Efficient Inverse Reinforcement Learning without Compounding Errors

Nicolas Espinosa Dice
ne229@cornell.edu
Department of Computer Science
Cornell University

Gokul Swamy
gswamy@cmu.edu
Robotics Institute
Carnegie Mellon University

Sanjiban Choudhury
sanjibanc@cornell.edu
Department of Computer Science
Cornell University

Wen Sun
ws455@cornell.edu
Department of Computer Science
Cornell University

Abstract

Inverse reinforcement learning (IRL) is an on-policy approach to imitation learning (IL) that allows the learner to observe the consequences of their actions at train-time. Accordingly, there are two seemingly contradictory desiderata for IRL algorithms: *(a)* preventing the compounding errors that stymie offline approaches like behavioral cloning and *(b)* avoiding the worst-case exploration complexity of reinforcement learning (RL). Prior work has been able to achieve either *(a)* or *(b)* but not both simultaneously. In our work, we first present a negative result showing that, without further assumptions, there are no efficient IRL algorithms that avoid compounding errors in the worst case. We then provide a positive result: under a novel structural condition we term *reward-agnostic policy completeness*, we prove that efficient IRL algorithms *do* avoid compounding errors, giving us the best of both worlds. We then address a practical constraint—the case of limited expert data—and propose a principled method for using sub-optimal data to further improve the sample-efficiency of IRL algorithms.

1 Introduction

Inverse reinforcement learning (IRL) is an on-policy approach to imitation learning that involves simultaneously learning a reward function from expert demonstrations and a policy that optimizes the learned reward (Ziebart et al., 2008a). IRL has been applied to a diverse set of applications, including robotics (Ratliff et al., 2007; Abbeel & Ng, 2008; Ratliff et al., 2009a; Silver et al., 2010; Zucker et al., 2011), autonomous driving (Bronstein et al., 2022; Igl et al., 2022; Vinitsky et al., 2022), and route finding (Ziebart et al., 2008a;b; Barnes et al., 2023).

While offline imitation learning approaches suffer from covariate shift between the training distribution (the expert's state distribution) and the test distribution (the learner's state distribution), interactive approaches like IRL allow the learner to roll-out its policy during train-time, effectively sampling states from the test distribution. This provides IRL with two concrete advantages over offline approaches like behavior cloning. First, IRL is more sample efficient, with respect to expert samples, than behavioral cloning (Swamy et al., 2022c). Second, IRL offers better error scaling, with respect to the horizon, than behavioral cloning (Ross & Bagnell, 2010; Swamy et al., 2021a; 2022c). In summary, for a fixed number of expert samples, IRL achieves a tighter performance gap with the expert policy compared to behavioral cloning.

However, the benefits of traditional IRL come at the cost of environment interactions. Because the reward function and policy are learned simultaneously, IRL requires policy optimization to be performed repeatedly, making it susceptible to the worst-case exploration complexity of reinforcement learning (RL) (Kakade, 2003; Swamy et al., 2023). Traditional IRL methods can require an exponential number of environment interactions in the worst case (Kakade, 2003; Swamy et al., 2023). To focus the exploration on useful states, prior work has leveraged the expert's state distribution, utilizing the fact that—by definition—the expert policy is optimal. Rather than reset the learner to the true starting state distribution, the learner is instead reset to states from the expert's demonstrations, resulting in an exponential decrease in interaction complexity (Swamy et al., 2023). We refer to this family of techniques as *efficient IRL*.

Unfortunately, the improvement of efficient IRL's interaction efficiency sacrifices traditional IRL's linear error scaling. For example, Swamy et al. (2023)'s Moment Matching by Dynamic Programming (MMDP) and No-Regret Moment Matching (NRMM) are exponentially faster than traditional IRL algorithms, but they suffer from quadratically compounding errors in the worst case. Intuitively, this is because a shift to a more off-policy approach (due to the expert resets) weakens the correlation between low training error and strong test time performance.

Based on the prior work, it seems that the two desiderata of IRL—interaction efficiency and avoidance of compounding errors—are contradictory, with algorithms only being able to attain one or the other. In our paper, we recognize that the commonly imposed assumption of *expert realizability* (i.e. the expert policy is within the learner's policy class) is insufficient to address both interaction efficiency and error scaling. *Our key insight is that, under a novel structural condition we call reward-agnostic policy completeness, IRL can both be efficient and avoid compounding errors.*

More explicitly, our contributions are as follows:

1. We first consider the *agnostic* setting, where no assumptions are made about the MDP's structure, and present a lower bound that shows it is impossible to learn a competitive policy with polynomial environment interaction complexity in the worst case. In other words, efficient IRL is not possible without assuming additional structure on the MDP.

2. We define a new structural condition, *reward-agnostic policy completeness*, under which efficient, reset-based IRL algorithms are capable of avoiding quadratically compounding errors. Importantly, our analysis holds for *approximate* policy completeness. Moreover, our condition does not require expert realizability, which is often an unrealistic assumption in practice.

3. We propose a principled method for incorporating sub-optimal data to improve the sample efficiency of IRL, and we prove the conditions under which it is beneficial. We provide a theoretical analysis of the common, practical setting of having limited expert (i.e. optimal) data but abundant quantities of sub-optimal data (e.g. poor teleoperation or imperfect driving).

2 Related Work

Reinforcement Learning. Prior work in reinforcement learning (RL) has examined leveraging exploration distributions to improve learning (Kakade & Langford, 2002; Bagnell et al., 2003; Ross et al., 2011). We adapt the Policy Search via Dynamic Programming (PSDP) algorithm of Bagnell et al. (2003) as our RL solver and leverage its performance guarantees in our analysis. Prior analyses of policy gradient RL algorithms—such as PSDP (Bagnell et al., 2003), Conservative Policy Iteration (CPI, Kakade & Langford (2002)), and Trust Region Policy Optimization (TRPO, Schulman et al. (2015))—use a *policy completeness* condition to establish a performance guarantee with respect to the *global*-optimal policy (Agarwal et al., 2019; Bhandari & Russo, 2024). In other words, policy completeness is used when comparing the learned policy to the optimal (i.e. best possible) policy and not simply the best policy in the policy class. We generalize the policy completeness condition from the RL setting with known rewards to the imitation learning setting with unknown rewards,

resulting in novel structural condition we term reward-agnostic policy completeness. Our paper also builds on work in statistically tractable agnostic RL (Jia et al., 2024). We use Jia et al. (2024)'s lower bound on agnostic RL with expert feedback to show why agnostic IRL is hard.

Imitation Learning. Our work examines the issue of distribution shift and compounding errors in IRL, which was introduced by Ross & Bagnell (2010). Ross et al. (2011)'s DAgger algorithm is capable of avoiding compounding errors but requires an interactive expert and *recoverability* (Rajaraman et al., 2021; Swamy et al., 2021a), which we do not assume in our setting.

Our algorithm and results are not limited to the tabular and linear MDP settings, differentiating it from prior work in efficient imitation learning (Xu et al., 2023; Viano et al., 2024). Our work relates to (Shani et al., 2022), who propose a Mirror Descent-based no-regret algorithm for online apprenticeship learning (OAL). We similarly use a mirror descent based update to our reward function, but differ from Shani et al. (2022)'s work by leveraging resets to expert and sub-optimal data to improve the interaction efficiency of our algorithm. Poiani et al. (2024) propose a technique of incorporating sub-optimal experts as a means of addressing the ambiguity in IRL problems, specifically the lack of uniqueness in reward functions that rationalize the observed behavior. In contrast, we do not use sub-optimal data in learning a reward function, instead using it to improve policy optimization.

Inverse Reinforcement Learning. We build upon Swamy et al. (2023)'s technique of leveraging the expert's state distribution for learner resets to speed up IRL. We make the following improvements to Swamy et al. (2023)'s work. First, we discard Swamy et al. (2023)'s assumptions of expert realizability and infinite expert data. Second, we demonstrate how to incorporate sub-optimal data into IRL. Third, we prove that our IRL algorithm avoids quadratically compounding errors efficiently under the approximate policy completeness condition. Swamy et al. (2023), in contrast, failed to show compounding error avoidance under expert realizability. Finally, our experiments focus on the setting of efficient IRL in environment where arbitrary learner resets are not possible.

