
Maximum-Likelihood Inverse Reinforcement Learning with Finite-Time Guarantees

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Inverse reinforcement learning (IRL) aims to recover the reward function and
2 the associated optimal policy that best fits observed sequences of states and
3 actions implemented by an expert. Many algorithms for IRL have an inherently
4 nested structure: the inner loop finds the optimal policy given parametrized
5 rewards while the outer loop updates the estimates towards optimizing a measure
6 of fit. For high dimensional environments such nested-loop structure entails
7 a significant computational burden. To reduce the computational burden of a
8 nested loop, novel methods such as SQIL [1] and IQ-Learn [2] emphasize policy
9 estimation at the expense of reward estimation accuracy. However, without
10 accurate estimated rewards, it is not possible to do counterfactual analysis such
11 as predicting the optimal policy under different environment dynamics and/or
12 learning new tasks. In this paper we develop a novel *single-loop* algorithm for
13 IRL that does not compromise reward estimation accuracy. In the proposed
14 algorithm, each policy improvement step is followed by a stochastic gradient
15 step for likelihood maximization. We show that the proposed algorithm provably
16 converges to a stationary solution with a finite-time guarantee. If the reward is
17 parameterized linearly, we show the identified solution corresponds to the solution
18 of the maximum entropy IRL problem. Finally, by using robotics control problems
19 in MuJoCo and their transfer settings, we show that the proposed algorithm achieves
20 superior performance compared with other IRL and imitation learning benchmarks.

21 1 Introduction

22 Given observed trajectories of states and actions implemented by an expert, we consider the problem
23 of estimating the reinforcement learning environment in which the expert was trained. This problem is
24 generally referred to as inverse reinforcement learning (IRL) (see [3] for a recent survey). Assuming
25 the environment dynamics are known (or available online), the IRL problem consists of estimating
26 the reward function and the expert’s policy (optimizing such rewards) that best fits the data. While
27 there are limitations on the identifiability of rewards [4], the estimation of rewards based upon expert
28 trajectories enables important counterfactual analysis such as the estimation of optimal policies under
29 *different* environment dynamics and/or reinforcement learning of *new* tasks.

30 In the seminal work [5], the authors developed an IRL formulation, in which the model for the expert’s
31 behavior is the policy that maximizes entropy subject to a constraint requiring that the expected
32 features under such policy match the empirical averages in the expert’s observation dataset. The
33 algorithms developed for MaxEnt-IRL [5-7] have a nested loop structure, alternating between an
34 outer loop with a reward update step, and an inner loop that calculates the explicit policy estimates.

35 The computational burden of this nested structure is manageable in tabular environments, but it
36 becomes significant in high dimensional settings requiring function approximation.

37 Towards developing more efficient IRL algorithms, a number of works [8-11] propose to leverage
38 the idea of adversarial training [12]. These algorithms learn a non-stationary reward function through
39 training a discriminator, which is then used to guide the policy to match the behavior trajectories
40 from the expert dataset. However, [13] pointed out that the resulting discriminator (hence the
41 reward function) typically cannot be used in new learning tasks, since it is highly dependent on the
42 corresponding policy and current environment dynamics. Moreover, due to the brittle approximation
43 techniques and sensitive hyperparameter choice in the adversarial training, these IRL algorithms can
44 be unstable. [14, 15].

45 More recent works [1, 2] have developed algorithms to alleviate the computational burden of the
46 nested-loop training procedures. In [1], the authors propose to model the IRL using certain maximum
47 entropy RL problem with specific reward function (which assigns $r = +1$ for matching expert
48 demonstrations and $r = 0$ for all other behaviors). Then a soft Q imitation learning (SQIL) algorithm
49 is developed. In [2], the authors propose to transform the standard formulation of IRL (discussed
50 above) into a single-level problem, through learning a soft Q-function to implicitly represent the
51 reward function and the policy. An inverse soft-Q learning (IQ-Learn) algorithm is then developed,
52 which is shown to be effective in estimating the policy for the environment that it is trained on.
53 Despite being computationally efficient, IQ-Learn sacrifices the accuracy in estimating the rewards
54 since it indirectly recovers rewards from a soft Q-function approximator which is highly dependent
55 upon the environment dynamics and does not strictly satisfy the soft-Bellman equation. Therefore it
56 is not well-suited for counterfactual prediction or transfer learning setting.

57 Finally, in f -IRL [13] the authors consider an approach for estimating rewards based on the
58 minimization of several measures of divergence with respect to the expert’s state visitation measure.
59 The approach is limited to estimating rewards that only depend on state. Moreover, while the results
60 reported are based upon a single-loop implementation, the paper does not provide a convergence
61 guarantee to support performance. We refer the readers to Appendix A for other related works.

62 **Our Contributions.** The goal of this work is to develop an algorithm for IRL which is capable of
63 producing high-quality estimates of *both* rewards and behavior policies with finite-time guarantees.
64 The major contributions of this work are listed below.

65 • We consider a formulation of IRL based on maximum likelihood (ML) estimation over optimal
66 (entropy-regularized) policies, and prove that a strong duality relationship with maximum entropy
67 IRL holds if rewards are represented by a *linear* combination of features.¹ The ML formulation is
68 a *bi-level* optimization problem, where the upper-level problem maximizes the likelihood function,
69 while the lower-level finds the optimal policy under the current reward parameterization. Such a
70 bi-level structure is not only instrumental to the subsequent algorithm design, but is also flexible to
71 incorporate the use of state-only, as well as the regular reward function (which depends on the state
72 and action pair). The former is suitable for transfer learning since it is *insensitive* to the changes of
73 the environment dynamics, while the latter can be used to efficiently imitate the expert policy.

74 • Based on the ML-IRL formulation, we develop an efficient algorithm. To avoid the computational
75 burden of repeatedly solving the lower-level policy optimization problem, the proposed algorithm
76 has a single-loop structure where the policy improvement step and reward optimization step are
77 performed alternately so that each step can be performed relatively cheaply. Further, we show that
78 the algorithm has strong theoretical guarantees: to achieve certain ϵ -approximate stationary solution
79 for a non-linearly parameterized problem, it requires $\mathcal{O}(\epsilon^{-2})$ steps of policy and reward updates each.
80 To our knowledge, it is the first algorithm which has finite-time guarantee for the IRL problem under
81 nonlinear parameterization of reward functions.

