Active Exploration for Inverse Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Inverse Reinforcement Learning (IRL) is a powerful paradigm for inferring a reward function from expert demonstrations. Many IRL algorithms require a known transition model and sometimes even a known expert policy, or they at least require access to a generative model. However, these assumptions are too strong for many real-world applications, where the environment can be accessed only through sequential interaction. We propose a novel IRL algorithm: **Ac**tive **e**xploration for **I**nverse **R**einforcement **L**earning (AceIRL), which actively explores an unknown environment and expert policy to quickly learn the expert's reward function and identify a good policy. AceIRL uses previous observations to construct confidence intervals that capture plausible reward functions and find exploration policies that focus on the most informative regions of the environment. AceIRL is the first approach to active IRL with sample-complexity bounds that does not require a generative model of the environment. AceIRL matches the sample complexity of active IRL with a generative model in the worst case. Additionally, we establish a problem-dependent bound that relates the sample complexity of AceIRL to the suboptimality gap of a given IRL problem. We empirically evaluate AceIRL in simulations and find that it significantly outperforms more naive exploration strategies.

1 Introduction

Reinforcement Learning (RL; Sutton and Barto, 2018) has achieved impressive results recently, from playing video games (Mnih et al., 2015) to solving robotic control problems (Haarnoja et al., 2019). However, in many applications, it is challenging to design a reward function that robustly describes the desired task (Amodei et al., 2016; Hendrycks et al., 2021). Instead of using an explicit reward function, Inverse Reinforcement Learning (IRL; Ng et al., 2000) seeks to recover the reward by observing an *expert*, e.g., an human who already knows how to perform a task. However, most existing IRL algorithms assume that the transition model, and in some cases, the expert's policy, are *known*. In many real-world applications, this is not given, and the agent needs to estimate the transition dynamics and the expert policy from samples. Figure 1 shows an illustrative example where the agent can choose between different paths that have different properties, e.g., walking speeds, and lead to different goals. The agent has to explore the environment and query the expert policy in order to infer the expert's reward function.

IRL with sample-based estimation was only recently analyzed formally by Metelli et al. (2021). They decompose the error on the reward into a contribution from estimating the transition model and estimating the expert policy. Based on this, Metelli et al. (2021) propose an efficient sampling strategy to recover a good reward function. However, they assume a *generative model* of the environment, i.e., the agent can query the transition dynamics for arbitrary states and actions. In practice, this

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

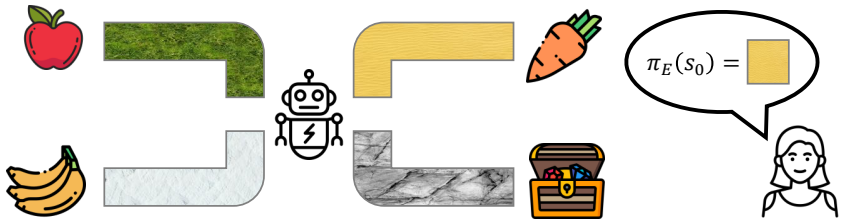

Figure 1: An illustrative example of the Active IRL problem. The agent can choose between four paths that lead to different objects. It can get action recommendations from an expert but does not know about the properties of the different paths (the transition dynamics) or the value of different items (the reward function). Observing the expert actions is not enough to infer a reward function. For example, from observing the expert recommending to take the yellow path, the agent cannot infer that the human prefers to find the carrot. The human might prefer to find the treasure but know it is too hard to reach. Therefore, the agent has to explore the environment and learn about its dynamics to infer a good reward function. AceIRL implements an exploration strategy that aims to infer which reward functions are consistent with the expert's recommendations as quickly as possible. We present experiments on a version of this environment in Section 7.

assumption is unrealistic. The agent in Figure 1 starts in the middle and cannot learn about the properties of any path without actually *exploring the environment*.

In this work, we consider IRL with unknown transition dynamics and expert policy and focus on exploring the environment in order to recover the expert's reward function efficiently. To the best of our knowledge, we present the first paper providing sample complexity guarantees for the active IRL problem without access to a generative model.

Our main contributions are:

- We propose the active IRL problem in a finite-horizon, undiscounted Markov Decision Process (MDP) and characterize necessary and sufficient conditions for solving it (Section 4.1).
- We analyze how the estimation errors of the transition model and the expert policy contribute to the estimation error of the reward function, extending prior work to the finite-horizon setting. We provide a novel analysis of how this error affects the performance of the policy, which optimizes the recovered reward function (Section 4.2).
- We propose a novel algorithm, **Ac**tive **e**xploration for **I**nverse **R**einforcement **L**earning (AceIRL), which actively explores the environment and the expert policy to infer a good reward function. In each iteration, AceIRL constructs an exploratory policy based on the estimation error of the recovered reward function (Section 6).
- We consider two different exploration strategies for AceIRL. The first, more straightforward, strategy provides a sample complexity similar to the algorithm proposed by Metelli et al. (2021), which has access to a generative model (Section 6.1). The second strategy takes the expected reduction in uncertainty into account (Section 6.2). This yields a tighter, problem-dependent sample complexity bound at the cost of solving a convex optimization problem in each iteration (Section 6.3).
- We evaluate AceIRL empirically in simulated environments and demonstrate that it achieves significantly better performance than move naive exploration strategies (Section 7).

The proofs of all results presented in the main paper can be found in Appendix B.