3 Setup and Motivation

3.1 Problem Setup

Markov Decision Process. We consider a finite-horizon Markov Decision Process (MDP), $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P_h, r^*, H, \mu \rangle$ (Puterman, 2014). $\mathcal{S}$ and $\mathcal{A}$ are the state space and action space, respectively. $P = \{P_h\}_{h=1}^H$ is the time-dependent transition function, where $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ and Δ is the probability simplex. $r^* : \mathcal{S} \times \mathcal{A} \to [0,1]$ is the ground-truth reward function, which is unknown, but we assume $r^* \in \mathcal{R}$, where $\mathcal{R}$ is a class of reward functions such that $r : \mathcal{S} \times \mathcal{A} \to [0,1]$ for all $r \in \mathcal{R}$. H is the horizon, and $\mu \in \Delta(\mathcal{S})$ is the starting state distribution. Let $\Pi = \{\pi : \mathcal{S} \to \Delta(\mathcal{A})\}$ be the class of stationary policies. We assume Π and $\mathcal{R}$ are convex and closed. Let the class of non-stationary policies be defined by $\Pi^H = \{\pi_h : \mathcal{S} \to \Delta(\mathcal{A})\}_{h=1}^H$. A trajectory is given by $\tau = \{(s_h, a_h, r_h)\}_{h=1}^H$, where $s_h \in \mathcal{S}, a_h \in \mathcal{A}$, and $r_h = f(s_h, a_h)$ for some $f \in \mathcal{R}$. The distribution over trajectories formed by a policy is given by: $a_h \sim \pi(\cdot \mid s_h)$, $r_h = R_h(s_h, a_h)$, and $s_{h+1} \sim P_h(\cdot \mid s_h, a_h)$, for $h = 1, \ldots, H$. Let $d^\pi_{s_0,h}(s) = \mathbb{P}^\pi[s_h = s \mid s_0]$ and $d^\pi_{s_0}(s) = \frac{1}{H}\sum_{h=1}^H d^\pi_{s_0,h}(s)$. Overloading notation slightly, we have $d^\pi_\mu = \mathbb{E}_{s_0 \sim \mu} d^\pi_{s_0}$.

We index the value function by the reward function, such that for any $\pi \in \Pi^H$ and $r \in \mathcal{R}$, $V^\pi_{r,h}(s) := \mathbb{E}_{\tau \sim \pi}\left[\sum_{h'=h}^H r_{h'} \mid s_h = s\right]$, and $V^\pi_r = \mathbb{E}_{\tau \sim \pi}\sum_{h=1}^H r(s_h, a_h)$. We do a corresponding indexing for the advantage function. We will overload notation such that a state-action pair can be sampled from the visitation distributions, e.g. $(s,a) \sim d^\pi_\mu$ and $(s,a) \sim \rho_E$, as well as a state, e.g. $s \sim d^\pi_\mu$ and $s \sim \rho_E$. Note that by definition of d^π_μ, $\mathbb{E}_{\tau \sim \pi}\left[\sum_{h=1}^H r(s_t, a_t)\right] = H\,\mathbb{E}_{(s,a) \sim d^\pi_\mu}[r(s,a)]$.

Expert Policy. Much of the theoretical analysis in IRL and IL relies on a realizable expert policy (i.e. one that is within the learner's policy class Π) (Swamy et al., 2021a;b; 2022a; Xu et al., 2023; Kidambi et al., 2021; Ren et al., 2024). The assumption that the expert policy is realizable is often unrealistic in practical applications. Consider, for example, that humanoid robots cannot exactly

Algorithm 1 Reset-Based IRL (Dual, Swamy et al. (2023))

1: **Input:** Expert state-action distributions ρ_E, policy class Π, reward class $\mathcal{R}$
2: **Output:** Trained policy π
3: **for** $i = 1$ to N **do**
4: `// No-regret step over rewards`
5: $r_i \leftarrow \text{argmax}_{r \in \mathcal{R}} J(\pi_E, r) - J(\text{Unif}(\pi_{1:i}), r)$
6: `// Expert-competitive response by an RL algorithm`
7: $\pi_i \leftarrow \texttt{RL}(r = r_i, \rho = \rho_E)$
8: **end for**
9: **Return** π_N

replicate human movements because of differences in morphology (Zhang et al., 2024; He et al., 2024). Or, consider that experts may have access to privileged information that the learner does not (e.g. an autonomous vehicle's limited perception compared to a human driver's) (Swamy et al., 2022b).

In contrast to prior work, we do not impose the unrealistic assumption of expert realizability. Instead, we consider the *agnostic* setting, where the expert policy π_E is not necessarily in the policy class Π. A sample of the expert policy's trajectories are known. The dataset of state-action pairs sampled from the expert is $D_E = D_1 \cup D_2 \cup \ldots \cup D_H$, where $D_h = \{s_h, a_h\} \sim d_{\mu,h}^{\pi_E}$ and $|D_E| = N$. Let ρ_h be a uniform distribution over the samples in D_h, and ρ_E be a uniform distribution over the samples in D_E.

Goal of IRL. We cast IRL as a Nash equilibrium computation (Syed & Schapire, 2007; Swamy et al., 2021a). The ultimate objective of IRL is to learn a policy that matches expert performance. Because the ground-truth reward is unknown but belongs to the reward class, we aim to learn a policy that performs well under any reward function in the reward class. This is equivalent to finding the best policy under the *worst-case* reward (i.e. the reward function that maximizes the performance difference between the expert and learner). Formally, our goal is to learn a policy π such that

$$\min_{\pi \in \Pi} \max_{r \in \mathcal{R}} J(\pi_E, r) - J(\pi, r),$$

where $J(\pi, r) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{T} r(s_t, a_t)\right]$.

IRL Taxonomy. IRL algorithms consist of two steps: a reward update and a policy update. In the reward update, a *discriminator* is learned with the aim of differentiating the expert and learner trajectories. The policy is then optimized by an RL algorithm, with reward labels from the discriminator.

IRL algorithms can be classified into *primal* and *dual* variants (Swamy et al., 2021a), the latter of which we use in our paper. An example dual algorithm is shown in Algorithm 1. In dual-variant IRL algorithms, the discriminator is updated slowly via a no-regret step (Line 5), and the policy is updated via a best response (Line 7) (Ratliff et al., 2006; 2009b; Ziebart et al., 2008a; Swamy et al., 2021a). The RL subroutine in Line 7 uses the reward labels r and the learner's reset distribution ρ. In traditional IRL algorithms, the reset distribution remains the true starting state distribution (i.e. $\rho = \mu$). In efficient IRL algorithms, we perform an expert-competitive response (Swamy et al., 2023; Ren et al., 2024), rather than a best response, by changing the reset distribution to the expert's state distribution (i.e. $\rho = \rho_E$).

3.2 Agnostic IRL is Hard

Before introducing any conditions or assumptions, we start by considering the most general setting of IRL: the *agnostic* setting, where no assumptions are made about the MDP's structure, the policy class, or the expert's policy (i.e., we do not assume $\pi_E \in \Pi^H$).

Algorithm 2 Policy Search Via Dynamic Programming (Bagnell et al., 2003)

Input: Reward function r_i, reset distribution ρ, and policy class Π
Output: Trained policy π
for $h = H, H-1, \ldots, 1$ **do**
    Optimize

$$\pi_h \leftarrow \operatorname*{argmax}_{\pi' \in \Pi} \mathop{\mathbb{E}}_{s \sim \rho} \mathop{\mathbb{E}}_{a \sim \pi'(\cdot|s)} A_{r_i}^{\pi_{h+1}, \ldots, \pi_H}(s, a)$$

end for
Return $\pi = \{\pi_h\}_{h=1}^{H}$

Theorem 3.1 (Lower Bound on Agnostic RL with Expert Feedback (Jia et al., 2024))**.** *For any $H \in \mathbb{N}$ and $C \in [2^H]$, there exists a policy class Π with $|\Pi| = C$, expert policy $\pi_E \notin \Pi$, and a family of MDPs $\mathcal{M}$ with state space $\mathcal{S}$ of size $O(2^H)$, binary action space, and horizon H such that any algorithm that returns a 1/4-optimal policy must either use $\Omega(C)$ queries to the expert oracle $O_{exp} : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$, which returns $Q^{\pi_E}(s, a)$ (i.e. the Q value of expert policy π_E), or $\Omega(C)$ queries to a generative model*[1]*.*

From Theorem 3.1, we establish that polynomial sample complexity in the agnostic IRL setting, where $\pi_E \notin \Pi$, cannot be guaranteed. In other words, efficient IRL is not possible with no structure assumed on the MDP.[2]

4 Policy Complete Inverse Reinforcement Learning

The result from Section 3.2, which establishes that efficient IRL is not possible in the agnostic setting, begging the question:

under what conditions can efficient IRL algorithms avoid quadratically compounding errors?

Expert realizability, a commonly imposed assumption, fails to enable compound error avoidance (Swamy et al., 2023). Expert realizability requires the learner to perform the same actions as the expert policy, but as previously discussed, the learner may be constrained by its own morphology and not have access to some of the expert's actions. The learner may, nonetheless, be able to match expert performance through a different sequence of actions. We formalize this notion via an extension of *policy completeness*, a condition used in the analysis of policy gradient RL algorithms.

The policy completeness condition requires the learner have a way of improving the current policy's performance—without the requirement of matching actions with the optimal (i.e. expert) policy—if some improvement is possible. Importantly, the policy completeness condition of RL algorithms depends on the MDP's reward function, which in the imitation learning setting is unknown and is instead learned throughout training. We introduce *reward-agnostic policy completeness*, a generalization of policy completeness extended to the imitation learning setting.

Definition 4.1 (Reward-Indexed Policy Completeness Error)**.** *Given some expert state distribution ρ_E, MDP $\mathcal{M}$ with policy class Π and reward class $\mathcal{R}$, learned policy π_i, and learned reward function*

[1] A generative model allows the learner to query the transition and reward associated with a state-action pair on any state, differentiated from the online interaction model that can only play actions on states in sequence in a trajectory. For a more thorough discussion of their differences, see Jia et al. (2024).