¹Heuristic arguments for this duality result are discussed in [5] wherein the distribution of state-action paths is approximated (see equation (4) in [5]) and the equivalence between maximum entropy estimation and maximum likelihood (over the class of exponential distributions) [16] is invoked.

• We conduct extensive experiments to demonstrate that the proposed algorithm outperforms many state-of-the-art IRL algorithms in *both* policy estimation and reward recovery. In particular, when transferring to a new environment, RL algorithms using rewards recovered by the proposed algorithm outperform those that use rewards recovered from existing IRL and imitation learning benchmarks.

2 Preliminaries

In this section, we review the fundamentals of the maximum entropy inverse reinforcement learning (MaxEnt-IRL). We consider an MDP defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \eta, r, \gamma)$; \mathcal{S} and \mathcal{A} denote the state space and the action space respectively; $\mathcal{P}(s'|s, a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denotes the transition probability; $\eta(\cdot)$ denotes the distribution for the initial state; $r(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function and γ is a discount factor.

Given a dataset \mathcal{D} which contains trajectories $\tau = \{(s_t, a_t)\}_{t=0}^{\infty}$ sampled from an expert policy, the MaxEnt-IRL formulation [6, 17, 18] consists of finding a policy maximizing entropy subject to the expected features under such policy matching the empirical averages in the expert's observation dataset. Specifically, the MaxEnt-IRL formulation is given by:

$$\begin{aligned} \max_{\pi} \quad & H(\pi) := \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} -\gamma^t \log \pi(a_t | s_t) \right] \\ \text{s.t.} \quad & \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t) \right] = \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t) \right] \end{aligned} \quad (\text{MaxEnt-IRL})$$

where $\phi(s, a)$ is the feature vector of the state-action pair (s, a) . Let θ denote the dual variable for the linear constraint, then the Lagrangian of (MaxEnt-IRL) is given by

$$\mathcal{L}(\pi, \theta) := H(\pi) + \left\langle \theta, \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t) \right] - \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t) \right] \right\rangle. \quad (1)$$

In [6, 17, 18], the authors proposed a "dual descent" algorithm, which alternates between *i*) solving $\max_{\pi} \mathcal{L}(\pi, \theta)$ for fixed θ , and *ii*) a gradient descent step to optimize the dual variable θ . It is shown that the optimizer π_{θ}^* in step *i*) can be recursively defined as $\pi_{\theta}^*(a_t | s_t) = \frac{Z_{a_t | s_t, \theta}}{Z_{s_t, \theta}}$, where $\log Z_{a_t | s_t, \theta} = \phi(s_t, a_t)^T \theta + \gamma \mathbb{E}_{s_{t+1} \sim \mathcal{P}(\cdot | s_t, a_t)} [\log Z_{s_{t+1}, \theta}]$ and $\log Z_{s_t, \theta} = \log (\sum_{a \in \mathcal{A}} Z_{a | s_t, \theta})$.

From a computational perspective, the above algorithm is not efficient: it has a nested-loop structure, which repeatedly computes the optimal policy π_{θ}^* under each variable θ . It is known that when the underlying MDP is of high-dimension, such an algorithm can be computationally prohibitive [9, 10].

Recent work [2] proposed an algorithm called IQ-Learn to improve upon the MaxEnt-IRL by considering a saddle-point formulation:

$$\min_r \max_{\pi} \left\{ H(\pi) + \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t, a_t) \right] - \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t \cdot r(s_t, a_t) \right] \right\} \quad (2)$$

where $r(s_t, a_t)$ is the reward associated with state-action pair (s_t, a_t) . The authors show that this problem can be transformed into an optimization problem *only* defined in terms of the soft Q-function, which implicitly represents both reward and policy. IQ-Learn is shown to be effective in imitating the expert behavior while only relying on the estimation of the soft Q-function. However, the implicit reward estimate obtained is not necessarily accurate since its soft Q-function estimate depends on the environment dynamics and does not strictly satisfy the soft-Bellman equation. Hence, it is difficult to transfer the recovered reward function to new environments.

3 Problem Formulation

In this section, we consider a ML formulation of the IRL problem and formalize a duality relationship with maximum entropy-based formulation (MaxEnt-IRL).

117 Maximum Log-Likelihood IRL (ML-IRL)

118 Let $\mathcal{D} := \{\tau_i\}_{i=1}^N$ denote the dataset containing observed trajectories. A model of the expert's
 119 behavior is a randomized policy $\pi_\theta(\cdot|s)$, where π_θ is a specific policy corresponding to the reward
 120 parameter θ . Assuming the state dynamics $\mathcal{P}(s_{t+1}|s_t, a_t)$ are known, the discounted log-likelihood
 121 of observing a sample trajectory τ under model π_θ can be written follows:

$$\begin{aligned} \mathbb{E}_{\tau \sim \mathcal{D}} \left[\log \prod_{t \geq 0} (\mathcal{P}(s_{t+1}|s_t, a_t) \pi_\theta(a_t|s_t))^{\gamma^t} \right] &= \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t \geq 0} \gamma^t \log \pi_\theta(a_t|s_t) \right] \\ &\quad + \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t \geq 0} \gamma^t \log \mathcal{P}(s_{t+1}|s_t, a_t) \right]. \end{aligned}$$

122 Then we consider the following maximum log-likelihood IRL formulation:

$$\begin{aligned} \max_{\theta} \quad & L(\theta) := \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} \gamma^t \log \pi_\theta(a_t|s_t) \right] & (\text{ML-IRL}) \\ \text{s.t.} \quad & \pi_\theta := \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t; \theta) + \mathcal{H}(\pi(\cdot|s_t)) \right) \right], & (3a) \end{aligned}$$

123 where $r(s, a; \theta)$ is the reward function and $\mathcal{H}(\pi(\cdot|s)) := -\sum_{a \in \mathcal{A}} \pi(a|s) \log \pi(a|s)$.