2 Related Work

Most IRL algorithms assume that the underlying transition model is known (Ratliff et al., 2006; Ziebart et al., 2008; Ramachandran and Amir, 2007; Levine et al., 2011). However, the transition model usually needs to be estimated from samples, which induces an error in the recovered reward function that most papers do not study. Metelli et al. (2021) analyze this error and the sample complexity of IRL in a tabular setting with a generative model. They propose an algorithm focused

on transferring the learned reward function to a fully known target environment. Dexter et al. (2021) provides a similar analysis in continuous state spaces and discrete action spaces, but they still require a generative model of the environment. In contrast, we *do not* assume access to a generative model and thus need to tackle the exploration problem in IRL.

Some prior work studies active learning algorithms for IRL in a Bayesian framework but without theoretical guarantees. Lopes et al. (2009) propose an active learning algorithm for IRL that estimates a posterior distribution over reward functions from demonstrations, requiring a prior distribution and full knowledge of the environment dynamics. Relatedly, Cohn et al. (2011) consider a Bayesian IRL setting with a semi-autonomous agent that asks an expert for advice if it is uncertain about the reward. Brown et al. (2018) empirically study active IRL in several safety-critical environments, selecting queries using value at risk. Kulick et al. (2013) consider active learning for a robotic manipulation task, asking a human expert for advice in situations with the highest predictive uncertainty. Similarly, Losey and O'Malley (2018) propose a method to learn uncertainty estimates from human corrections in a robotics context. All of these papers assume a Bayesian framework and do not provide theoretical guarantees. In contrast, our setup does not require a prior over reward functions, and we provide theoretical sample complexity guarantees for our algorithm.

3 Preliminaries

Let us first introduce necessary background and notation that we use throughout the paper.

Markov decision process. A finite-horizon (or episodic) Markov Decision Process without reward function (MDP\R) is a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, P, H, s_0)$, where $\mathcal{S}$ is the finite state space of size S; $\mathcal{A}$ is the finite action space of size A; $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is the transition model; H is the horizon and s_0 is the initial state. [1] In other words, a finite-horizon MDP\R is a finite-horizon MDP (Puterman, 2014) without the reward function. We describe an agent's behaviour with a (possible stochastic) policy $\pi \in \mathcal{S} \times [H] \to \Delta_{\mathcal{A}}$.

Reward function. A reward function $r : \mathcal{S} \times \mathcal{A} \times [H] \to [0, R_{\max}]$ maps state-action-time step triplets to a reward. Given an MDP\R $\mathcal{M}$ and a reward function r, we denote the resulting MDP by $\mathcal{M} \cup r$.

Value functions and optimality conditions. We define the *Q-function* $Q^{\pi,h}_{\mathcal{M}\cup r}(s, a)$ and *value-function* $V^{\pi,h}_{\mathcal{M}\cup r}(s)$ of a policy π in the MDP $\mathcal{M} \cup r$ at time step h, state s and action a as:

$$Q^{\pi,h}_{\mathcal{M}\cup r}(s,a) = r_h(s,a) + \sum_{s',a'} \pi_{h+1}(a'|s') P(s'|s,a) Q^{\pi,h+1}_{\mathcal{M}\cup r}(s',a'); \;\; V^{\pi,h}_{\mathcal{M}\cup r}(s) = \sum_a \pi_h(a|s) Q^{\pi,h}_{\mathcal{M}\cup r}(s,a)$$

We define the *advantage function* $A^{\pi,h}_{\mathcal{M}\cup r}(s, a)$ as $A^{\pi,h}_{\mathcal{M}\cup r}(s, a) = Q^{\pi,h}_{\mathcal{M}\cup r}(s, a) - V^{\pi,h}_{\mathcal{M}\cup r}(s)$. A policy π is optimal if $A^{\pi,h}_{\mathcal{M}\cup r}(s, a)$ is ≤ 0 for each time step $h \in [H]$, state $s \in \mathcal{S}$, action $a \in \mathcal{A}$. We denote the set of optimal policies for the MDP $\mathcal{M} \cup r$ with $\Pi^*_{\mathcal{M}\cup r}$.

State-visitation frequencies. We define $\eta^{h,h'}_{\mathcal{M},\pi}(s|s_0)$ as the probability of being in state s at time $h' \geq h$ following policy π in MDP\R $\mathcal{M}$ starting in state s_0 at time h. We can compute it recursively:

$$\eta^{h,h}_{\mathcal{M},\pi}(s'|s) := \mathbb{1}_{\{s'=s\}} \;\text{ and }\; \eta^{h,h'+1}_{\mathcal{M},\pi}(s'|s) := \sum_{s'',\tilde{a}} P(s'|s'',\tilde{a}) \pi_{h'}(\tilde{a}|s'') \eta^{h,h'}_{\mathcal{M},\pi}(s''|s).$$

We can define the visitation frequencies for state-action pairs analogously (see Appendix B).

4 Active Learning for Inverse Reinforcement Learning (Active IRL)

In this section, we first introduce the Active Inverse Reinforcement Learning problem with and without a generative model (Section 4.1). Then, we define the feasible reward set for finite-horizon MDPs (Section 4.2) and characterize the error propagation on the reward function and the value function (Section 4.3), extending results by Metelli et al. (2021) to the finite horizon setting.

[1] We can model any initial state distribution as a single initial state by modifying the transitions.

