EFFICIENT IMITATION UNDER MISSPECIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive imitation learning (IL) is a powerful paradigm for learning to make sequences of decisions from an expert demonstrating how to perform a task. Prior work in efficient imitation learning has focused on the *realizable* setting, where the expert's policy lies within the learner's policy class (i.e. the learner can perfectly imitate the expert in all states). However, in practice, perfect imitation of the expert is often impossible due to differences in state information and action space expressiveness (e.g. morphological differences between robots and humans.) In this paper, we consider the more general *misspecified setting*, where no assumptions are made about the expert policy's realizability. We introduce a novel structural condition, *reward-agnostic policy completeness*, and prove that it is sufficient for interactive IL algorithms to efficiently avoid the quadratically compounding errors that stymie offline approaches like behavioral cloning. We address an additional practical constraint—the case of limited expert data—and propose a principled method for using additional sub-optimal data to further improve the sample-efficiency of interactive IL algorithms. Finally, we corroborate our theory with experiments on a suite of continuous control tasks.

1 INTRODUCTION

Interactive imitation learning (IL) is a powerful paradigm for learning to make sequences of decisions from an expert demonstrating how to perform a task. While offline imitation learning approaches suffer from covariate shift between the training distribution (i.e. the expert's state distribution) and the test distribution (i.e. the learner's state distribution), interactive approaches enable the learner to roll-out its policy during train-time, effectively allowing it to train on states from the test distribution (Ross et al., 2011).

Broadly speaking, the difference between the expert's performance and the learned policy's performance can be attributed to three forms of error:

1. *Optimization error*: The error resulting from imperfect search within a policy class (e.g. due to non-convexity)
2. *Finite sample error*: The statistical error arising from limited expert demonstrations
3. *Misspecification error*: The irreducible error resulting from the learner's policy class not containing the expert's policy

Notably, misspecification error is a function of the expert policy and the learner's policy class; it cannot be improved by a better algorithm or more computation. However, prior work in imitation learning has focused on the first two sources of error and has avoided the misspecification error by imposing *expert realizability*: the assumption that the expert policy is within the learner's policy class (Kidambi et al., 2021; Swamy et al., 2021a; 2022c; Xu et al., 2023; Swamy et al., 2023; Ren et al., 2024). In other words, expert realizability implies that the learner can perfectly imitate the expert's actions in *all* states.

Unfortunately, in practice, it is often impossible to imitate the expert perfectly due to differences in state information. In self-driving applications, autonomous vehicles often have distinct perception compared to human drivers, leading to the human expert having privileged information over the learner (Swamy et al., 2022b). Similarly, in legged locomotion, recent work has frequently used an expert policy trained with privileged information about the environment configuration, which is

unknown in the real world and therefore unavailable to the learner (Kumar et al., 2021; 2022; Liang et al., 2024).

Realizability can also be an inaccurate assumption due to a learner's limited action space expressiveness. In humanoid robotics, the morphological differences between robots and humans prevent perfect human-to-robot motion retargeting (Zhang et al., 2024; He et al., 2024; Al-Hafez et al., 2023). More generally, physical robots face the problem of changing dynamics, due to wear and tear and manufacturing imperfections that result in varying link lengths and other physical properties between robots of the same make and model. It is, therefore, unreasonable to assume that realizability holds for robots of the same model.

In this paper, we consider the more general *misspecified* setting, where the learner is not *necessarily* capable of perfectly imitating the expert's behavior. We analyze how the misspecification error interrelates with the optimization and finite sample errors. More specifically, our paper addresses the question:

Under what conditions can interactive imitation learning in the misspecified setting avoid quadratically compounding errors, while retaining sample efficiency?

The last two words of the preceding question are central to our study. By reducing the problem of imitation learning to reinforcement learning with a learned reward, interactive imitation learning approaches like inverse reinforcement learning (IRL; Ziebart et al. (2008); Ho & Ermon (2016)) face a similar exploration problem to that of reinforcement learning— in the worst case, needing to explore all paths through the state space to find one reward (e.g. a tree structured problem with sparse rewards) (Swamy et al., 2023). In order to focus the exploration on useful states, efficient imitation algorithms leverage the expert's state distribution. 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 reset-based techniques as *efficient IRL*. Intuitively, efficient IRL can be understood as replacing the hard problem of global reinforcement learning (RL) exploration to the local exploration problem of "staying on the expert's path."