124 We now make some remarks about ML-IRL. First, the problem takes the form of a *bi-level*
 125 optimization problem, where the *upper-level* problem (ML-IRL) optimizes the reward parameter
 126 θ , while the *lower-level* problem describes the corresponding policy π_θ as the solution to an
 127 entropy-regularized MDP ([19, 20]). In what follows we will leverage recently developed (stochastic)
 128 algorithms for bi-level optimization [21–23], that avoid the high complexity resulted from nested
 129 loop algorithms. Second, it is reasonable to use the ML function as the loss, because it searches
 130 for a reward function which generates a behavior policy that can best fit the expert demonstrations.
 131 While the ML function has been considered in [24, 25], they rely on heuristic algorithms with
 132 nested-loop computations to solve their IRL formulations, and the theoretical properties are not
 133 studied. Finally, the lower-level problem has been well-studied in the literature [19, 20, 26–28]. The
 134 entropy regularization in (3a) ensures the uniqueness of the optimal policy π_θ under the fixed reward
 135 function $r(s, a; \theta)$ [19, 26]. Even when the underlying MDP is high-dimensional and/or complex, the
 136 optimal policy could still be obtained; see recent developments in [19, 20]. We close this section by
 137 formally establishing a connection between (MaxEnt-IRL) and (ML-IRL).

138 **Theorem 1. (Strong Duality)** Suppose that the reward function is given as: $r(s, a; \theta) := \phi(s, a)^T \theta$,
 139 for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Then (ML-IRL) is the Lagrangian dual of (MaxEnt-IRL). Furthermore,
 140 strong duality holds, that is: $L(\theta^*) = H(\pi^*)$, where θ^* and π^* are the global optimal solutions for
 141 problems (ML-IRL) and (MaxEnt-IRL), respectively.

142 The proof of Theorem 1 is relegated to Appendix G. To our knowledge this result which specifically
 143 addresses the (MaxEnt-IRL) formulation is novel. Under finite horizon, a duality between ML
 144 estimation and maximum causal entropy is obtained in [17, Theorem 3]. However, the problem
 145 considered in that paper is not in RL nor IRL setting, therefore they cannot be directly used in the
 146 context of the present paper.

147 The above duality result reveals a strong connection between the two formulations under linear reward
 148 parameterization. Due to the duality result, we know that (ML-IRL) is a concave problem under
 149 linear reward parameterization. In this case, any stationary solution to (ML-IRL) is a global optimal
 150 estimator of the reward parameter.

151 4 The Proposed Algorithm

152 In this section, we design algorithms for (ML-IRL). Recall that one major drawback of algorithms
 153 for (MaxEnt-IRL) is that, they repeatedly solve certain policy optimization problem in the inner loop.
 154 Even though the recently proposed algorithm IQ-Learn [2] tries to improve the computational

Algorithm 1 *Maximum Likelihood Inverse Reinforcement Learning (ML-IRL)*

Input: Initialize reward parameter θ_0 and policy π_0 . Set the reward parameter's stepsize as α .
for $k = 0, 1, \dots, K - 1$ **do**
 Policy Evaluation: Compute $Q_{r_{\theta_k}, \pi_k}^{\text{soft}}(\cdot, \cdot)$ under reward function $r(\cdot, \cdot; \theta_k)$
 Policy Improvement: $\pi_{k+1}(\cdot|s) \propto \exp(Q_{r_{\theta_k}, \pi_k}^{\text{soft}}(s, \cdot)), \forall s \in \mathcal{S}$.
 Data Sampling I: Sampling expert trajectory $\tau_k^E := \{s_t, a_t\}_{t \geq 0}$ from the dataset D
 Data Sampling II: Sampling agent trajectory $\tau_k^A := \{s_t, a_t\}_{t \geq 0}$ from the policy π_{k+1}
 Estimating Gradient: $g_k := h(\theta_k; \tau_k^E) - h(\theta_k; \tau_k^A)$ where $h(\theta; \tau) := \sum_{t \geq 0} \gamma^t \nabla_{\theta} r(s_t, a_t; \theta)$
 Reward Parameter Update: $\theta_{k+1} := \theta_k + \alpha g_k$
end for

155 efficiency through implicitly representing the reward function and the policy by a Q-function
156 approximator, it has sacrificed the estimation accuracy of the recovered reward. Therefore, one
157 important goal of our design is to find provably efficient algorithms that can avoid high-complexity
158 operations and accurately recover the reward function. Specifically, it is desirable that the resulting
159 algorithm only uses a finite number of reward and policy updates to reach certain high-quality
160 solutions.

161 To proceed, we will leverage the special *bi-level* structure of the ML-IRL problem. The idea is to
162 alternate between one step of policy update to improve the solution of the lower-level problem, and
163 one step of the parameter update which improves the upper-level loss function. At each iteration k ,
164 given the current policy π_k and the reward parameter θ_k , a new policy π_{k+1} is generated from the
165 policy improvement step, and θ_{k+1} is generated by the reward optimization step.

166 This kind of alternating update is efficient, because there is no need to completely solve the policy
167 optimization subproblem, before updating the reward parameters. It has been used in many other
168 RL related settings as well. For example, the well-known actor-critic (AC) algorithm for policy
169 optimization [29, 30, 21] alternates between one step of policy update, and one step of critic parameter
170 update. Below we present the details of our algorithm at a given iteration k .

171 **Policy Improvement Step.** Let us consider optimizing the lower-level problem, when the reward
172 parameter θ_k is held fixed. Towards this end, define the so-called soft Q and soft value functions for a
173 given policy π_k and a reward parameter θ_k :

$$V_{r_k, \pi_k}^{\text{soft}}(s) = \mathbb{E}_{\pi_k} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t; \theta_k) + \mathcal{H}(\pi_k(\cdot|s_t)) \right) \middle| s_0 = s \right] \quad (4a)$$

$$Q_{r_k, \pi_k}^{\text{soft}}(s, a) = r(s, a; \theta_k) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s, a)} [V_{r_k, \pi_k}^{\text{soft}}(s')] \quad (4b)$$

174 We will adopt the well-known *soft policy iteration* [19] to optimize the lower-level problem [3a].
175 Under the current reward parameter θ_k and the policy π_k , the soft policy iteration generates a new
176 policy π_{k+1} as follows

$$\pi_{k+1}(a|s) \propto \exp(Q_{r_{\theta_k}, \pi_k}^{\text{soft}}(s, a)), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}. \quad (5)$$

177 Under a fixed reward function, it can be shown that the new policy π_{k+1} monotonically improves π_k ,
178 and it converges linearly to the optimal policy; see [19, Theorem 4] and [26, Theorem 1].