4.1 Problem Definition

Our goal is to design an exploration strategy to construct a dataset of demonstrations $\mathcal{D}$ such that an arbitrary IRL algorithm can recover a *good* reward function from it. To be agnostic to the choice of IRL algorithm, we consider the set of all feasible reward functions for a specific expert policy. Formally, we consider IRL problems $(\mathcal{M}, \pi^E)$ consisting of an MDP\R and an expert policy π^E, and we define the feasible reward set as follows.

Definition 1 (Feasible reward set)**.** *A reward function r is* feasible *for an IRL problem $(\mathcal{M}, \pi^E)$, if and only if the expert policy π^E is optimal in $\mathcal{M} \cup r$. We call the set of all feasible reward functions $\mathcal{R}_{\mathcal{M}\cup r}$ the* feasible reward set*. If we estimate the transition model and expert policy from samples we refer to the* recovered feasible set $\mathcal{R}_{\hat{\mathfrak{B}}} = \mathcal{R}_{\widehat{\mathcal{M}}\cup\hat{r}}$ *in contrast to the* exact feasible set $\mathcal{R}_{\mathfrak{B}} = \mathcal{R}_{\mathcal{M}\cup r}$.

Now, we can formalize the goal of Active IRL as finding a sampling strategy that satisfies the following PAC optimality criterion.

Definition 2 (Optimality criterion)**.** *Let $\mathscr{S}$ be a sampling strategy. Let $\mathcal{R}_{\mathfrak{B}}$ be the exact feasible set and $\mathcal{R}_{\hat{\mathfrak{B}}}$ be the feasible set recovered after observing $n \geq 0$ samples collected from $\mathcal{M}$ and π^E. We say that $\mathscr{S}$ is* (ϵ, δ, n)-correct *if with probability at least $1-\delta$ it holds that:*

$$\inf_{\hat{r}\in\mathcal{R}_{\hat{\mathfrak{B}}}} \sup_{\hat{\pi}^*\in\Pi^*_{\widehat{\mathcal{M}}\cup\hat{r}}} \max_{s,a,h} \left| Q^{\pi^*,h}_{\mathcal{M}\cup r}(s,a) - Q^{\hat{\pi}^*,h}_{\mathcal{M}\cup r}(s,a) \right| \leq \epsilon \quad \textit{for each } r \in \mathcal{R}_{\mathfrak{B}},$$

$$\inf_{r\in\mathcal{R}_{\mathfrak{B}}} \sup_{\pi^*\in\Pi^*_{\mathcal{M}\cup r}} \max_{s,a,h} \left| Q^{\pi^*,h}_{\mathcal{M}\cup r}(s,a) - Q^{\hat{\pi}^*,h}_{\mathcal{M}\cup r}(s,a) \right| \leq \epsilon \quad \textit{for each } \hat{r} \in \mathcal{R}_{\hat{\mathfrak{B}}},$$

where π^* is an optimal policy in $\mathcal{M} \cup r$ and $\hat{\pi}^*$ is an optimal policy in $\widehat{\mathcal{M}} \cup \hat{r}$.

The first condition states that for each reward in the exact feasible set, the best reward we could estimate in the recovered feasible set has a low error everywhere. This condition is a type of "recall": every possible true reward function needs to be captured by the recovered feasible set. The second condition ensures that there is a possible true reward function with a low error for every possible recovered reward function. This avoids an unnecessarily large recovered feasible set. This condition is a type of "precision": if we recover a reward function, it has to be close to a possible true reward function. Note, that Metelli et al. (2021) consider a similar optimality criterion in their Definition 5.1. However, they consider a known target environment; hence, our Definition 2 is a stronger requirement.

4.2 Feasible Rewards in Finite-horizon MDPs

Ng et al. (2000) characterize the feasible reward set implicitly in the infinite horizon setting, whereas Metelli et al. (2021) characterize it explicitly. Here, we provide similar results for a finite horizon.

Lemma 1 (Feasible Reward Set Implicit)**.** *A reward function r is feasible if and only if for all s, a, h it holds that: $A^{\pi,h}_{\mathcal{M}\cup r}(s,a) = 0$ if $\pi^E_h(a|s) \geq 0$ and $A^{\pi,h}_{\mathcal{M}\cup r}(s,a) \leq 0$ if $\pi^E_h(a|s) = 0$. Moreover, if the second inequality is strict, π^E is uniquely optimal, i.e., $\Pi^*_{\mathcal{M}\cup r} = \{\pi^E\}$.*

Lemma 2 (Feasible Reward Set Explicit)**.** *A reward function r is feasible if and only if there exists an $\{A_h \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}_{\geq 0}\}_{h\in[H]}$ and $\{V_h \in \mathbb{R}^{\mathcal{S}}\}_{h\in[H]}$ such that for all s, a, h it holds that:*

$$r_h(s,a) = -A_h(s,a)\mathbb{1}_{\{\pi^E_h(a|s)=0\}} + V_h(s) + \sum_{s'} P(s'|s,a)V_{h+1}(s')$$

Here, the **first term** ensures there is an advantage function for π^E and it is 0 for actions the expert takes and $A_h(s,a)$ for actions the expert does not take. The **second term** corresponds to reward-shaping by the value function.

4.3 Error Propagation

Next, we study the error propagation of estimating the transition model P with $\widehat{P}$ and the expert policy π^E with $\hat{\pi}^E$. In particular, we bound the estimation error on the reward as a function of the estimation errors of $\widehat{P}$ and $\hat{\pi}^E$, extending a result by Metelli et al. (2021) to the finite horizon setting.