However, prior work assumes that "staying on the expert's path" is possible by expert realizability (Swamy et al., 2023). In the misspecified setting, following the expert's path may not be possible for the learner, and it is thus unclear whether expert resets can introduce compounding errors. We address this open question with a novel structural condition that, crucially, does not imply expert realizability.

Contribution 1. We define a new structural condition for the misspecification setting, *reward-agnostic policy completeness*, under which efficient imitation learning algorithms can avoid quadratically compounding errors.

Prior work in efficient IRL only considers resets to the expert's state distribution, but in the misspecified setting, it is unknown whether resetting to states from an *unrealizable* expert policy is *always* beneficial.

Contribution 2. We consider the question of what the optimal reset distribution is in the misspecified setting. We present two settings in which we show that resetting to offline data from a *realizable* behavior policy outperforms expert resets.

Finally, in addition to expert realizability, prior work imposes two additional assumptions that are often untrue in practice. First, prior work assumes access to infinite expert data (Swamy et al., 2021a; 2023; Ren et al., 2024), which is unrealistic when collecting expert data is resource-intensive. For instance, the process of collecting expert data through robot teleoperation is laborious and requires human teleoperators (Fu et al., 2024). In the finite expert sample regime, there may only be partial coverage of the expert's true state distribution. However, for many problems of practical interest, there is often access to a larger source of offline data (e.g. poor teleoperation or imperfect driving). We propose using this data to complement a limited set of expert demonstrations, without compromising solution quality guarantees.

Contribution 3. We propose a principled method for incorporating offline data to improve the sample efficiency of IRL and prove the conditions under which it is beneficial.

We begin by defining the problem, and we postpone an in-depth discussion of related work to Appendix A.

2 Imitation Learning in the Misspecified Setting

2.1 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, which is defined as $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$. 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. As previous stated, much of the theoretical analysis in IL relies on the impractical assumption of a realizable expert policy (i.e. one that lies within the learner's policy class Π) (Kidambi et al., 2021; Swamy et al., 2021a;b; 2022a; Xu et al., 2023; Ren et al., 2024). In contrast to prior work, *we focus on the more realistic misspecified setting*, where the expert policy π_E is not necessarily in the policy class Π. We consider a known sample of the expert policy's trajectories, where the dataset of state-action pairs sampled from the expert is $D_E = D_1 \cup D_2 \cup \ldots \cup D_H$, where $D_h = \{s_h, a_h\} \sim d^{\pi_E}_{\mu,h}$ 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, we find an equilibrium strategy for the game

$$\min_{\pi\in\Pi}\max_{r\in\mathcal{R}} J(\pi_E, r) - J(\pi, r), \tag{1}$$

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 (e.g. Line 5, Follow The Regularized Leader McMahan (2011)), and the policy is updated via a best response (e.g. Line 7,

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

Input: Expert state-action distributions ρ_E, policy class Π, reward class $\mathcal{R}$
Output: Trained policy π
for $i = 1$ to N **do**
    `// No-regret step over rewards (e.g. FTRL)`
    $r_i \leftarrow \arg\max_{r \in \mathcal{R}} J(\pi_E, r) - J(\text{Unif}(\pi_{1:i}), r)$
    `// Expert-competitive response by RL algorithm (e.g. PSDP, Alg. 3)`
    $\pi_i \leftarrow \texttt{RL}(r = r_i, \rho = \rho_E)$
end for
Return π_N

PSDP, Algorithm 3) using an RL subroutine with reward labels r (Ratliff et al., 2006; 2009; Ziebart et al., 2008; Swamy et al., 2021a).

Reset Distribution. RL algorithms require a reset distribution be specified. Often, this is simply the MDP's starting state distribution, μ (Mnih, 2013; Schulman et al., 2015; 2017; Haarnoja et al., 2018). Because IRL algorithms use an RL subroutine, we differentiate between traditional IRL and efficient IRL by their RL subroutine'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, the reset distribution is the expert's state distribution (i.e. $\rho = \rho_E$), which changes the RL subroutine from a best response step to an expert-competitive response (Swamy et al., 2023; Ren et al., 2024). Swamy et al. (2023) proved that using the expert's state distribution in the realizable setting sped up learning without compromising solution quality.