[2] More specifically, Theorem 3.1 presents a lower bound on agnostic RL with expert feedback. It assumes access to the true reward function and an expert oracle, $O_{\text{exp}} : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$, which returns $Q^{\pi_E}(s, a)$ for a given state-action pair (s, a). The lower bound in Theorem 3.1 applies in the case where the expert oracle is replaced with a weaker expert action oracle (i.e. $\pi_E(s) : \mathcal{S} \to \mathcal{A}$) (Amortila et al., 2022; Jia et al., 2024). In agnostic IRL, we consider the even weaker setting of having a dataset of state-action pairs from the expert policy π_E. It should be noted that the classical importance sampling (IS) algorithm (Kearns et al., 1999) can be employed to find an approximately optimal policy in the agnostic setting, but it requires an exponential number of interactions (Agarwal et al., 2019; Jia et al., 2024).

r_i, define the reward-indexed policy completeness error of $\mathcal{M}$ to be

$$\epsilon_{\Pi}^{\pi_i, r_i} := \mathop{\mathbb{E}}_{s\sim\rho_E}\left[\max_{a\in\mathcal{A}} A_{r_i}^{\pi_i}(s,a)\right] - \max_{\pi'\in\Pi} \mathop{\mathbb{E}}_{s\sim\rho_E} \mathop{\mathbb{E}}_{a\sim\pi'(\cdot|s)}\left[A_{r_i}^{\pi_i}(s,a)\right].$$

We first present *reward-indexed policy completeness error*, which measures the policy class's ability to approximate the maximum possible advantage over the current policy. Intuitively, we can think of the second term as the learner's ability to improve the policy based on its policy class, and the first term as the maximum possible improvement.

The values and advantages are computed under current policy π_i and reward r_i, which represent an intermediate reward function and policy learned during IRL training. Note that the intermediate reward function r_i is not necessarily the ground-truth reward function r^*, so the expert policy may not be optimal under r_i. Consequently, to approximate the optimal improvement (i.e. advantage) over π_i, we consider a maximum over all possible actions, rather than sampling actions from the expert policy. In the worst case, where the policy class is poorly restricted under the expert's state distribution, then $\epsilon_{\Pi}^{\pi_i, r_i} = H$, since $r(s,a) \leq 1$ and $A(s,a) \leq H$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$, and $r \in \mathcal{R}$.

Recall that, at each iteration, IRL algorithms compute a policy and reward function (π_i and r_i, respectively) from the policy and reward classes (Π and $\mathcal{R}$, respectively). We measure the worst-case policy completeness error that can be attained during IRL training by adversarially selecting the learned policy and reward function.

Definition 4.2 (Reward-Agnostic Policy Completeness Error)**.** *Given some expert state distribution ρ_E and MDP $\mathcal{M}$ with policy class Π and reward class $\mathcal{R}$, define the reward-agnostic policy completeness error of $\mathcal{M}$ to be*

$$\begin{aligned}\epsilon_{\Pi} &:= \max_{\pi\in\Pi, r\in\mathcal{R}} \epsilon_{\Pi}^{\pi, r} \\ &= \max_{\pi\in\Pi, r\in\mathcal{R}} \left(\mathop{\mathbb{E}}_{s\sim\rho_E}\left[\max_{a\in\mathcal{A}} A_r^{\pi}(s,a)\right] - \max_{\pi'\in\Pi} \mathop{\mathbb{E}}_{s\sim\rho_E} \mathop{\mathbb{E}}_{a\sim\pi'(\cdot|s)}\left[A_r^{\pi}(s,a)\right] \right)\end{aligned}$$

Reward-agnostic policy completeness is therefore a measure of the policy class's ability to approximate the maximum possible advantage, over the expert's state distribution, under any reward function in the reward class. Note that $0 \leq \epsilon_{\Pi}^{\pi_i, r_i} \leq \epsilon_{\Pi} \leq H$ for any $\pi_i \in \Pi$, $r_i \in \mathcal{R}$. In the *approximate policy completeness* setting, we assume $\epsilon_{\Pi} = O(1)$.

4.1 Efficient IRL Under Approximate Policy Completeness

We present **GU**iding **I**mi**T**aters with **A**rbitrary **R**oll-ins (`GUITAR`), an efficient, reset-based IRL algorithm. The full IRL procedure is outlined in Algorithm 3. It can be summarized as (1) a no-regret reward update using Online Mirror Descent, and (2) an expert-competitive policy update using PSDP as the RL solver, where the learner is reset to a distribution ρ in the RL subroutine.

Existing efficient IRL algorithms, such as MMDP (Swamy et al., 2023), reset the learner exclusively to expert states (i.e. the case where $\rho = \rho_E$). We will focus on this setting first, and we will then consider the setting of resets to a mixture of expert and sub-optimal states (i.e. $\rho = \rho_{\text{mix}}$) in Section 5. In this section, we focus on the case of expert resets (i.e. $\rho = \rho_E$), and in Section 5, we discuss the case of resets to a mixture of expert and sub-optimal data (i.e. $\rho = \rho_{\text{mix}}$).

Policy Update. We employ PSDP (Bagnell et al., 2003) for the policy update step, shown in Algorithm 2, which performs an expert competitive response. We denote ρ as the reset distribution in PSDP. We consider resets to expert states ($\rho = \rho_E$) in this section. We then incorporate sub-optimal data into the reset distribution ($\rho = \rho_{\text{mix}}$) in Section 5.

Reward Update. We employ Online Mirror Descent (Nemirovskij & Yudin, 1983; Beck & Teboulle, 2003; Srebro et al., 2011) for the no-regret reward update. The reward function is updated via

$$r_i \leftarrow \operatorname*{argmax}_{r\in\mathcal{R}} \hat{L}(\pi_{i-1}, r) + \eta^{-1}\Delta_R(r \mid r_{i-1}),$$

Algorithm 3 GUiding **I**mi**T**aters with **A**rbitrary **R**oll-ins (`GUITAR`)

1: **Input:** Expert state-action distributions ρ_E, mixture of expert and offline state-action distributions ρ_{mix}, policy class Π, reward class $\mathcal{R}$
2: **Output:** Trained policy π
3: Set $\pi_0 \in \Pi$
4: **for** $i = 1$ to N **do**
5: Let
$$\hat{L}(\pi, r) = \mathop{\mathbb{E}}_{(s,a)\sim\rho_E} r(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^\pi} r(s,a)$$
6: Optimize
$$r_i \leftarrow \operatorname*{argmax}_{r\in\mathcal{R}} \hat{L}(\pi_{i-1}, r) + \eta^{-1}\Delta_R(r \mid r_{i-1}).$$
7: Optimize
$$\pi_i \leftarrow \texttt{PSDP}(r = r_i, \rho = \rho_{\text{mix}})$$
8: **end for**
9: **Return** π_i with lowest validation error

where Δ_R is the Bregman divergence with respect to the negative entropy function R. $\hat{L}(\pi, r)$ is the loss, defined by

$$\hat{L}(\pi, r) = \mathop{\mathbb{E}}_{(s,a)\sim\rho_E} r(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^\pi} r(s,a),$$

with respect to the distribution of expert samples, ρ_E.

4.2 Analysis in the Infinite-Sample Regime

For clarity, we first present the sample complexity of Algorithm 3 in the infinite expert sample regime (i.e., when we have infinite samples from the expert policy, so $\rho_E = d_\mu^{\pi_E}$). We present the bound in the finite sample regime in Section 5.2.

Theorem 4.3 (Sample Complexity of Algorithm 3)**.** *Consider the case of infinite expert data samples, such that $\rho_E = d_\mu^{\pi_E}$. Denote $\pi_i = (\pi_{i,1}, \pi_{i,2}, \ldots, \pi_{i,H})$ as the policy returned by ϵ-approximate PSDP at iteration $i \in [n]$ of Algorithm 3. Then,*

$$V^{\pi_E} - V^{\overline{\pi}} \leq \underbrace{H^2\epsilon}_{\text{policy optimization error}} + \underbrace{H\epsilon_\Pi}_{\text{policy completeness error}} + \underbrace{H\sqrt{\frac{\ln|\mathcal{R}|}{n}}}_{\text{reward regret}},$$

where H is the horizon, n is the number of outer-loop iterations of the algorithm, and $\overline{\pi}$ is the per-timestep average of the learned policies (i.e. π_i at each iteration $i \in [n]$).

The error is comprised of three terms. The first term, $H^2\epsilon$, stems from the policy optimization error of PSDP. It can be mitigated be improving the accuracy parameter ϵ of PSDP. Set to $\epsilon = \frac{1}{H}$, the term is reduced to linear error in the horizon H. This error can be interpreted as representing a tradeoff between environment interactions (i.e. computation) and error.

The second term, $H\epsilon_\Pi$, stems from the richness of the policy class. In the worst case where the policy class cannot approximate the maximum advantage, $\epsilon_\Pi = H$, resulting in quadratically compounding errors. Unlike the policy optimization error, the policy completeness error cannot be reduced with more environment interactions. Instead, it represents a fixed error that is a property of the MDP, the policy class, and the reward class. Under the approximate policy completeness setting, we assume $\epsilon_\Pi = O(1)$, reducing the error to linear in the horizon.