Theorem 1 (Error Propagation)**.** *Let $(\mathcal{M}, \pi^E)$ and $(\widehat{\mathcal{M}}, \widehat{\pi}^E)$ be two IRL problems. Then, for any $r \in \mathcal{R}_{(\mathcal{M},\pi^E)}$ there exists $\widehat{r} \in \widehat{\mathcal{R}}_{(\widehat{\mathcal{M}},\widehat{\pi}^E)}$ such that:*

$$|r_h(s,a) - \widehat{r}_h(s,a)| \leq A_h(s,a)|\pi_h^E(a|s) - \widehat{\pi}_h^E(a|s)| + \sum_{s'} V_{h+1}(s')|P(s'|s,a) - \widehat{P}(s'|s,a)|$$

and we can bound $V_h \leq (H-h)R_{\max}$ and $A_h \leq (H-h)R_{\max}$.

In IRL, we cannot hope to recover the expert's reward function perfectly. Instead, we aim to estimate a reward function that leads to an optimal policy with performance close to the expert's policy under the (unknown) real reward function. For example, suppose a specific state s is difficult to reach in the environment. In that case, the error on the reward function $r(s,\cdot)$ will not impact the performance of the induced policy much. Formally, we are interested in studying the error propagation to the optimal value function. The next lemma will be crucial for analyzing this.

Lemma 3. *Let $\mathcal{M}$ be an MDP$\backslash$R, $r, \hat{r}$ two reward functions with optimal policies $\pi^*, \hat{\pi}^*$. Then,*

$$Q^{\pi^*,h}_{\mathcal{M}\cup r}(s,a) - Q^{\hat{\pi}^*,h}_{\mathcal{M}\cup r}(s,a) \leq \sum_{h'=h}^{H} \sum_{s',a'} \left(\eta^{h,h'}_{\mathcal{M},\pi^*}(s',a'|s,a) - \eta^{h,h'}_{\mathcal{M},\hat{\pi}^*}(s',a'|s,a)\right)(r_{h'}(s',a') - \hat{r}_{h'}(s',a'))$$

By combining this lemma with Theorem 4, we can decompose the error in the value function and Q-function into the error in estimating the transition model and the error in estimating the expert policy.

5 Recovering Feasible Rewards with a Generative Model

As a warmup, let us first study the sample complexity of a simple *uniform sampling* strategy with access to the generative model of $\mathcal{M}$. We assume we can query a generative model about a state-action pair (s,a) to receive a next state $s' \sim P(\cdot|s,a)$ and an expert action $a_E \sim \pi^E(\cdot|s)$. This allows us to introduce key ideas and serves as a baseline to compare later results to. We adapt the infinite-horizon results by Metelli et al. (2021) to the finite-horizon setting, and our stronger PAC requirement in Definition 2. We first discuss how we can estimate the transition model and the policy (Section 5.1) before stating the sample complexity of the uniform sampling strategy (Section 5.2).

5.1 Estimating Transition Model and Expert Policy

In each iteration k, let $n_k^h(s,a,s')$ be the number of times we observed the transitions (s,a,s) at time h up to iteration k. Also, we define $n_k^h(s,a) = \sum_{s'} n_k^h(s,a,s')$, and $n_k^h(s) = \sum_a n_k^h(s,a)$. Then we can estimate the transition model and expert policy by

$$\widehat{P}_k(s'|s,a) = \frac{\sum_{h=1}^{H} n_k^h(s,a,s')}{\max(1, \sum_{h=1}^{H} n_k^h(s,a))} \qquad \hat{\pi}^E_{k,h}(a|s) = \frac{n_k^h(s,a)}{\max(1, n_k^h(s))}.$$

In Appendix B we derive Hoeffding's confidence intervals for the transition model and the expert policy. Combining these with Theorem 4, we can compute the uncertainty on the recovered reward as:

$$C_k^h(s,a) = (H-h)R_{\max} \min\left(1, 2\sqrt{\frac{2\ell_k^h(s,a)}{n_k^h(s,a)}}\right),$$

where $\ell_k^h(s,a) = \log\left(24SAH(n_k^h(s,a))^2/\delta\right)$. We can show that for any pair of reward functions $r \in \mathcal{R}_{\mathfrak{B}}$ and $\hat{r} \in \mathcal{R}_{\hat{\mathfrak{B}}}$, the difference $|r_h(s,a) - \hat{r}_{k,h}(s,a)| \leq C_k^h(s,a)$. This uncertainty estimate will be a key component in all of our theoretical analysis.

5.2 Uniform Sampling Strategy

In each iteration k, the *uniform sampling* strategy allocates $n_{\max}$ samples uniformly over $[H] \times \mathcal{S} \times \mathcal{A}$. It estimates the reward uncertainty and stops as soon as $H \max_{h,s,a} C_k^h(s,a) \leq \epsilon$. The next theorem characterizes the sample complexity of uniform sampling with a generative model.

Algorithm 1 AceIRL algorithm for IRL in an unknown environment.

