results can be found in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and instructions are included in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We have provided all the training and hyperparameters details in the Experiments section, and in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] This can be seen in all the Figures.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We used an internal cluster with CPU nodes for the experiments; but we do not have an estimate of the total amount of compute.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]