179 Note that in practice, we usually do not have direct access to the exact soft Q-function in (4b). In
180 order to perform the policy improvement, a few stochastic update steps in soft Q-learning [19] or soft
181 Actor-Critic (SAC) [20] could be used to replace the one-step soft policy iteration (5). In the appendix,
182 we present Alg. 2 to demonstrate such alternative implementation of our proposed algorithm.

183 **Reward Optimization Step.** We propose to use a stochastic gradient-type algorithm to optimize θ .
184 Towards this end, let us first derive the exact gradient $\nabla L(\theta)$. See Appendix D for detailed proof.

185 **Lemma 1.** *The gradient of the likelihood function $L(\theta)$ can be expressed as follows:*

$$\nabla L(\theta) = \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{t \geq 0} \gamma^t \nabla_{\theta} r(s_t, a_t; \theta) \right] - \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t \geq 0} \gamma^t \nabla_{\theta} r(s_t, a_t; \theta) \right]. \quad (6)$$

186 To obtain stochastic estimators of the exact gradient $\nabla L(\theta_k)$, we take two approximation steps: 1)
 187 approximate the optimal policy π_{θ_k} by π_{k+1} in (5), since the optimal policy π_{θ_k} is not available
 188 throughout the algorithm; 2) sample one expert trajectory τ_k^E from the dataset \mathcal{D} and one agent
 189 trajectory τ_k^A from the current policy π_{k+1} so to approximate the expectation operators in (6).

190 Following the approximation steps mentioned above, we construct a stochastic estimator g_k to
 191 approximate the exact gradient $\nabla L(\theta_k)$ in (6) as follows:

$$g_k := h(\theta_k; \tau_k^E) - h(\theta_k; \tau_k^A), \text{ where } h(\theta; \tau) := \sum_{t \geq 0} \gamma^t \nabla_{\theta} r(s_t, a_t; \theta). \quad (7)$$

192 With the stochastic gradient estimator g_k , the reward parameter θ_k is updated as:

$$\theta_{k+1} = \theta_k + \alpha g_k. \quad (8)$$

193 where α is the stepsize in updating the reward parameter.

194 In summary, the proposed algorithm for solving the ML-IRL problem (ML-IRL) is given in Alg. 1.

195 5 Theoretical Analysis

196 In this section, we present finite-time guarantees for the proposed algorithm.

197 To begin with, first recall that in Sec. 3 we have mentioned that (ML-IRL) is a bi-level problem,
 198 where the upper level (resp. the lower level) problem optimizes the reward parameter (resp. the
 199 policy). In order to solve (ML-IRL), our algorithm 1 has a *single-loop* structure, which alternates
 200 between one step of policy update and one step of the reward parameter update. Such a single-loop
 201 structure indeed has computational benefit, but it also leads to potential unstableness, since the lower
 202 level problem can stay far away from its true solutions. Specifically, at each iteration k , the potential
 203 unstableness is induced by the distribution mismatch between the policy π_{k+1} and π_{θ_k} , when we use
 204 estimator g_k (7) to approximate the exact gradient $\nabla L(\theta_k)$ (6) in updating the reward parameter θ_k .

205 Towards stabilizing the algorithm, we adopt the so-called *two-timescale* stochastic approximation
 206 (TTSA) approach [31, 21], where the lower-level problem updates in a faster time-scale (i.e.,
 207 converges faster) compared with its upper-level counterpart. Intuitively, the TTSA enables the
 208 π_{k+1} tracks the optimal π_{θ_k} , leading to a stable algorithm. In the proposed Algorithm 1 the policy
 209 (lower-level variable) is continuously updated by the soft policy iteration (5), and it is ‘fast’ because it
 210 converges linearly to the optimal policy under a fixed reward function [26, Theorem 1]. On the other
 211 hand, the reward parameter update (8) does not have such linear convergence property, therefore it
 212 works in a ‘slow’ timescale. To begin our analysis, let us first present a few technical assumptions.

213 **Assumption 1** (Ergodicity). *For any policy π , assume the Markov chain with transition kernel \mathcal{P} is*
 214 *irreducible and aperiodic under policy π . Then there exist constants $\kappa > 0$ and $\rho \in (0, 1)$ such that*

$$\sup_{s \in \mathcal{S}} \|\mathbb{P}(s_t \in \cdot | s_0 = s, \pi) - \mu_{\pi}(\cdot)\|_{TV} \leq \kappa \rho^t, \quad \forall t \geq 0$$

215 where $\|\cdot\|_{TV}$ is the total variation (TV) norm; μ_{π} is the stationary state distribution under π .

216 Assumption 1 assumes the Markov chain mixes at a geometric rate. It is a common assumption in the
 217 literature of RL [32, 33, 30], which holds for any time-homogeneous Markov chain with finite-state
 218 space or any uniformly ergodic Markov chain with general state space.

219 **Assumption 2.** *For any $s \in \mathcal{S}$, $a \in \mathcal{A}$ and any reward parameter θ , the following holds:*

$$\|\nabla_{\theta} r(s, a; \theta)\| \leq L_r, \quad (9a)$$

$$\|\nabla_{\theta} r(s, a; \theta_1) - \nabla_{\theta} r(s, a; \theta_2)\| \leq L_g \|\theta_1 - \theta_2\| \quad (9b)$$

220 where L_r and L_g are positive constants.

221 Assumption 2 assumes that the parameterized reward function has bounded gradient and is Lipschitz
 222 smooth. Such assumption in Lipschitz property are common in the literature of min-max / bi-level
 223 optimization [34, 21, 35, 23, 36].