Input: significance $\delta \in (0,1)$, target accuracy ϵ, IRL algorithm $\mathscr{A}$, number of episodes N_E
Initialize $k \leftarrow 0$, $\epsilon_0 \leftarrow H/10$
while $\epsilon_k > \epsilon/4$ **do**
    Solve (convex) optimization problem (ACE) to obtain π_k
    Explore with policy π_k for N_E episodes, observing transitions and expert actions
    $k \leftarrow k+1$
    Update $\widehat{P}_k$, $\hat{\pi}_k$, C_k^h, and $\hat{r}_k \leftarrow \mathscr{A}(\mathcal{R}_{\hat{\mathfrak{B}}})$
    Update accuracy $\epsilon_k \leftarrow \max_a \hat{E}_k^0(s_0, a)$
end while
return Estimated reward function $\hat{r}_k$

Theorem 2 (Sample Complexity of Uniform Sampling IRL)**.** *The uniform sampling strategy fulfills Definition 2 with a number of samples upper bounded by:*

$$n \leq \tilde{\mathcal{O}}\Big(H^5 R_{\max}^2 SA/\epsilon^2\Big),$$

where $\mathcal{O}$ suppresses logarithmic terms.

This sample complexity bound appears slightly worse than the one in Metelli et al. (2021), who find $(1-\gamma)^{-4}$ which would translate to H^4. This is, however, due to the fact that we consider reward functions that can depend on the timestep h. If we assume the reward function does not depend on h, we gain a factor of H, obtaining the same result as Metelli et al. (2021).

6 Active Exploration for Inverse Reinforcement Learning

Let us now turn to our original problem: recovering the expert's reward function in an unknown environment *without* a generative model. This problem is harder since we need to create an exploration strategy to acquire the desired information about the environment. We now propose a novel sample-efficient exploration algorithm for IRL that we call **Ac**tive **e**xploration for **I**nverse **R**einforcement **L**earning (AceIRL). The algorithm takes inspiration from recent works on reward-free exploration (Kaufmann et al., 2021) and exploration strategies in RL (Auer et al., 2008). We divide the explanation of the algorithm into two parts. First, we introduce a simplified version of the algorithm, which comes with a problem independent sample complexity result (Section 6.1). Next, we introduce the full algorithm, which considers the expected reduction of uncertainty in the next iteration to improve exploration and maintains a confidence set of plausibly optimal policies to focus on the most relevant regions (Section 6.2). The full algorithm provides a tighter, problem-dependent sample complexity bound (Section 6.3). Algorithm 1 contains pseudo-code of AceIRL, and Appendix B contains the detailed theoretical analysis including proofs of all results discussed here.

6.1 Uncertainty-based Exploration for IRL

The first idea of AceIRL is similar to reward-free UCRL (Kaufmann et al., 2021). Our goal is to fulfill the PAC requirement in Definition 2. Hence, we start from an upper bound on the estimation error between the performance of the optimal policy $\hat{\pi}^*$ for a reward $\hat{r} \in \mathcal{R}_{\hat{\mathfrak{B}}}$ in the recovered feasible set and the optimal policy π^* for a reward function $r \in \mathcal{R}_{\mathfrak{B}}$ in the true MDP $\mathcal{M}$. We will then use this upper bound to drive the exploration. For each timestep h and iteration k, we define the error:

$$\hat{e}_k^h(s,a;\pi^*,\hat{\pi}^*) = \Big| Q_{\mathcal{M}\cup r}^{\pi^*,h}(s,a) - Q_{\mathcal{M}\cup r}^{\hat{\pi}^*,h}(s,a) \Big|. \tag{1}$$

We can define an upper bound on these errors recursively with $C_k^H(s,a) = 0$ and

$$E_k^h(s,a) = \min\Big((H-h)R_{\max}, C_k^h(s,a) + \sum_{s'} \widehat{P}(s'|s,a) \max_{a' \in \mathcal{A}} E_k^{h+1}(s',a')\Big). \tag{EB1}$$

It is straightforward to show that $\hat{e}_k^h(s,a;\pi^*,\hat{\pi}^*) \leq E_k^h(s,a)$ for any two policies $\pi^*, \hat{\pi}^*$. Using this error bound, we can introduce a simplified version of AceIRL that explores greedily with respect to $E_k^h(s,a)$. We call this algorithm "AceIRL Greedy". Note that this is equivalent to solving the

RL problem defined by $\mathcal{M} \cup C_k^h$; hence, we can use any RL solver to find the exploration policy in practice. If we explore with this greedy policy, we can stop if:

$$4 \max_a E_k^0(s_0, a) \leq \epsilon. \tag{SP1}$$

We can show that when this stopping condition holds, the solution fulfills the PAC requirement 2. Furthermore, we show in Appendix B that AceIRL Greedy achieves a sample complexity on order $\tilde{\mathcal{O}}\left(H^5 R_{\max}^2 SA/\epsilon^2\right)$, which matches the sample complexity of uniform sampling *with a generative model*. This is already a strong result implying that we do not need a generative model to achieve a good sample complexity for IRL. However, it turns out we can improve the algorithm further.

6.2 Problem Dependent Exploration

AceIRL Greedy is limited in two ways: (i) it explores states that have high uncertainty so far, whereas our goal is to reduce uncertainty *in the next iteration*, and (ii) it explores to reduce the uncertainty about all policies, whereas our goal is to reduce the uncertainty primarily about *plausibly optimal* policies. To address these limitations, we propose two modifications that yield the full AceIRL algorithm.