Finally, the last term $H\sqrt{\frac{\ln|\mathcal{R}|}{n}}$ stems from the regret of the Online Mirror Descent update to the reward function. By the no-regret property, we can reduce this term (to zero) by running more

outer-loop iterations of GUITAR. Assuming approximate policy completeness, such that $\epsilon_\Pi = O(1)$, Theorem 4.3 shows that quadratically compounding errors in the horizon can be avoided by setting a small accuracy parameter ϵ in the PSDP procedure.

In short, with sufficient iterations of Algorithm 3, GUITAR can avoid quadratically compounding errors under approximate policy completeness—notably, without relying on expert realizability.

5 Leveraging Sub-Optimal Data in IRL

Recall the two desiderata of IRL, which motivate our algorithm and results: (1) prevent compounding errors and (2) avoid the worst-case exploration complexity of RL. We accomplish the latter with learner resets to expert states and the former with the approximate policy completeness. In this section, we augment these theoretical motivations with common, practical constraints that significantly impact IRL performance.

First, much of the prior work in efficient IRL focuses on the infinite expert sample regime (Swamy et al., 2021a; 2022c; 2023; Ren et al., 2024), with some exceptions in the IL setting (Swamy et al., 2022c; Xu et al., 2023). This is often an unreasonable assumption to make in practice, where collecting expert data can be a resource-intensive process. Consider, for example, how resource-intensive the process of collecting expert data through robot teleoperation is (Fu et al., 2024). In this section, we consider the case of limited expert data and provide sample complexity bounds in this finite expert sample regime.

Second, in cases where collecting expert data is expensive and thus limited, there is often access to a larger source of offline, sub-optimal data. In this section, we describe how sub-optimal data can be leveraged in IRL. Specifically, we describe the conditions under which sub-optimal data is beneficial to efficient IRL's interaction efficiency.

5.1 Resetting to Sub-Optimal Data

In addition to the expert dataset, we have an offline dataset $D_{\text{off}} = \{s_i, a_i\}_{i=1}^M$, where $(s, a) \sim d_\mu^{\pi_B}$ and π_B is some behavior policy that is not necessarily as a high-quality as the expert π_E. We measure the overlap of π_B to the expert π_E using the standard concentrability coefficient: $C_B = \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty$. We show that we can gain benefit of using D_{off} as long as $C_B < \infty$ and the number of offline data points M is large. We define $D_{\text{mix}} = D_E \cup D_{\text{off}}$ and ρ_{mix} as the uniform distribution over D_{mix}. Let

$$\nu = \frac{N}{N+M} d_\mu^{\pi_E} + \frac{M}{N+M} d_\mu^{\pi_B}.$$

No change to the structure of Algorithm 3 is needed to incorporate sub-optimal data. Instead, we simply set PSDP's reset distribution to the mixture of sub-optimal and expert states, $\rho = \rho_{\text{mix}}$. The reward update remains the same,[3] and the approximate policy completeness condition remains $\epsilon_\Pi = O(1)$. The only modification to ϵ_Π is a change in the state distribution, replacing the distribution over expert samples, ρ_E, with the mixed distribution, ρ_{mix}.

5.2 Analysis in the Finite-Sample Regime

Next, we present the sample complexity bounds for GUITAR with sub-optimal data in the finite expert sample regime. For clarity, we present the case when PSDP's accuracy parameter is set to $\epsilon = 0$. (The $\epsilon > 0$ case follows Theorem 4.3's analysis.)

[3]IRL aims to learn a reward function such that the expert data under the learned reward function is optimal (Ziebart et al., 2008a; Swamy et al., 2023). Incorporating sub-optimal data into the discriminator update (i.e. the reward function) would result in the sub-optimal behavior being valued as optimal—an undesirable training outcome.

Theorem 5.1 (Sample Complexity of Algorithm 3)**.** *Suppose that PSDP's accuracy parameter is set to $\epsilon = 0$. Then, upon termination of Algorithm 3, with probability at least $1-\delta$, we have*

$$V^{\pi_E} - V^{\overline{\pi}} \leq \underbrace{H \min\left\{\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_{\Pi,\mathcal{R}}}{N}}, C_B\left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_{\Pi,\mathcal{R}}}{N+M}}\right)\right\}}_{\text{policy completeness error}} + \underbrace{H\sqrt{\frac{C_{\mathcal{R}}}{N}}}_{\text{statistical error of finite expert data}} + \underbrace{H\sqrt{\frac{\ln|\mathcal{R}|}{n}}}_{\text{reward regret}}$$

where H is the horizon, N is the number of expert state-action pairs, M is the number of offline state-action pairs, n is the number of reward updates, $C_{\Pi,\mathcal{R}} = \ln\frac{|\Pi||\mathcal{R}|}{\delta}$, $C_{\mathcal{R}} = \ln\frac{|\mathcal{R}|}{\delta}$, and $C_B = \left\|\frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}}\right\|_\infty$.

Theorem 5.1 upper bounds the sample complexity of Algorithm 3 in the sub-optimal data setting. The bound differs from Theorem 4.3 in the following ways. First, the policy optimization error term vanishes by setting $\epsilon = 0$. Importantly, the assumption of $\epsilon = 0$ is not necessary but rather convenient in simplifying the analysis. Moreover, the $\epsilon > 0$ case was presented in Theorem 4.3.

Second, we consider the finite expert sample regime, resulting in statistical error of estimating the expert policy's state distribution $d_\mu^{\pi_E}$ with the distribution over samples ρ_E.

Finally, we incorporate sub-optimal data into the reset distribution, resulting in a modified policy completeness error. We observe the condition under which sub-optimal data benefits learning is when

$$\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N}} < \left\|\frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}}\right\|_\infty \left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right).$$

In other words, it depends on the how well the sub-optimal data covers the expert data and the amount of expert and sub-optimal data. Intuitively, we can think of the coverage coefficient C_B as the "exchange rate," measuring how useful the sub-optimal data is in comparison to the expert data. When the sub-optimal data covers the expert data well, $C_B = \left\|\frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}}\right\|_\infty$ is small, so the sub-optimal data may be beneficial. Considering the special case where the "sub-optimal" data is collected from the expert policy π_E, then $C_B = \left\|\frac{d_\mu^{\pi_E}}{d_\mu^{\pi_E}}\right\|_\infty = 1$. The bound becomes equivalent to the case of having $N+M$ number of expert data samples. Because we only use the expert data for the reward update, rather than the sub-optimal data, the reward error terms remain the same.

In summary, we have demonstrated the following:

1. We consider the setting with finite expert data and differentiate between three sources of error: the policy completeness error, the statistical error from the finite expert samples, and the regret of the reward estimate.

2. We show the conditions under which sub-optimal data improves the sample efficiency of IRL.

3. With the RL solver's accuracy parameter set to $\epsilon = 0$, we establish a performance bound that is linear in the horizon under approximate policy completeness.

6 Discussion

We address the seemingly contradictory goals of preventing compounding errors in IRL and avoiding the worst-case exploration complexity of RL. We introduce a novel structural condition, reward-agnostic policy completeness, under which both compounding errors can be avoided efficiently. We then present a reset-based IRL algorithm and perform a finite-sample analysis. Finally, we identify the conditions under which sub-optimal data can be beneficial to the sample-efficiency of the algorithm. One direction for future work is generalizing our policy optimization step to other policy gradient algorithms beyond PSDP. Another direction is to empirically demonstrate the tradeoff between the coverage and amount of sub-optimal data in terms of IRL performance.

Broader Impact Statement

Our paper seeks to understand conditions under which efficient IRL works. Improving the efficiency of IRL can reduce computational costs, lessening the environmental impact of training IRL agents.

References

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*. ACM, 2008.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32, 2019.

Philip Amortila, Nan Jiang, Dhruv Madeka, and Dean P Foster. A few expert queries suffices for sample-efficient rl with resets and linear value approximation. *Advances in Neural Information Processing Systems*, 35:29637–29648, 2022.

James Bagnell, Sham M Kakade, Jeff Schneider, and Andrew Ng. Policy search by dynamic programming. *Advances in neural information processing systems*, 16, 2003.

Matt Barnes, Matthew Abueg, Oliver F Lange, Matt Deeds, Jason Trader, Denali Molitor, Markus Wulfmeier, and Shawn O'Banion. Massively scalable inverse reinforcement learning in google maps. *arXiv preprint arXiv:2305.11290*, 2023.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *Operations Research*, 2024.

Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, et al. Hierarchical model-based imitation learning for planning in autonomous driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8652–8659. IEEE, 2022.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

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

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

Maximilian Igl, Daewoo Kim, Alex Kuefler, Paul Mougin, Punit Shah, Kyriacos Shiarlis, Dragomir Anguelov, Mark Palatucci, Brandyn White, and Shimon Whiteson. Symphony: Learning realistic and diverse agents for autonomous driving simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2445–2451. IEEE, 2022.