224 Based on Assumptions 1-2 we next provide the following Lipschitz properties:

225 **Lemma 2.** Suppose Assumptions [1](#) - [2](#) hold. For any reward parameter θ_1 and θ_2 , the following
 226 results hold:

$$|Q_{r_{\theta_1}, \pi_{\theta_1}}^{\text{soft}}(s, a) - Q_{r_{\theta_2}, \pi_{\theta_2}}^{\text{soft}}(s, a)| \leq L_q \|\theta_1 - \theta_2\|, \quad \forall s \in \mathcal{S}, a \in \mathcal{A} \quad (10a)$$

$$\|\nabla L(\theta_1) - \nabla L(\theta_2)\| \leq L_c \|\theta_1 - \theta_2\| \quad (10b)$$

227 where $Q_{r_{\theta}, \pi_{\theta}}^{\text{soft}}(\cdot, \cdot)$ denotes the soft Q -function under the reward function $r(\cdot, \cdot; \theta)$ and the policy π_{θ} .
 228 The positive constants L_q and L_c are defined in Appendix [E](#).

229 The Lipschitz properties identified in Lemma [2](#) are vital for the convergence analysis. Then we
 230 present the main results, which show the convergence speed of the policy $\{\pi_k\}_{k \geq 0}$ and the reward
 231 parameter $\{\theta_k\}_{k \geq 0}$ in the Alg. [1](#). Please see Appendix [E](#) for the detailed proof.

232 **Theorem 2.** Suppose Assumptions [1](#) - [2](#) hold. Selecting stepsize $\alpha := \frac{\alpha_0}{K^\sigma}$ for the reward update step
 233 [\(8\)](#) where $\alpha_0 > 0$ and $\sigma \in (0, 1)$ are some fixed constants, and K is the total number of iterations to
 234 be run by the algorithm. Then the following result holds:

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} [\|\log \pi_{k+1} - \log \pi_{\theta_k}\|_{\infty}] = \mathcal{O}(K^{-1}) + \mathcal{O}(K^{-\sigma}) \quad (11a)$$

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} [\|\nabla L(\theta_k)\|^2] = \mathcal{O}(K^{-\sigma}) + \mathcal{O}(K^{-1+\sigma}) + \mathcal{O}(K^{-1}) \quad (11b)$$

235 where we denote $\|\log \pi_{k+1} - \log \pi_{\theta_k}\|_{\infty} := \max_{s \in \mathcal{S}, a \in \mathcal{A}} |\log \pi_{k+1}(a|s) - \log \pi_{\theta_k}(a|s)|$. In
 236 particular, setting $\sigma = 1/2$, then both quantities in [\(11a\)](#) and [\(11b\)](#) converge with the rate $\mathcal{O}(K^{-1/2})$.

237 In Theorem [2](#), we present the finite-time guarantee for the convergence of the Alg. [1](#). Moreover,
 238 as a special case, when the reward is parameterized as a linear function, we know that [\(ML-IRL\)](#)
 239 is concave and Theorem [2](#) provides a stronger guarantee which identify the global optimal reward
 240 estimator in finite time.

241 We provide a proof sketch below to present the key steps. The detailed proof is in Appendix [H](#).

242 *Proof sketch.* We outline our main steps in analyzing [\(11a\)](#) and [\(11b\)](#) respectively.

In order to show the convergence of policy estimates in [\(11a\)](#), there are several key steps. First,
 we note that both policies π_{k+1} and π_{θ_k} are in the softmax parameterization, where $\pi_{k+1}(\cdot|s) \propto$
 $\exp(Q_{r_{\theta_k}, \pi_k}^{\text{soft}}(s, \cdot))$ and $\pi_{\theta_k}(\cdot|s) \propto \exp(Q_{r_{\theta_k}, \pi_{\theta_k}}^{\text{soft}}(s, \cdot))$. Then, we can show a Lipschitz continuity
 property between the policy and the soft Q -function:

$$\|\log \pi_{k+1} - \log \pi_{\theta_k}\|_{\infty} \leq 2 \|Q_{r_{\theta_k}, \pi_k}^{\text{soft}} - Q_{r_{\theta_k}, \pi_{\theta_k}}^{\text{soft}}\|_{\infty},$$

243 where the infinity norm $\|\cdot\|_{\infty}$ is defined over the state-action space $\mathcal{S} \times \mathcal{A}$. Moreover, by analyzing
 244 the contraction property of the soft policy iteration [\(5\)](#), we bound $\|Q_{r_{\theta_k}, \pi_k}^{\text{soft}} - Q_{r_{\theta_k}, \pi_{\theta_k}}^{\text{soft}}\|_{\infty}$ as:

$$\|Q_{r_{\theta_k}, \pi_k}^{\text{soft}} - Q_{r_{\theta_k}, \pi_{\theta_k}}^{\text{soft}}\|_{\infty} \leq \gamma \|Q_{r_{\theta_{k-1}}, \pi_{k-1}}^{\text{soft}} - Q_{r_{\theta_{k-1}}, \pi_{\theta_{k-1}}}^{\text{soft}}\|_{\infty} + 2L_q \|\theta_k - \theta_{k-1}\|.$$

245 To ensure that the error term $\|\theta_k - \theta_{k-1}\|$ is small, we select the stepsize of reward parameters as
 246 $\alpha := \frac{\alpha_0}{K^\sigma}$, where K is the total number of iterations and $\sigma > 0$. Then, by combining previous two
 247 steps, we could further show the convergence rate of the policy estimates in [\(11a\)](#).

248 To prove the convergence of the reward parameters in [\(11b\)](#), we first leverage the Lipschitz smooth
 249 property of $L(\theta)$ in [\(10b\)](#). However, one technical challenge in the convergence analysis is how
 250 to handle the bias between the gradient estimator g_k defined in [\(7\)](#) and the exact gradient $\nabla L(\theta_k)$.
 251 When we construct the gradient estimator g_k in [\(7\)](#), we need to sample trajectories from the current
 252 policy π_{k+1} and the expert dataset \mathcal{D} . However, according to the expression of $\nabla L(\theta_k)$ in [\(6\)](#), the
 253 trajectories are sampled from the optimal policy π_{θ_k} and the dataset \mathcal{D} . Hence, there is a distribution
 254 mismatch between π_{k+1} and π_{θ_k} . Our key idea is to leverage [\(11a\)](#) to handle this distribution
 255 mismatch error, and thus show that the bias between g_k and $\nabla L(\theta_k)$ could be controlled. \square

256 To the best of our knowledge, Theorem [2](#) is the first non-asymptotic convergence result for IRL with
 257 nonlinear reward parameterization.