Reducing future uncertainty. The greedy policy w.r.t. E_k^h explores states in which the estimation error on the Q-functions is large. However, note that this is not exactly what we want, namely, to reduce the uncertainty the most. In particular, if we explore for more than one episode before updating the exploration policy, we should choose an exploration policy that considers how the uncertainty will reduce during exploration. Ideally, we would explore with a policy that minimizes E_{k+1}^h. However, we cannot compute this quantity exactly. Instead, we can approximate it using our current estimate of the transition model. Concretely, if we have an exploration policy π, we can estimate the reward uncertainty at the next iteration as:

$$\hat{C}_{k+1}^h(s,a) = (H-h) R_{\max} \min\left(1, 2\sqrt{\frac{2\ell_k^h(s,a)}{n_k^h(s,a) + \hat{n}_\pi^h(s,a)}}\right),$$

where $\hat{n}_\pi^h(s,a) = N_E \cdot \eta_{\mathcal{M},\pi}^{0,h}(s,a|s_0)$ is the expected number of times π visits (s,a) at time h and N_E is the number of episodes we will explore with π. We can use this estimate to find a policy that minimizes our estimate of E_k^{h+1}. While our original approach was akin to "uncertainty sampling", we now have a better way to measure the "informativeness" of choosing an exploration policy. This is a common pattern when designing query strategies in active learning (Settles, 2012). Note, that this argument does not rely on the IRL problem and can be used to independently improve algorithms for reward-free exploration (cf. Appendix D).

Focusing on plausibly optimal policies. By exploring greedily w.r.t. E_k^h, we reduce the estimation error of all policies. However, we are primarily interested in estimating the distance between policies $\pi^* \in \Pi^*_{\mathcal{M}\cup r}$ and $\hat{\pi}^* \in \Pi^*_{\mathcal{M}\cup \hat{r}}$ with $r \in \mathcal{R}_{\mathfrak{B}}$ and $\hat{r} \in \mathcal{R}_{\hat{\mathfrak{B}}}$. Of course, we do not know these sets, so we cannot use them directly to target the exploration. Instead, assume we can construct a set of plausibly optimal policies $\hat{\Pi}_k$ that contains all π^* and $\hat{\pi}_k^*$ with high probability. Then, we can redefine our upper bounds recursively as $\hat{E}_k^H(s,a) = 0$ and:

$$\hat{E}_k^h(s,a) = \min\Big((H-h)R_{\max}, C_k^h(s,a) + \sum_{s'} \widehat{P}(s'|s,a) \max_{\pi \in \hat{\Pi}_{k-1}} \pi(a'|s') \hat{E}_k^{h+1}(s',a')\Big), \tag{EB2}$$

In contrast to (EB1), we maximize over policies in $\hat{\Pi}_k$ rather than all actions. Acting greedily with respect to $\hat{E}_k^h(s,a)$ is equivalent to finding the optimal policy $\pi_k \in \hat{\Pi}_k$ for the RL problem defined by $\mathcal{M} \cup C_k^h$. To construct the set of plausibly optimal policies, we use an arbitrary IRL algorithm $\mathscr{A}$. We only assume that $\mathscr{A}$ will return a reward function $\hat{r}_k \in \mathcal{R}_{\hat{\mathfrak{B}}}$. Then, we can construct a set of plausibly optimal policies as $\hat{\Pi}_k = \{\pi | V^{*,}_{\widehat{\mathcal{M}}\cup\hat{r}_k}(s_0) - V^{\pi,}_{\widehat{\mathcal{M}}\cup\hat{r}_k}(s_0) \leq 10\epsilon_k\}$. We show in Appendix B that $\hat{\Pi}_k$ contains both π^* and $\hat{\pi}_k^*$ with high probability. This choice is based on ideas by Zanette et al. (2019).

We can define a stopping condition analogously to (SP1):

$$4 \max_a \hat{E}_k^0(s_0, a) \leq \epsilon. \tag{SP2}$$

Again, we can prove that if the algorithms stops due to (SP2), then $\mathcal{R}_{\hat{\mathfrak{B}}}$ respects Definition 2.

Implementing AceIRL. To implement the full algorithm, we need to solve an optimization problem:

$$\pi_k \in \operatorname*{argmin}_{\pi} \max_{\hat{\pi} \in \hat{\Pi}_{k-1}} \hat{E}^0_{k+1}(s_0, \hat{\pi}(s_0)) \tag{ACE}$$

The solution to this problem is the exploration policy that minimizes the uncertainty at the next iteration about plausibly optimal policies. This combines both modifications we just discussed. This problem might seem difficult to solve at first, but, perhaps surprisingly, it can be formulated as a convex optimization problem solvable with standard techniques (cf. Appendix B).

6.3 Sample Complexity of AceIRL

In this section, we present our main result about the sample complexity of AceIRL. The result is problem-dependent, and, in particular, depends on the advantage function $A^{*,h}_{\mathcal{M}\cup r}(s,a)$, where r is the reward function in the exact feasible set $\mathcal{R}_{\mathfrak{B}}$ closest to the reward function $\hat{r}_k$ which belongs to the estimated feasible set $\mathcal{R}_{\hat{\mathfrak{B}}}$. The advantage function $A^{*,h}_{\mathcal{M}\cup r}(s,a)$ acts similarly to a suboptimality gap: the closer the value of the second best action is to the best action, the harder it is to identify the best action and infer the correct reward function.