Zeyu Jia, Gene Li, Alexander Rakhlin, Ayush Sekhari, and Nati Srebro. When is agnostic reinforcement learning statistically tractable? *Advances in Neural Information Processing Systems*, 36, 2024.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pp. 267–274, 2002.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning.* University of London, University College London (United Kingdom), 2003.

Michael Kearns, Yishay Mansour, and Andrew Ng. Approximate planning in large pomdps via reusable trajectories. *Advances in Neural Information Processing Systems*, 12, 1999.

Rahul Kidambi, Jonathan Chang, and Wen Sun. Mobile: Model-based imitation learning from observation alone. *Advances in Neural Information Processing Systems*, 34:28598–28611, 2021.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Riccardo Poiani, Gabriele Curti, Alberto Maria Metelli, and Marcello Restelli. Inverse reinforcement learning with sub-optimal experts. *arXiv preprint arXiv:2401.03857*, 2024.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

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

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Nived Rajaraman, Yanjun Han, Lin Yang, Jingbo Liu, Jiantao Jiao, and Kannan Ramchandran. On the value of interaction and function approximation in imitation learning. *Advances in Neural Information Processing Systems*, 34:1325–1336, 2021.

N. Ratliff, J. Bagnell, and M. Zinkevich. (semi-) autonomous navigation (san) using the maximum margin planning framework. In *Proceedings of Robotics: Science and Systems*. MIT Press, 2007.

N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *Proceedings of the 23rd International Conference on Machine learning*, pp. 729–736. ACM, 2009a.

Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 729–736, 2006.

Nathan D Ratliff, David Silver, and J Andrew Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27:25–53, 2009b.

Juntao Ren, Gokul Swamy, Zhiwei Steven Wu, J Andrew Bagnell, and Sanjiban Choudhury. Hybrid inverse reinforcement learning. *arXiv preprint arXiv:2402.08848*, 2024.

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

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

Lior Shani, Tom Zahavy, and Shie Mannor. Online apprenticeship learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8240–8248, 2022.

David Silver, J Andrew Bagnell, and Anthony Stentz. Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29 (12):1565–1592, 2010.

Nati Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. *Advances in neural information processing systems*, 24, 2011.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning*, pp. 10022–10032. PMLR, 2021a.

Gokul Swamy, Sanjiban Choudhury, J Andrew Bagnell, and Zhiwei Steven Wu. A critique of strictly batch imitation learning. *arXiv preprint arXiv:2110.02063*, 2021b.

Gokul Swamy, Sanjiban Choudhury, Drew Bagnell, and Steven Wu. Causal imitation learning under temporally correlated noise. In *International Conference on Machine Learning*, pp. 20877–20890. PMLR, 2022a.

Gokul Swamy, Sanjiban Choudhury, J Bagnell, and Steven Z Wu. Sequence model imitation learning with unobserved contexts. *Advances in Neural Information Processing Systems*, 35:17665–17676, 2022b.

Gokul Swamy, Nived Rajaraman, Matt Peng, Sanjiban Choudhury, J Bagnell, Steven Z Wu, Jiantao Jiao, and Kannan Ramchandran. Minimax optimal online imitation learning via replay estimation. *Advances in Neural Information Processing Systems*, 35:7077–7088, 2022c.

Gokul Swamy, David Wu, Sanjiban Choudhury, Drew Bagnell, and Steven Wu. Inverse reinforcement learning without reinforcement learning. In *International Conference on Machine Learning*, pp. 33299–33318. PMLR, 2023.

Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. *Advances in neural information processing systems*, 20, 2007.

Luca Viano, Stratis Skoulakis, and Volkan Cevher. Imitation learning in discounted linear mdps without exploration assumptions. *arXiv preprint arXiv:2405.02181*, 2024.

Eugene Vinitsky, Nathan Lichtlé, Xiaomeng Yang, Brandon Amos, and Jakob Foerster. Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. *Advances in Neural Information Processing Systems*, 35:3962–3974, 2022.

Tian Xu, Ziniu Li, Yang Yu, and Zhi-Quan Luo. Provably efficient adversarial imitation learning with unknown transitions. In *Uncertainty in Artificial Intelligence*, pp. 2367–2378. PMLR, 2023.

Chong Zhang, Wenli Xiao, Tairan He, and Guanya Shi. Wococo: Learning whole-body humanoid control with sequential contacts. *arXiv preprint arXiv:2406.06005*, 2024.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008a.

Brian D Ziebart, Andrew L Maas, Anind K Dey, and J Andrew Bagnell. Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior. In *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 322–331, 2008b.

Matt Zucker, Nathan Ratliff, Martin Stolle, Joel Chestnutt, J Andrew Bagnell, Christopher G Atkeson, and James Kuffner. Optimization and learning for rough terrain legged locomotion. *The International Journal of Robotics Research*, 30(2):175–191, 2011.

A Proofs of Section 4

A.1 Proof of Theorem 4.3

Proof. We consider the imitation gap of the expert and the average of the learned policies $\overline{\pi}$,

$$\begin{aligned}
V^{\pi_E} - V^{\overline{\pi}} &= \frac{1}{n}\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{\zeta\sim\pi_E}\sum_{h=1}^{H} r^*(s,a) - \mathop{\mathbb{E}}_{\zeta\sim\pi_i}\sum_{h=1}^{H} r^*(s,a)\right) \\
&= H\frac{1}{n}\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_E}} r^*(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_i}} r^*(s,a)\right) \\
&= H\frac{1}{n}\sum_{i=1}^{n} L(\pi_i, r^*) \\
&\le H\frac{1}{n}\max_{r\in\mathcal{R}}\sum_{i=1}^{n} L(\pi_i, r) \\
&\le H\frac{1}{n}\max_{r\in\mathcal{R}}\sum_{i=1}^{n}\left(L(\pi_i, r) - L(\pi_i, r_i) + L(\pi_i, r_i)\right) \\
&= H\frac{1}{n}\sum_{i=1}^{n} L(\pi_i, r_i) + H\frac{1}{n}\max_{r\in\mathcal{R}}\sum_{i=1}^{n}\left(L(\pi_i, r) - L(\pi_i, r_i)\right)
\end{aligned}$$

Applying the regret bound of Online Mirror Descent (Theorem E.2), we have

$$\begin{aligned}
V^{\pi_E} - V^{\overline{\pi}} &\le H\frac{1}{n}\sum_{i=1}^{n} L(\pi_i, r_i) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \\
&= H\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{H}\sum_{h=1}^{H}\mathop{\mathbb{E}}_{(s_h,a_h)\sim d_h^{\pi_E}} r_i(s_h,a_h) - \frac{1}{H}\sum_{h=1}^{H}\mathop{\mathbb{E}}_{(s_h,a_h)\sim d_h^{\pi_i}} r_i(s_h,a_h)\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\mathop{\mathbb{E}}_{s\sim\mu} V_{r_i}^{\pi_E} - \mathop{\mathbb{E}}_{s\sim\mu} V_{r_i}^{\pi_i}\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\left(\mathop{\mathbb{E}}_{(s_h,a_h)\sim d_h^{\pi_E}} A_{r_i,h}^{\pi_i}(s_h,a_h)\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \qquad (1)
\end{aligned}$$

Focusing on the interior summation, we have

$$\begin{aligned}
\sum_{h=0}^{H-1}\mathop{\mathbb{E}}_{(s_h,a_h)\sim d_h^{\pi_E}} A_h^{\pi_i}(s_h,a_h) &\le \sum_{h=0}^{H-1}\mathop{\mathbb{E}}_{s_h\sim d_h^{\pi_E}}\max_{a\in\mathcal{A}} A_h^{\pi_i}(s_h,a) \\
&= \sum_{h=0}^{H-1}\mathop{\mathbb{E}}_{s_h\sim d_h^{\pi_E}}\max_{a\in\mathcal{A}} A_h^{\pi_i}(s_h,a) - \epsilon_{\Pi,h} + \epsilon_{\Pi,h} \\
&= \sum_{h=0}^{H-1}\max_{\pi'\in\Pi}\mathop{\mathbb{E}}_{s_h\sim d_h^{\pi_E}}\mathop{\mathbb{E}}_{a\sim\pi'(\cdot|s)} A_h^{\pi_i}(s_h,a) + \epsilon_{\Pi,h} \\
&\le H^2\epsilon + H\epsilon_{\Pi,h} \qquad (2)
\end{aligned}$$

where the last line holds by PSDP's performance guarantee (Bagnell et al., 2003).