6 A Discussion over State-Only Reward

In this section we consider the IRL problems modeled by using rewards that are only a function of the state. A lower dimensional representation of the agent’s preferences (i.e. in terms only of states as opposed to states *and* actions) is more likely to facilitate counterfactual analysis such as predicting the optimal policy under different environment dynamics and/or learning new tasks. This is because the estimation of preferences which are only defined in terms of states is less sensitive to the specific environment dynamics in the expert’s demonstration dataset. Moreover, in application such as healthcare [37] and autonomous driving [38], where simply imitating the expert policy can potentially result in poor performance, since the learner and the expert may have different transition dynamics. Similar points have also been argued in recent works [13, 39–41].

Next, let us briefly discuss how we can understand (ML-IRL) and Alg 1, when the reward is parameterized as a state-only function. First, it turns out that there is an equivalent formulation of (ML-IRL), when the expert trajectories only contain the visited states.

Lemma 3. *Suppose the expert trajectories τ is sampled from a policy π^E , and the reward is parameterized as a state-only function $r(s; \theta)$. Then (ML-IRL) is equivalent to the following:*

$$\min_{\theta} \mathbb{E}_{s_0 \sim \eta(\cdot)} [V_{r_{\theta}, \pi_{\theta}}^{\text{soft}}(s_0)] - \mathbb{E}_{s_0 \sim \eta(\cdot)} [V_{r_{\theta}, \pi^E}^{\text{soft}}(s_0)] \quad (12a)$$

$$s.t. \pi_{\theta} := \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t; \theta) + \mathcal{H}(\pi(\cdot|s_t)) \right) \right]. \quad (12b)$$

Please see Appendix F for the detailed derivation. Intuitively, the above lemma says that, when dealing with the state-only IRL, (ML-IRL) minimizes the gap between the soft value functions of the optimal policy π_{θ} and that of the expert policy π^E . Moreover, Alg 1 can also be easily implemented with the state-only reward. In fact, the entire algorithm essentially stays the same, and the only change is that $r(s, a; \theta)$ will be replaced by $r(s; \theta)$. In this way, by only using the visited states in the trajectories, one can still compute the stochastic gradient estimator in (7). Therefore, even under the state-only IRL setting where the expert dataset only contains visited states, our formulation and the proposed algorithm still work if we parameterize the reward as a state-only function.

7 Numerical Results

In this section, we test the performance of our algorithm on a diverse collection of RL tasks and environments. In each experiment set, we train algorithms until convergence and average the scores of the trajectories over multiple random seeds. The hyperparameter settings and simulation details are provided in Appendix B.

MuJoCo Tasks For Inverse Reinforcement Learning. In this experiment set, we test the performance of our algorithm on imitating the expert behavior. We consider several high-dimensional robotics control tasks in MuJoCo [42]. Two class of existing algorithms are considered as the comparison baselines: 1) imitation learning algorithms that only learn the policy to imitate the expert, including Behavior Cloning (BC) [43] and Generative Adversarial Imitation Learning (GAIL) [10]; 2) IRL algorithms which learn a reward function *and* a policy simultaneously, including Adversarial Inverse Reinforcement Learning (AIRL) [11], f -IRL [13] and IQ-Learn [2]. To ensure fair comparison, all imitation learning / IRL algorithms use soft Actor-Critic [20] as the base RL algorithm. For the expert dataset, we use the data provided in the official implementation² of f -IRL.

In this experiment, we implement two versions of our proposed algorithm: ML-IRL(State-Action) where the reward is parameterized as a function of state and action; ML-IRL(State-Only) which utilizes the state-only reward function. In Table 1 we present the simulation results under a limited data regime where the expert dataset only contains a single expert trajectory. The scores (cumulative rewards) reported in the table is averaged over 6 random seeds. In each random seed, we train

²<https://github.com/twni2016/f-IRL>

algorithm from initialization and collect 20 trajectories to average their cumulative rewards after the algorithms converge. The results reported in Table 1 show that our proposed algorithms outperform the baselines. The numerical results with confidence intervals are in Table 3 (See Appendix).

We observe that BC fails to imitate the expert’s behavior. It is due to the fact that BC is based on supervised learning and thus could not learn a good policy under such a limited data regime. Moreover, we notice the training of IQ-Learn is unstable, which may be due to its inaccurate approximation to the soft Q-function. Therefore, in the MuJoCo tasks where IQ-Learn does not perform well, so that we cannot match the results presented in the original paper [2], we directly report results from there (and mark them by * in Table 1). The results of AIRL are not presented in Table 1 since it performs poorly even after spending significant efforts in parameter tuning (similar observations have been made in [44, 13]).

Task	BC	GAIL	IQ-Learn	f -IRL	ML-IRL (State-Only)	ML-IRL (State-Action)	Expert
Hopper	20.49	2815.59	2981.01	3074.55	3089.79	3121.68	3592.63
Half-Cheetah	-1.87	3301.52	4175.88	4375.88	4472.85	4086.92	5098.3
Walker	-14.01	1112.79	3961.42	4464.20	4380.17	4504.88	5344.21
Ant	760.46	1154.27	4362.90*	4571.71	4675.34	4984.34	5926.18
Humanoid	78.48	3016.40	5227.10*	5243.90	5390.31	5240.57	5351.08

Table 1: **MuJoCo Results.** The performance of benchmark algorithms under a single expert trajectory.

Transfer Learning Across Changing Dynamics. We further evaluate IRL algorithms on the transfer learning setting. We follow the environment setup in [11], where two environments with different dynamics are considered: Custom-Ant vs Disabled-Ant. We compare ML-IRL(State-Only) with several existing IRL methods: 1) AIRL [11], 2) f -IRL [13]; 3) IQ-Learn [2].