Theorem 3. *[AceIRL Sample Complexity] AceIRL returns a (ϵ, δ, n)-correct solution with*

$$n \leq \tilde{\mathcal{O}}\left(\min\left[\frac{H^5 R^2_{\max} SA}{\epsilon^2}, \frac{H^4 R^2_{\max} SA \epsilon^2_{\tau-1}}{\min_{s,a,h}(A^{*,h}_{\mathcal{M}\cup r}(s,a))^2 \epsilon^2}\right]\right)$$

where $\epsilon_{\tau-1}$ depends on the choice of N_E, the number of episodes of exploration in each iteration. $A^{*,h}_{\mathcal{M}\cup r}(s,a)$ is the advantage function of $r \in \operatorname{argmin}_{r\in\mathcal{R}_{\mathfrak{B}}} \max_{h,s,a}(r_h(s,a) - \hat{r}_{k,h}(s,a))$, the reward function from the feasible set $\mathcal{R}_{\mathfrak{B}}$ closest to the estimated reward function $\hat{r}_k$.

This result is the minimum of two terms. The first term is problem independent and it is achieved both by AceIRL Greedy and the full AceIRL. This bound matches the bound we saw previously with a generative model. Hence, AceIRL achieves the same results without access to the generative model. Using (ACE) can yield a better sample complexity, represented by the second term in the minimum. This bound depends on two main components: the ratio $\epsilon_{\tau-1}/\epsilon$ and the advantage function $A^{*,h}_{\mathcal{M}\cup r}(s,a)$. The ratio depends on the choice of N_E, the number of exploration episodes per iteration. If N_E is small, then the ϵ-ratio will be also small. If N_E is large the algorithm will perform similarly to a uniform sampling strategy. Appendix B provides the full proof of this theorem.

7 Experiments

We perform a series of simulation experiments to evaluate AceIRL. We simulate the expert policy using an underlying true reward function, and compare it to the recovered reward functions. Our main evaluation metric is a *normalized regret*:

$$\left(V^{\pi^*,0}_{\mathcal{M}\cup r}(s_0) - V^{\hat{\pi}^*,0}_{\mathcal{M}\cup r}(s_0)\right) / \left(V^{\pi^*,0}_{\mathcal{M}\cup r}(s_0) - V^{\bar{\pi}^*,0}_{\mathcal{M}\cup r}(s_0)\right),$$

where π^* is the optimal policy for $\mathcal{M}\cup r$, $\hat{\pi}^*$ is the optimal policy for $\widehat{\mathcal{M}}\cup\hat{r}$, and $\bar{\pi}^*$ is the worst possible policy for r, i.e., the optimal policy for $\mathcal{M}\cup(-r)$.

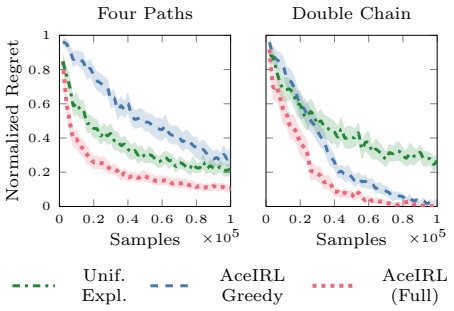

Figure 2: Normalized regret (lower is better) of the policy optimizing for the inferred reward in the estimated MDP as a function of the number of samples. The plots show the mean and 95% confidence intervals computed using 50 random seeds. We use $N_E = 100$.

We introduce the *Four Paths* environment shown in Figure 1, which consists of four chains of states that have different randomly sampled transition probabilities. One path has a goal with reward 1; all other rewards are 0. We also evaluate on *Random MDPs* with uniformly sampled transition dynamics and reward functions, the *Double Chain* environment proposed by Kaufmann et al. (2021), and the *Chain* and *Grid-world* environments proposed by Metelli et al. (2021). Appendix C provides details on the transition dynamics and rewards of all environments.

	Uniform sampling (gener. model)	TRAVEL (gener. model) (Metelli et al., 2021)	Random Exploration	AceIRL Greedy	AceIRL (Full)
Four Paths (Figure 1)	1900 ± 71		15580 ± 1954		
– $N_E = 50$		1600 ± 70		26660 ± 2315	$\mathbf{7720 \pm 907}$
– $N_E = 100$		2000 ± 0		43240 ± 2799	8880 ± 718
– $N_E = 200$		4000 ± 0		49120 ± 4053	14560 ± 1525
Double Chain (Kaufmann et al., 2021)	1980 ± 66		18980 ± 1776		
– $N_E = 50$		1120 ± 46		16320 ± 871	$\mathbf{10820 \pm 686}$
– $N_E = 100$		2000 ± 0		25200 ± 1418	14680 ± 1223
– $N_E = 200$		4000 ± 0		34720 ± 1921	18320 ± 1644
Metelli et al. (2021):					
Random MDPs ($N_E = 1$)	22 ± 1	23 ± 2	23 ± 1	$\mathbf{21 \pm 1}$	24 ± 1
Chain ($N_E = 1$)	78 ± 2	96 ± 3	155 ± 9	145 ± 9	$\mathbf{126 \pm 7}$
Gridworld ($N_E = 1$)	43 ± 2	38 ± 2	47 ± 2	$\mathbf{43 \pm 2}$	$\mathbf{44 \pm 2}$

Table 1: Sample complexity of AceIRL compared to random exploration and methods that use a generative model. We show the number of samples necessary to find a policy with normalized regret less than 0.4. We report means and standard errors computed over 50 random seeds each. For each environment, we highlight in **bold** the method that achieves the best performance without access to a generative model. If multiple methods are within one standard error distance, we highlight all of them. Overall, AceIRL is the most sample efficient method without a generative model if N_E is chosen small enough. In Appendix C, we show learning curves for all individual experiments.