2.2 Efficient Imitation Under Misspecification is Hard

Our paper considers efficient IRL in the misspecified setting, to which the obvious question arises:

Is efficient IRL possible in the misspecified setting?

Efficiency is measured with respect to environment interactions, and we consider polynomial interaction complexity in the MDP's horizon as efficient. We start by considering the most general setting of IRL, where no assumptions are made about the MDP's structure, the policy class, or the expert's policy (i.e., we do not assume $\pi_E \in \Pi^H$). We present a lower bound on efficient IRL that shows, in general, efficient IRL under misspecification is not possible.

Theorem 2.1 (Lower Bound on Misspecified 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 2.1, we establish that polynomial sample complexity in the misspecified 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, even with access to a queryable expert policy like DAgger (Ross et al., 2011). Notably, this lower bound focuses on *sample* efficiency. *Statistically* efficient—with respect to the amount of expert data—imitation *is* possible, and we present a statistically optimal imitation learning algorithm for the misspecified setting in Appendix B.

3 Policy Complete Inverse Reinforcement Learning

The result from Section 2.2 establishes that sample efficient IRL in the misspecified setting is not possible without assuming additional structure on the MDP, begging the question of what assump-

[1] A generative model allows the learner to query the transition and reward associated with a state-action pair on any state, in contrast to an online interaction model that can only play actions on sequential states in a trajectory. For a more thorough discussion of their differences, see Kakade (2003).

tions suffice. We provide an answer via an extension of *policy completeness*—a condition used in the analysis of policy gradient RL algorithms—and a corresponding efficient algorithm.

3.1 Approximate Policy Completeness

Intuitively, the policy completeness condition requires that the learner have a way of improving the current policy's performance—*without* the requirement of matching the actions of 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 inverse reinforcement learning setting is unknown and is instead learned throughout training. In response, we introduce *reward-agnostic policy completeness*, the natural generalization of policy completeness extended to the imitation learning setting.

Definition 3.1 (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 r_i, define the reward-indexed policy completeness error of $\mathcal{M}$ to be*

$$\epsilon_{\Pi}^{\pi_i, r_i} := \mathbb{E}_{s\sim\rho_E}\left[\max_{a\in\mathcal{A}} A_{r_i}^{\pi_i}(s,a)\right] - \max_{\pi'\in\Pi} \mathbb{E}_{s\sim\rho_E}\mathbb{E}_{a\sim\pi'(\cdot|s)}\left[A_{r_i}^{\pi_i}(s,a)\right]. \tag{2}$$

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 over all policies, including those not in the policy class.

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 3.2 (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*

$$\epsilon_{\Pi} := \max_{\pi\in\Pi, r\in\mathcal{R}} \epsilon_{\Pi}^{\pi, r} \tag{3}$$

$$= \max_{\pi\in\Pi, r\in\mathcal{R}} \left(\mathbb{E}_{s\sim\rho_E}\left[\max_{a\in\mathcal{A}} A_r^{\pi}(s,a)\right] - \max_{\pi'\in\Pi} \mathbb{E}_{s\sim\rho_E}\mathbb{E}_{a\sim\pi'(\cdot|s)}\left[A_r^{\pi}(s,a)\right]\right) \tag{4}$$

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 $\epsilon_{\Pi} = O(1)$.

3.2 Efficient IRL with Approximate Policy Completeness

We begin by presenting our efficient, reset-based IRL algorithm, **GU**iding **I**mi**T**aters with **A**rbitrary **R**oll-ins (`GUITAR`), before proving its policy performance bounds under approximate policy completeness. The full IRL procedure is outlined in Algorithm 2.

Existing efficient IRL algorithms, such as MMDP (Swamy et al., 2023), reset the learner exclusively to expert states (i.e. the case where $\rho = \rho_E$). `GUITAR` can be seen as extending MMDP to a general reset distribution in the misspecified setting. We will focus on expert resets in the misspecified setting first, and we then consider other reset distributions in Section 4. For a more thorough description of the algorithm and its updates, see Appendix D.