Applying (2) to (1), we have

$$\begin{aligned} V^{\pi_E} - V^{\overline{\pi}} &\leq \frac{1}{n} \sum_{i=1}^{n} \sum_{h=0}^{H-1} \left(\mathop{\mathbb{E}}_{(s_h, a_h) \sim d_h^{\pi_E}} A_{r_i, h}^{\pi_i}(s_h, a_h) \right) + H \sqrt{\frac{\ln |\mathcal{R}|}{n}} \\ &\leq \frac{1}{n} \sum_{i=1}^{n} \left(H^2 \epsilon + H \epsilon_{\Pi, h} \right) + H \sqrt{\frac{\ln |\mathcal{R}|}{n}} \\ &\leq H^2 \epsilon + H \epsilon_{\Pi} + H \sqrt{\frac{\ln |\mathcal{R}|}{n}} \end{aligned}$$

which completes the proof. □

B Proofs of Section 5

B.1 Lemmas of Theorem 5.1

Lemma B.1 (Reward Regret Bound)**.** *Recall that*

$$\hat{L}(\pi, r) = \mathop{\mathbb{E}}_{(s,a)\sim\rho_E} r(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^\pi} r(s,a).$$

Suppose that we update the reward via the Online Mirror Descent algorithm. Since $0 \le r(s,a) \le 1$ *for all* s, a*, then* $\sup_{\pi\in\Pi, r\in\mathcal{R}} \hat{L}(\pi, r) \le 1$*. Applying Theorem E.2 with* $B = 1$*, the regret is given by*

$$\begin{aligned} Reg_n &= \sup_{r\in\mathcal{R}} \frac{1}{n}\sum_{i=1}^{n} \hat{L}(\pi_i, r) - \frac{1}{n}\sum_{i=1}^{n} \hat{L}(\pi_i, r_i) \\ &\le \sqrt{\frac{2\ln|\mathcal{R}|}{n}} \\ &= \sqrt{\frac{C_1}{n}}, \end{aligned}$$

where $C_1 = 2\ln|\mathcal{R}|$ *and* n *is the number of updates.*

Lemma B.2 (Statistical Difference of Losses)**.** *With probability at least* $1-\delta$*,*

$$L(\pi, r) \le \hat{\mathrm{L}}(\pi, r) + \sqrt{\frac{C}{N}},$$

where $C = \ln\frac{2|\mathcal{R}|}{\delta}$ *and* N *is the number of state-action pairs from the expert.*

Proof. By definition of L and $\hat{\mathrm{L}}$, for any $\pi \in \Pi$ and $r \in \mathcal{R}$, we have

$$\begin{aligned} \left|L(\pi, r) - \hat{\mathrm{L}}(\pi, r)\right| &= \left| \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_E}} r(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi}} r(s,a) - \left(\mathop{\mathbb{E}}_{(s,a)\sim\rho_E} r(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi}} r(s,a) \right) \right| \\ &= \left| \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_E}} r(s,a) - \mathop{\mathbb{E}}_{(s,a)\sim\rho_E} r(s,a) \right| \\ &= \left| \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_E}} r(s,a) - \frac{1}{N}\sum_{(s_i,a_i)\in D_E}^{N} r(s_i,a_i) \right| \\ &\le \sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{R}|}{\delta}} \\ &\le \sqrt{\frac{C}{N}}, \end{aligned}$$

where $C = 4\ln\frac{2|\mathcal{R}|}{\delta}$. The fourth line holds by Hoeffding's inequality and a union bound. Specifically, we apply Corollary E.1 with $c = 1$, since all rewards are bounded by 0 and 1. We take a union bound over all reward functions in the reward class $\mathcal{R}$. Note that the terms involving π cancel out, so the union bound only applies to the reward function class $\mathcal{R}$. Rearranging terms gives the desired bound. □

Lemma B.3 (Advantage Bound)**.** *Suppose that* $\epsilon = 0$ *and reward function* r_i *are the input parameters to PSDP, and* $\pi_i = (\pi_1^i, \pi_2^i, \ldots, \pi_H^i)$ *is the output learned policy. Then, with probability at least* $1-\delta$*,*

$$\mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \le \min\left\{ \epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N}}, C_B\left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right)\right\}$$

where $C_B = \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty$, *$H$ is the horizon, N is the number of expert state-action pairs, M is the number of offline state-action pairs, and* $C_0 = 2 \ln \frac{|\Pi||\mathcal{R}|}{\delta}$.

Proof. Suppose that $\epsilon = 0$ is the input accuracy parameter to PSDP, and the advantages are computed under reward function r_i. PSDP is guaranteed to terminate and output a policy $\pi_i = (\pi_1^i, \pi_2^i, \ldots, \pi_H^i)$, such that

$$H\epsilon \geq \max_{\pi' \in \Pi} \mathop{\mathbb{E}}_{s_h \sim \rho_{mix,h}} \mathop{\mathbb{E}}_{a \sim \pi'(\cdot|s)} A_h^{\pi_i}(s_h, a)$$

for all $h \in [H]$ (Bagnell et al., 2003). Consequently, we have

$$\begin{aligned} H\epsilon &\geq \max_{\pi' \in \Pi} \mathop{\mathbb{E}}_{s \sim \rho_{\text{mix}}} \mathop{\mathbb{E}}_{a \sim \pi'(\cdot|s)} A^{\pi_i}(s, a) \\ &= \max_{\pi' \in \Pi} \mathop{\mathbb{E}}_{s \sim \rho_{\text{mix}}} \mathop{\mathbb{E}}_{a \sim \pi'(\cdot|s)} A^{\pi_i}(s, a) + \epsilon_{\Pi, r_i} - \epsilon_{\Pi, r_i} \\ &= \mathop{\mathbb{E}}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \epsilon_{\Pi, r_i} \end{aligned}$$

By definition, $0 \leq \epsilon_{\Pi, r_i} \leq \epsilon_\Pi$, so for any $x \in \mathbb{R}$, $x - \epsilon_{\Pi, r_i} \geq x - \epsilon_\Pi$, so

$$H\epsilon \geq \mathop{\mathbb{E}}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \epsilon_\Pi.$$

Rearranging the terms gives us

$$\begin{aligned} \mathop{\mathbb{E}}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) &\leq H\epsilon + \epsilon_\Pi \\ &= \epsilon_\Pi, \end{aligned} \tag{3}$$

where the last line holds by our assumption that $\epsilon = 0$.

Case 1: Jettison Offline Data We will first consider the case where offline data is useless, in which case we will focus on the expert data.

Note that $\max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \geq 0$ for all $s \in \mathcal{S}$ and $h \in [H]$. Applying the definition of ρ_{mix},

$$\mathop{\mathbb{E}}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) = \mathop{\mathbb{E}}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) + \mathop{\mathbb{E}}_{s \sim \rho_b} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a).$$

Consequently, we know that

$$\begin{aligned} \epsilon_\Pi &\geq \mathop{\mathbb{E}}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \\ &= \frac{1}{N} \sum_{s_i \in D_E}^{N} \max_{a \in \mathcal{A}} A^{\pi_i}(s_i, a) \end{aligned} \tag{4}$$

Because $\max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \geq 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we know $\max_{a \in \mathcal{A}} A^{\pi_i}(s_i, a) \leq \epsilon_\Pi$ for all $s_i \in D_E$. We apply Hoeffding's inequality (Corollary E.1) with $c = {\epsilon_\Pi}^2$ to bound the difference between $\mathbb{E}_{s \sim d^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a)$ and $\mathbb{E}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a)$. We apply a union bound on the policy and reward function. As stated previously, $\max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \geq 0$ for all $s \in \mathcal{S}$. By Hoeffding's inequality, with probability $1 - \delta$, we have

$$\begin{aligned} \left| \mathop{\mathbb{E}}_{s \sim d_\mu^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \mathop{\mathbb{E}}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \right| &= \left| \mathop{\mathbb{E}}_{s \sim d_\mu^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \frac{1}{N} \sum_{s_i \in D_E}^{N} \max_{a \in \mathcal{A}} A^{\pi_i}(s_i, a) \right| \\ &\leq \sqrt{\epsilon_\Pi^2 \frac{1}{2N} \ln \frac{|\Pi||\mathcal{R}|}{\delta}} \\ &\leq \epsilon_\Pi \sqrt{\frac{C_0}{N}}, \end{aligned}$$

where $C_0 = 2 \ln \frac{|\Pi||\mathcal{R}|}{\delta}$. Note that the cardinality of the set of advantage functions over all possible policies is upper bounded by the cardinalities of the policy and reward classes. Rearranging the terms and applying (4) yields

$$\mathop{\mathbb{E}}_{s \sim d_\mu^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s,a) \leq \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}.$$

Case 2: Leverage Offline Data Next, we consider the case where offline data is useful, specifically where there is good coverage of the expert data.