We consider two transfer learning settings: 1) data transfer; 2) reward transfer. For both settings, the expert dataset / trajectories are generated in Custom-Ant. In the data transfer setting, we train IRL agents in Disabled-Ant by using the expert trajectories, which are generated in Custom-Ant. In the reward transfer setting, we first use IRL algorithms to infer the reward functions in Custom-Ant, and then transfer these recovered reward functions to Disabled-Ant for further evaluation. In both settings, we also train SAC with the ground-truth reward in Disabled-Ant and report the scores.

The numerical results are reported in Table 2. the proposed ML-IRL(State-Only) achieves superior performance compared with the existing IRL benchmarks in both settings. We notice that IQ-Learn fails in both settings since it indirectly recovers the reward function from a soft Q-function approximator, which could be inaccurate and is highly dependent upon the environment dynamics. Therefore, the reward function recovered by IQ-Learn can not be disentangled from the expert actions and environment dynamics, which leads to its failures in the transfer learning tasks.

Setting	IQ-Learn	AIRL	f -IRL	ML-IRL(State-Only)	Groud-Truth
Data Transfer	-11.78	-5.39	188.85	221.51	320.15
Reward Transfer	-1.04	130.3	156.45	187.69	320.15

Table 2: **Transfer Learning.** The performance of benchmark algorithms under a single expert trajectory. The scores in the table are obtained similarly as in Table 1.

8 Conclusion

In this paper, we present a maximum likelihood IRL formulation and propose a provably efficient algorithm with a single-loop structure. To our knowledge, we provide the first non-asymptotic analysis for IRL algorithm under nonlinear reward parameterization. As a by-product, when we parameterize the reward as a state-only function, our algorithm could work in state-only IRL setting and enable reward transfer to new environments with different dynamics. Our algorithm outperforms existing IRL methods on high-dimensional robotics control tasks and corresponding transfer learning settings. A limitation of our method is the requirement for online training, so one future direction of this work is to further extend our algorithm and the theoretical analysis to the offline IRL setting.

References

- [1] S. Reddy, A. D. Dragan, and S. Levine, “SQIL: Imitation learning via reinforcement learning with sparse rewards,” in *International Conference on Learning Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=S1xKd24twB>
- [2] D. Garg, S. Chakraborty, C. Cundy, J. Song, and S. Ermon, “Iq-learn: Inverse soft-q learning for imitation,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [3] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, *An Algorithmic Perspective on Imitation Learning*, ser. Foundations and Trends in Robotics, 2018, vol. 7.
- [4] K. Kim, S. Garg, K. Shiragur, and S. Ermon, “Reward identification in inverse reinforcement learning,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 5496–5505.
- [5] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, “Maximum entropy inverse reinforcement learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.
- [6] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, “Modeling interaction via the principle of maximum causal entropy,” in *International Conference on Machine Learning*, 2010.
- [7] M. Wulfmeier, P. Ondruska, and I. Posner, “Maximum entropy deep inverse reinforcement learning,” *arXiv preprint arXiv:1507.04888*, 2015.
- [8] C. Finn, P. Christiano, P. Abbeel, and S. Levine, “A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models,” *arXiv preprint arXiv:1611.03852*, 2016.
- [9] C. Finn, S. Levine, and P. Abbeel, “Guided cost learning: Deep inverse optimal control via policy optimization,” in *International Conference on Machine Learning*. PMLR, 2016, pp. 49–58.
- [10] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [11] J. Fu, K. Luo, and S. Levine, “Learning robust rewards with adversarial inverse reinforcement learning,” *arXiv preprint arXiv:1710.11248*, 2017.
- [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [13] T. Ni, H. Sikchi, Y. Wang, T. Gupta, L. Lee, and B. Eysenbach, “f-irl: Inverse reinforcement learning via state marginal matching,” *arXiv preprint arXiv:2011.04709*, 2020.
- [14] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, “The gan landscape: Losses, architectures, regularization, and normalization,” 2018.
- [15] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson, “Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=Hk4fpoA5Km>
- [16] E. T. Jaynes, “Information theory and statistical mechanics,” *Physical review*, vol. 106, no. 4, p. 620, 1957.
- [17] B. D. Ziebart, J. A. Bagnell, and A. K. Dey, “The principle of maximum causal entropy for estimating interacting processes,” *IEEE Transactions on Information Theory*, vol. 59, no. 4, pp. 1966–1980, 2013.

- [18] M. Bloem and N. Bambos, “Infinite time horizon maximum causal entropy inverse reinforcement learning,” in *53rd IEEE conference on decision and control*. IEEE, 2014, pp. 4911–4916.
- [19] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, “Reinforcement learning with deep energy-based policies,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1352–1361.
- [20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 1861–1870.
- [21] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, “A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic,” *arXiv preprint arXiv:2007.05170*, 2020.
- [22] K. Ji, J. Yang, and Y. Liang, “Bilevel optimization: Convergence analysis and enhanced design,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 4882–4892.
- [23] P. Khanduri, S. Zeng, M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, “A near-optimal algorithm for stochastic bilevel optimization via double-momentum,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [24] V. Jain, P. Doshi, and B. Banerjee, “Model-free irl using maximum likelihood estimation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3951–3958.
- [25] N. Sanghvi, S. Usami, M. Sharma, J. Groeger, and K. Kitani, “Inverse reinforcement learning with explicit policy estimates,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9472–9480.
- [26] S. Cen, C. Cheng, Y. Chen, Y. Wei, and Y. Chi, “Fast global convergence of natural policy gradient methods with entropy regularization,” *Operations Research*, 2021.
- [27] S. Cayci, N. He, and R. Srikant, “Linear convergence of entropy-regularized natural policy gradient with linear function approximation,” *arXiv preprint arXiv:2106.04096*, 2021.
- [28] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, “Bridging the gap between value and policy based reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [29] V. Konda and J. Tsitsiklis, “Actor-critic algorithms,” *Advances in Neural Information Processing Systems*, vol. 12, 1999.
- [30] Y. F. Wu, W. Zhang, P. Xu, and Q. Gu, “A finite-time analysis of two time-scale actor-critic methods,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 617–17 628, 2020.
- [31] V. S. Borkar, “Stochastic approximation with two time scales,” *Systems & Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.
- [32] J. Bhandari, D. Russo, and R. Singal, “A finite time analysis of temporal difference learning with linear function approximation,” in *Conference on learning theory*. PMLR, 2018, pp. 1691–1692.
- [33] S. Zou, T. Xu, and Y. Liang, “Finite-sample analysis for sarsa with linear function approximation,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [34] C. Jin, P. Netrapalli, and M. Jordan, “What is local optimality in nonconvex-nonconcave minimax optimization?” in *International Conference on Machine Learning*. PMLR, 2020, pp. 4880–4889.