We compare AceIRL and AceIRL Greedy to a uniformly random exploration policy, as a naive exploration strategy. Further, we consider uniform sampling with a generative model as well as TRAVEL (Metelli et al., 2021), which can be more sample efficient because they do not need to explore the environment. Note that TRAVEL is designed to learn a reward to be transferred to a known target environment. Instead, we use a modified version that uses the estimated MDP as a target. Appendix C provides more details on our implementations, and we provide source code in the supplemental material.

Table 1 shows the sample efficiency of all algorithms in all environments, measured as the number of samples needed to achieve a regret threshold of 0.4 (different thresholds yield similar conclusions; cf. Appendix C). AceIRL is the most sample efficient exploration strategy without access to a generative model; but the relative differences between the methods depend on the environment. In some cases, AceIRL even performs comparably to methods using a generative model, as the theory predicts.

In the *Four Paths* and *Double Chain* environments, we also vary the N_E parameter. AceIRL performs better for small values at the computational cost of updating the exploration policy more often. If N_E is too large, using AceIRL can be as bad as a uniformly random exploration policy. Increasing N_E hurts the performance of AceIRL Greedy more severely, which does not consider N_E explicitly. Figure 2 shows the normalized regret as a function of the number of samples in *Four Paths* and *Double Chain*. In both cases AceIRL performs best. However, AceIRL Greedy is worse than random exploration in the *Four Paths* environment. Hence, we find that the problem dependent exploration strategy of the full algorithm significantly improves the sample efficiency.

8 Conclusion

We considered active inverse reinforcement learning (IRL) with unknown transition dynamics and expert policy and introduced AceIRL, an efficient exploration strategy to learn about both the dynamic and the expert policy with the goal of inferring the reward function as efficiently as possible.

Our approach is a crucial step towards IRL algorithms with theoretical guarantees, but future work is needed to move to more practical settings. In particular, it would be interesting to extend the approach to continuous state and action spaces (e.g. using function approximation), and to obtain an efficient algorithm that does not require solving convex optimization problems. From a theoretical perspective, it would be useful to derive a lower bound on the sample complexity of the IRL problem, to understand if the IRL problem is inherently more difficult than usual RL. Beyond IRL, we believe that some of our methods could be useful for other settings, such as reward-free exploration (cf. Appendix D).

Sample efficient IRL is a promising way to apply RL in situations where there is no well-specified reward function available. Of course, even robust IRL algorithms pose a risk of misuse. But, we are optimistic that these methods will overall lead to safer RL systems that can be used in real applications.

References

D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *arXiv:1606.06565*, 2016.

P. Auer, T. Jaksch, and R. Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in Neural Information Processing Systems*, 2008.

D. S. Brown, Y. Cui, and S. Niekum. Risk-aware active inverse reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2018.

R. Cohn, E. Durfee, and S. Singh. Comparing action-query strategies in semi-autonomous agents. In *AAAI Conference on Artificial Intelligence*, 2011.

G. Dexter, K. Bello, and J. Honorio. Inverse reinforcement learning in a continuous state space with formal guarantees. In *Advances in Neural Information Processing Systems*, 2021.

T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine. Learning to walk via deep reinforcement learning. In *Proceedings of Robotics: Science and Systems (RSS)*, 2019.

D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt. Unsolved problems in ML safety. *arXiv:2109.13916*, 2021.

T. Jaksch, R. Ortner, and P. Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of International Conference on Machine Learning (ICML)*, 2002.

E. Kaufmann, P. Ménard, O. D. Domingues, A. Jonsson, E. Leurent, and M. Valko. Adaptive reward-free exploration. In *Algorithmic Learning Theory*, 2021.

J. Kulick, M. Toussaint, T. Lang, and M. Lopes. Active learning for teaching a robot grounded relational symbols. In *Proceedings of International Joint Conferences on Artificial Intelligence*, 2013.

S. Levine, Z. Popovic, and V. Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. *Advances in Neural Information Processing Systems*, 2011.

M. Lopes, F. Melo, and L. Montesano. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009.

D. P. Losey and M. K. O'Malley. Including uncertainty when learning from human corrections. In *Conference on Robot Learning (CoRL)*, 2018.

A. M. Metelli, G. Ramponi, A. Concetti, and M. Restelli. Provably efficient learning of transferable rewards. In *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

A. Y. Ng, S. J. Russell, et al. Algorithms for inverse reinforcement learning. In *Proceedings of International Conference on Machine Learning (ICML)*, 2000.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. In *Proceedings of International Joint Conferences on Artificial Intelligence*, 2007.

N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *Proceedings of International Conference on Machine Learning (ICML)*, 2006.

B. Settles. *Active Learning*. Morgan & Claypool Publishers, 2012.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

A. Zanette, M. J. Kochenderfer, and E. Brunskill. Almost horizon-free structure-aware best policy identification with a generative model. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2019.

B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, et al. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2008.

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] In Appendix B.
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] In the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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 provide an estimate in Appendix C.
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? [Yes] We use existing programming libraries that are mentioned in Appendix C.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide code to reproduce our experiments in the supplemental material.

(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]