Next, we apply Hoeffding's inequality (Corollary E.1) to bound the difference between $\mathbb{E}_{s\sim\nu} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a)$ and $\mathbb{E}_{s\sim\rho_{\text{mix}}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a)$. We apply a union bound on the policy and reward function. We use $c = \epsilon_\Pi^2$ for a similar argument to the one used in Case 1. As stated previously, $\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \geq 0$ for all $s \in \mathcal{S}$. By Hoeffding's inequality, with probability $1-\delta$, we have

$$\begin{aligned}
\left| \mathop{\mathbb{E}}_{s\sim\nu} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) - \mathop{\mathbb{E}}_{s\sim\rho_{\text{mix}}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \right| &= \left| \mathop{\mathbb{E}}_{s\sim\nu} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) - \frac{1}{N+M} \sum_{s_i \in D_{mix}}^{N+M} \max_{a\in\mathcal{A}} A^{\pi_i}(s_i,a) \right| \\
&\leq \sqrt{\epsilon_\Pi \frac{1}{2(N+M)} \ln \frac{|\Pi||\mathcal{R}|}{\delta}} \\
&\leq \epsilon_\Pi \sqrt{\frac{C_0}{N+M}},
\end{aligned}$$

where $C_0 = 2 \ln \frac{|\Pi||\mathcal{R}|}{\delta}$. Note that the cardinality of the set of advantage functions over all possible policies is upper bounded by the cardinalities of the policy and reward classes. Rearranging the terms and applying (3) yields

$$\mathop{\mathbb{E}}_{s\sim\nu} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \leq \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}}. \tag{5}$$

By linearity of expectation, and using the fact that $1 \leq C_B < \infty$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) &= \frac{N}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + \frac{M}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \\
&\leq \frac{N}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + C_B \frac{M}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_B}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \\
&\leq C_B \frac{N}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + C_B \frac{M}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_B}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \\
&= C_B \left(\frac{N}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + \frac{M}{N+M} \mathop{\mathbb{E}}_{s\sim d^{\pi_B}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \right) \\
&\leq C_B \mathop{\mathbb{E}}_{s\sim\nu} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a).
\end{aligned} \tag{6}$$

Applying (6) to (5), we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) &\leq C_B \mathop{\mathbb{E}}_{s\sim\nu} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \\
&\leq C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right)
\end{aligned}$$

Final Result Using the bounds from Case 1 and Case 2, we know that

$$\mathop{\mathbb{E}}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \leq \min \left\{ \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}, C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right) \right\}$$

where $C_B = \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty$, H is the horizon, N is the number of expert state-action pairs, M is the number of offline state-action pairs, and $C_0 = 2 \ln \frac{|\Pi||\mathcal{R}|}{\delta}$. □

Lemma B.4 (Loss Bound)**.** *Suppose that $\epsilon = 0$ and reward function r_i are the input parameters to PSDP, and $\pi_i = (\pi_1^i, \pi_2^i, \ldots, \pi_H^i)$ is the output learned policy. Then, with probability at least $1 - \delta$,*

$$\hat{\mathrm{L}}(\pi_i, r_i) \leq \min \left\{ \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}, C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right) \right\} + \sqrt{\frac{C}{N}},$$

where $C_B = \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty$, H is the horizon, N is the number of expert state-action pairs, M is the number of offline state-action pairs, $C_0 = 2 \ln \frac{|\Pi||\mathcal{R}|}{\delta}$, and $C = \ln \frac{2|\mathcal{R}|}{\delta}$.

Proof. By Lemma B.2, we have

$$\begin{aligned}
\hat{\mathrm{L}}(\pi_i, r_i) &\leq L(\pi_i, r_i) + \sqrt{\frac{C}{N}} \\
&= \mathop{\mathbb{E}}_{(s,a) \sim d_\mu^{\pi_E}} [r_i(s,a)] - \mathop{\mathbb{E}}_{(s,a) \sim d_\mu^{\pi_i}} [r_i(s,a)] + \sqrt{\frac{C}{N}} \\
&= \frac{1}{H} \left(V_{r_i}^{\pi_E} - V_{r_i}^{\pi_i} \right) + \sqrt{\frac{C}{N}} \\
&= \frac{1}{H} \left(\sum_{h=1}^{H} \mathop{\mathbb{E}}_{(s_h, a_h) \sim d_h^{\pi_E}} A_{r_i, h}^{\pi_i}(s_h, a_h) \right) + \sqrt{\frac{C}{N}} \\
&\leq \frac{1}{H} \left(\sum_{h=1}^{H} \mathop{\mathbb{E}}_{s_h \sim d_h^{\pi_E}} \max_{a \in \mathcal{A}} A_{r_i, h}^{\pi_i}(s_h, a) \right) + \sqrt{\frac{C}{N}} \\
&= \frac{1}{H} \left(H \mathop{\mathbb{E}}_{s \sim d^{\pi_E}} \max_{a \in \mathcal{A}} A_{r_i}^{\pi_i}(s, a) \right) + \sqrt{\frac{C}{N}}
\end{aligned}$$

where $C = \ln \frac{2|\mathcal{R}|}{\delta}$. The second line holds by the definition of $L(\pi_i, r_i)$, and the third line holds by the definition of the reward-indexed value function. The fourth line holds by the Performance Difference Lemma (PDL). Applying Lemma B.3, we have

$$\hat{\mathrm{L}}(\pi_i, r_i) \leq \min \left\{ \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}, C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right) \right\} + \sqrt{\frac{C}{MN}},$$

where $C_B = \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty$, H is the horizon, N is the number of expert state-action pairs, M is the number of offline state-action pairs, $C_0 = 2 \ln \frac{|\Pi||\mathcal{R}|}{\delta}$, and $C = \ln \frac{2|\mathcal{R}|}{\delta}$. □

B.2 Proof of Theorem 5.1

Proof. We consider the imitation gap of the expert and the averaged learned policies, $\overline{\pi}$,

$$\begin{aligned}
V^{\pi_E} - V^{\overline{\pi}} &= \frac{1}{n}\sum_{i=0}^{n}\left(\mathop{\mathbb{E}}_{\zeta\sim\pi_E}\left[\sum_{h=1}^{H} r^*(s_h,a_h)\right] - \mathop{\mathbb{E}}_{\zeta\sim\pi_i}\left[\sum_{h=1}^{H} r^*(s_h,a_h)\right]\right) \\
&= \frac{1}{n}H\sum_{i=0}^{n}\left(\mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_E}}[r^*(s,a)] - \mathop{\mathbb{E}}_{(s,a)\sim d_\mu^{\pi_i}}[r^*(s,a)]\right) \\
&= \frac{1}{n}H\sum_{i=0}^{n} L(\pi_i, r^*) \\
&\leq \frac{1}{n}H\max_{r\in\mathcal{R}}\sum_{i=0}^{n} L(\pi_i, r)
\end{aligned}$$

where n is the number of updates to the reward function. The second line holds by definition of d_μ^π. The third line holds by definition of L. Applying the Statistical Difference of Losses (Lemma B.2), we have

$$\begin{aligned}
V^{\pi_E} - V^{\overline{\pi}} &\leq \frac{1}{n}H\max_{r\in\mathcal{R}}\sum_{i=0}^{n}\left(\hat{\mathrm{L}}(\pi_i, r) + \sqrt{\frac{C}{N}}\right) \\
&= \frac{1}{n}H\max_{r\in\mathcal{R}}\sum_{i=0}^{n}\left(\hat{\mathrm{L}}(\pi_i, r) - \hat{\mathrm{L}}(\pi_i, r_i) + \hat{\mathrm{L}}(\pi_i, r_i) + \sqrt{\frac{C}{N}}\right)
\end{aligned}$$

where $C = \ln\frac{2|\mathcal{R}|}{\delta}$ and M is the number of state-action pairs from the expert. Applying the Reward Regret Bound (Lemma B.1), we have

$$V^{\pi_E} - V^{\overline{\pi}} \leq \frac{1}{n}H\sum_{i=0}^{n}\left(\hat{\mathrm{L}}(\pi_i, r_i) + \sqrt{\frac{C}{N}}\right) + H\sqrt{\frac{C_1}{n}}$$

where $C_1 = 2\ln|\mathcal{R}|$. Applying the Loss Bound (Lemma B.4), we have

$$V^{\pi_E} - V^{\overline{\pi}} \leq \frac{1}{n}H\sum_{i=0}^{n}\left(\min\left\{\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N}}, C_B\left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right)\right\} + \sqrt{\frac{C}{N}},\right) + H\sqrt{\frac{C_1}{n}},$$

which simplifies to

$$V^{\pi_E} - V^{\overline{\pi}} \leq H\min\left\{\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N}}, C_B\left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right)\right\} + H\sqrt{\frac{C}{N}}, + H\sqrt{\frac{C_1}{n}},$$

where $C_B = \left\|\frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}}\right\|_\infty$, H is the horizon, N is the number of expert state-action pairs, M is the number of offline state-action pairs, n is the number of reward updates, $C_0 = 2\ln\frac{|\Pi||\mathcal{R}|}{\delta}$, $C = \ln\frac{2|\mathcal{R}|}{\delta}$, and $C_1 = 2\ln|\mathcal{R}|$. □

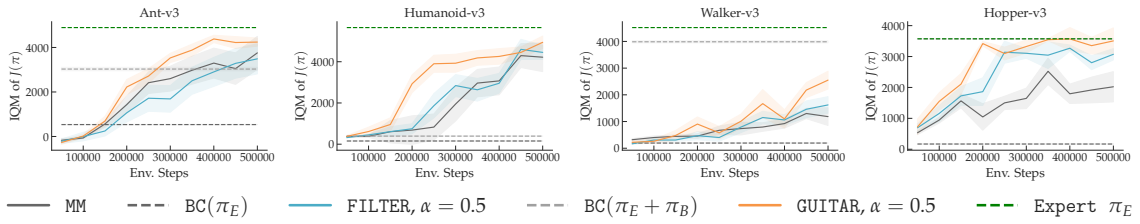

Figure 1: GUITAR, an IRL algorithm that uses resets to expert and sub-optimal data, outperforms other IRL algorithms—FILTER (which resets to expert states) and MM (which resets to the starting state)—on 3 out of the 4 environments considered. Standard errors are computed across 5 seeds. For all MuJoCo tasks, we use less than 1 full trajectory (100 expert state-action pairs for Ant and Humanoid, 300 state-action pairs for Walker, and 600 state-action pairs for Hopper). For antmaze-large, we use 1 successful trajectory (1000 expert state-action pairs).