- [35] Z. Guan, T. Xu, and Y. Liang, “When will generative adversarial imitation learning algorithms attain global convergence,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1117–1125.
- [36] T. Chen, Y. Sun, and W. Yin, “Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 25 294–25 307, 2021.
- [37] C. Yu, J. Liu, S. Nemati, and G. Yin, “Reinforcement learning in healthcare: A survey,” *ACM Computing Surveys (CSUR)*, vol. 55, no. 1, pp. 1–36, 2021.
- [38] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, “Deep reinforcement learning for autonomous driving: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [39] T. Gangwani and J. Peng, “State-only imitation with transition dynamics mismatch,” in *International Conference on Learning Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=HJgLLyrYwB>
- [40] L. Viano, Y.-T. Huang, P. Kamalaruban, A. Weller, and V. Cevher, “Robust inverse reinforcement learning under transition dynamics mismatch,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [41] F. Torabi, G. Warnell, and P. Stone, “Generative adversarial imitation from observation,” *arXiv preprint arXiv:1807.06158*, 2018.
- [42] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [43] D. A. Pomerleau, “ALVINN: An autonomous land vehicle in a neural network,” *Advances in Neural Information Processing Systems*, vol. 1, 1988.
- [44] F. Liu, Z. Ling, T. Mu, and H. Su, “State alignment-based imitation learning,” in *International Conference on Learning Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=rylrdrxHFDdr>
- [45] H. Cao, S. Cohen, and L. Szpruch, “Identifiability in inverse reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [46] R. Wang, C. Ciliberto, P. V. Amadori, and Y. Demiris, “Random expert distillation: Imitation learning via expert policy support estimation,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6536–6544.
- [47] D. S. Brown, W. Goo, and S. Niekum, “Better-than-demonstrator imitation learning via automatically-ranked demonstrations,” in *Conference on robot learning*. PMLR, 2020, pp. 330–359.
- [48] P. Barde, J. Roy, W. Jeon, J. Pineau, C. Pal, and D. Nowrouzezahrai, “Adversarial soft advantage fitting: Imitation learning without policy optimization,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 334–12 344, 2020.
- [49] H. Kretschmar, M. Spies, C. Sprunk, and W. Burgard, “Socially compliant mobile robot navigation via inverse reinforcement learning,” *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.
- [50] C. Xia and A. El Kamel, “Neural inverse reinforcement learning in autonomous navigation,” *Robotics and Autonomous Systems*, vol. 84, pp. 1–14, 2016.

- [51] J. Rust and C. Phelan, “How social security and medicare affect retirement behavior in a world of incomplete markets,” *Econometrica: Journal of the Econometric Society*, pp. 781–831, 1997.
- [52] Z. Eckstein and K. I. Wolpin, “Why youths drop out of high school: The impact of preferences, opportunities, and abilities,” *Econometrica*, vol. 67, no. 6, pp. 1295–1339, 1999.
- [53] E. Duflo, R. Hanna, and S. P. Ryan, “Incentives work: Getting teachers to come to school,” *American Economic Review*, vol. 102, no. 4, pp. 1241–78, 2012.
- [54] H. Fang and Y. Wang, “Estimating dynamic discrete choice models with hyperbolic discounting, with an application to mammography decisions,” *International Economic Review*, vol. 56, no. 2, pp. 565–596, 2015.
- [55] L. Caliendo, M. Dvorkin, and F. Parro, “Trade and labor market dynamics: General equilibrium analysis of the china trade shock,” *Econometrica*, vol. 87, no. 3, pp. 741–835, 2019.
- [56] C. Cirillo, R. Xu, and F. Bastin, “A dynamic formulation for car ownership modeling,” *Transportation Science*, vol. 50, no. 1, pp. 322–335, 2016.
- [57] T. Xu, Z. Wang, and Y. Liang, “Improving sample complexity bounds for (natural) actor-critic algorithms,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 4358–4369, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? **[Yes]** See Section ??.
- Did you include the license to the code and datasets? **[No]** The code and the data are proprietary.
- Did you include the license to the code and datasets? **[N/A]**

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...

- (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? **[Yes]**
- (b) Did you describe the limitations of your work? **[Yes]**
- (c) Did you discuss any potential negative societal impacts of your work? **[Yes]**
- (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[Yes]**

2. If you are including theoretical results...

- (a) Did you state the full set of assumptions of all theoretical results? **[Yes]**
- (b) Did you include complete proofs of all theoretical results? **[Yes]**

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)? **[No]**
- (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[Yes]**

- 507 (c) Did you report error bars (e.g., with respect to the random seed after running
508 experiments multiple times)? [Yes]
- 509 (d) Did you include the total amount of compute and the type of resources used (e.g., type
510 of GPUs, internal cluster, or cloud provider)? [Yes]
- 511 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
- 512 (a) If your work uses existing assets, did you cite the creators? [Yes]
- 513 (b) Did you mention the license of the assets? [Yes]
- 514 (c) Did you include any new assets either in the supplemental material or as a URL? [No]
- 515 (d) Did you discuss whether and how consent was obtained from people whose data you're
516 using/curating? [N/A]
- 517 (e) Did you discuss whether the data you are using/curating contains personally identifiable
518 information or offensive content? [N/A]
- 519 5. If you used crowdsourcing or conducted research with human subjects...
- 520 (a) Did you include the full text of instructions given to participants and screenshots, if
521 applicable? [N/A]
- 522 (b) Did you describe any potential participant risks, with links to Institutional Review
523 Board (IRB) approvals, if applicable? [N/A]
- 524 (c) Did you include the estimated hourly wage paid to participants and the total amount
525 spent on participant compensation? [N/A]