C Experiments

In this section, we aim to answer the following questions:

1. In settings without access to arbitrary learner resets, can the sample efficiency of IRL be improved via roll-ins with a BC policy? We consider one additional, practical constraint to the two outlined in Section 5: in real-world robotics training, the learner cannot be reset to arbitrary states. In other words, the learner cannot be reset to a particular reset distribution (e.g. the expert states). Instead, we roll-in with a BC policy trained on the intended reset distribution.

2. Does incorporating sub-optimal data improve the sample efficiency of efficient IRL in the limited expert data setting? We consider the setting of limited expert data, which we supplement with sub-optimal data. We compare the results of BC and IRL algorithms that exclusively use the expert data to GUITAR, which incorporates both expert and sub-optimal data.

Because we consider the low expert data regime, we use the minimum amount of expert data that allows the baseline IRL algorithm to learn in each environment (less than one complete trajectory). We implement GUITAR with Soft Actor Critic (Haarnoja et al., 2018) for the policy and critic updates and a discriminator network for reward labels. We generate sub-optimal data by rolling out the expert policy with a probability $p^{\pi_b}_{\text{tremble}}$ of sampling a random action. We consider both high-quality offline data in the Walker and Hopper environments, each with $p^{\pi_B}_{\text{tremble}} = 0.05$, and low-quality offline data in the Ant and Humanoid environments, where $p^{\pi_B}_{\text{tremble}} = 0.25$. Additional implementation details can be found in Appendix D.

We compare GUITAR against two behavioral cloning baselines (Pomerleau, 1988) and two IRL baselines (Swamy et al., 2023). The first behavioral cloning baseline is trained exclusively on the expert data, BC(π_E), and the second is trained on the combination of expert and sub-optimal data, BC($\pi_E + \pi_b$). We compare against two IRL algorithms: a traditional IRL algorithm, MM, and an efficient IRL algorithm, FILTER (Swamy et al., 2023). The differences between MM, FILTER, and GUITAR can be summarized by what reset distribution they use. MM resets the learner to the true starting state (i.e. $\rho = \mu$); FILTER resets the learner to expert states (i.e. $\rho = \rho_E$), and GUITAR resets the learner to expert and suboptimal states (i.e. $\rho = \rho_{\text{mix}}$).

We see that the benefit of rolling in with a BC policy is dependent on the performance of the BC policy. In environments where the BC policy performs poorly, FILTER does not outperform MM (Ant, Humanoid, and Walker). However, by incorporating additional sub-optimal data, GUITAR is able to outperform poor-performing BC policies (Ant and Humanoid) and consistently outperform the other IRL algorithms.

D Implementation Details

We describe the implementation details in this section. We compare `GUITAR` against two behavioral cloning baselines (Pomerleau, 1988) and two IRL baselines (Swamy et al., 2023). The first behavioral cloning baseline is trained exclusively on the expert data, $\texttt{BC}(\pi_E)$, and the second is trained on the combination of expert and sub-optimal data, $\texttt{BC}(\pi_E + \pi_b)$. We compare against two IRL algorithms: (1) Swamy et al. (2021a)'s moment-matching algorithm, `MM`, a traditional IRL algorithm with the Jensen-Shannon divergence replaced by an integral probability metric, and (2) Swamy et al. (2023)'s efficient IRL algorithm, `FILTER`, that exclusively leverages expert data for resets. The differences between `MM`, `FILTER`, and `GUITAR` can be summarized by what reset distribution they use. `MM` resets the learner to the true starting state (i.e. $\rho = \mu$); `FILTER` resets the learner to expert states (i.e. $\rho = \rho_E$), and `GUITAR` resets the learner to expert and suboptimal states (i.e. $\rho = \rho_{\text{mix}}$).

We adapt Ren et al. (2024)'s codebase for our implementation and follow their implementation details. The details are restated here, with modifications where necessary. We apply Optimistic Adam (Daskalakis et al., 2017) for all policy and discriminator optimization. We also apply gradient penalties (Gulrajani et al., 2017) on all algorithms to stabilize the discriminator training. The policies, value functions, and discriminators are all 2-layer ReLu networks with a hidden size of 256. We sample 4 trajectories to use in the discriminator update at the end of each outer-loop iteration, and a batch size of 4096. In all IRL variants (`MM`, `FILTER`, and `GUITAR`), we re-label the data with the current reward function during policy improvement, rather than keeping the labels that were set when the data was added to the replay buffer. Ren et al. (2024) empirically observed such re-labeling to improve performance.

The code is available at **https://nico-espinosadice.github.io/efficient-IRL**.

D.1 MuJoCo Tasks

We detail below the specific implementations used in all MuJoCo experiments (Ant, Hopper, Humanoid, and Walker).

Parameter	Value
buffer size	1e6
batch size	256
γ	0.98
τ	0.02
Training Freq.	64
Gradient Steps	64
Learning Rate	Lin. Sched. 7.3e-4
policy architecture	256 x 2
state-dependent exploration	true
training timesteps	1e6

Table 1: Hyperparameters for baselines using SAC.

Expert Data. To experiment under the conditions of limited expert data, we set the amount of expert data to be the lowest amount that still enabled the baseline IRL algorithms to learn. For Ant and Humanoid, this was 100 expert state-action pairs. For Walker, this was 300 expert state-action pairs. For Hopper, this was 600 expert state-action pairs.

Sub-optimal Data. We generate the sub-optimal data by rolling out the expert policy with a probability $p^{\pi_B}_{\text{tremble}}$ of sampling a random action. $p^{\pi_B}_{\text{tremble}} = 0.25$ for the Ant and Humanoid environments, and $p^{\pi_B}_{\text{tremble}} = 0.05$ for the Walker and Hopper environments.

Discriminator. For our discriminator, we start with a learning rate of 8e−4 and decay it linearly over outer-loop iterations. We update the discriminator every 10,000 actor steps.

Baselines. We train all behavioral cloning baselines for 300k steps for Ant, Hopper, and Humanoid, and 500,000 steps for Walker2d. For `MM` and `FILTER` baselines, we follow the exact hyperparameters in Ren et al. (2024), with a notable modification to how resets are performed, discussed below. We use the Soft Actor Critic (Haarnoja et al., 2018) implementation provided by Raffin et al. (2021) with the hyperparameters in Table 1.

Reset Substitute. We mimic resets by training a BC policy on the reset distribution specified by each algorithm. `MM` does not employ resets. `FILTER`'s reset distribution is the expert data. `GUITAR`'s reset distribution is a mixture of the expert and sub-optimal data. The BC roll-in logic follows Ren et al. (2024)'s reset logic. The probability of performing a non-starting-state reset (i.e. an expert reset in `FILTER`) is α. If a non-starting-state reset is performed, we sample a random timestep t between 0 and the horizon, and we roll-out the BC policy in the environment for t steps.

`GUITAR`. `GUITAR` follows the same implementation and reset logic as `FILTER`, with the only change being the training data for the BC roll-in policy.

E Useful Lemmas

Theorem E.1 (Hoeffding's Inequality)**.** *If $Z_1, \ldots, Z_M$ are independent with $P(a \leq Z_i \leq b) = 1$ and common mean μ, then, with probability at least $1 - \delta$,*

$$|\overline{Z}_M - \mu| \leq \sqrt{\frac{c}{2M} \ln \frac{2}{\delta}}$$

where $c = \frac{1}{M} \sum_{i=1}^{M} (b_i - a_i)^2$.

Lemma E.2 (Online Mirror Descent Regret)**.** *Regret is defined as*

$$Reg_N = \frac{1}{N} \sum_{t=1}^{N} \ell(\hat{\mathbf{y}}_t, z_t) - \inf_{\mathbf{f} \in \mathcal{F}} \frac{1}{N} \sum_{t=1}^{N} \ell(\mathbf{f}, z_t).$$

Given $\mathcal{F} = \Delta(\mathcal{F}')$ and $\langle \mathbf{f}, \nabla_t \rangle = \mathbb{E}_{f' \sim \mathbf{f}}[\ell(f', (x_t, y_t))]$, where $\sup_{\nabla \in \mathcal{D}} \|\nabla\|_\infty \leq B$, let R be any 1-strongly convex function. If we use the Mirror descent algorithm with $\eta = \sqrt{\frac{2 \sup_{\mathbf{f} \in \mathcal{F}} R(\mathbf{f})}{NB^2}}$, then,

$$Reg_n \leq \sqrt{\frac{2B^2 \sup_{\mathbf{f} \in \mathcal{F}} R(\mathbf{f})}{N}}.$$

If R is the negative entropy function, then $\sup_{\mathbf{f} \in \mathcal{F}} R(\mathbf{f}) \leq \log |\mathcal{F}'|$.