Policy Update. Following Ren et al. (2024)'s reduction of inverse RL to expert-competitive RL, we can use any RL algorithm to generate an expert-competitive response. We employ Policy Search by Dynamic Programming (PSDP, Bagnell et al. (2003)), shown in Algorithm 3, for its strong theoretical guarantees. In practice, any RL algorithm can be used, such as Soft Actor Critic (SAC, Haarnoja et al. (2018)).

Algorithm 2 GUiding ImiTaters with Arbitrary Roll-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
$$\hat{L}(\pi, r) = \mathbb{E}_{(s,a)\sim\rho_E} r(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^\pi} r(s,a) \quad \texttt{// Loss function} \tag{5}$$
6: Optimize
$$r_i = \texttt{OMD}(\pi_1, \ldots, \pi_{i-1}) \quad \texttt{// Reward's no-regret update} \tag{6}$$
7: Optimize
$$\pi_i = \texttt{PSDP}(r = r_i, \rho = \rho_{\text{mix}}) \quad \texttt{// RL's expert-competitive response} \tag{7}$$
8: **end for**
9: **Return** π_i with lowest validation error

Reward Update. We employ a no-regret update to the reward function. We employ Online Mirror Descent (OMD, Nemirovskij & Yudin (1983)) for its strong theoretical guarantees (Beck & Teboulle, 2003; Srebro et al., 2011), but in practice, any no-regret update can be used, such as gradient descent.

3.3 Analysis in the Infinite-Sample Regime

For clarity, we first present the sample complexity of Algorithm 2 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 Appendix **??**.

Theorem 3.3 (Sample Complexity of Algorithm 2)**.** *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 2. Then,*

$$V^{\pi_E} - V^{\overline{\pi}} \leq \underbrace{H\epsilon_\Pi}_{\textit{Misspecification Error}} + \underbrace{H^2\epsilon}_{\textit{Policy Optimization Error}} + \underbrace{H\sqrt{\frac{\ln|\mathcal{R}|}{n}}}_{\textit{Reward Regret}}, \tag{8}$$

where H is the horizon, n is the number of outer-loop iterations of the algorithm, and $\overline{\pi}$ is the trajectory-level average of the learned policies (i.e. π_i at each iteration $i \in [n]$ of Algorithm 2).

Misspecification Error. The error is comprised of three terms. The first 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, without requiring $\pi_E \in \Pi$.

Policy Optimization Error. The second 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.

Reward Regret. 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.

In short, with sufficient iterations of Algorithm 2, GUITAR can avoid quadratically compounding errors under approximate policy completeness, even in the misspecified setting.

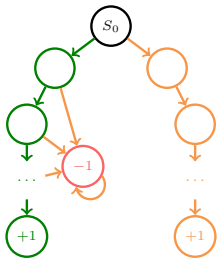

Figure 1: In the MDP, π_E selects the green (left) actions, and $\Pi = \{\pi\}$, where π selects the orange (right) actions. The reward class $\mathcal{R}$ contains one reward function shown in the figure.

4 Reset Distributions under Misspecification

In Section 3, we proved a condition under which efficient IRL with expert resets avoids quadratically compounding error in the misspecified setting, but is resetting to states from an *unrealizable* expert policy *always* beneficial? This question remains unaddressed by prior work, and we present two cases under which expert resets in the misspecified setting may not be optimal.

Claim 4.1 (The Perils of Misspecification)**.** *We claim that resetting the learner to expert states in the misspecified setting does not necessarily lead to optimal IRL sample efficiency. We present two settings under which the expert's state distribution (i.e. expert resets) is not the optimal reset distribution, specifically when*

1. *The expert walks along a cliff, and*
2. *There is finite expert data.*

4.1 Misspecified Setting 1: Expert Cliff Walks

Consider the MDP in Figure 1, an example of a misspecified setting where the learner's policy class is deficient on expert states. The efficient IRL problem is reduced to performing RL with resets to the expert's states. Intuitively, the RL optimization problem over π_E's state distribution (the green states) is harder than over π's state distribution (the orange states): the learner can never reach the optimal realizable policy from expert states (other than the starting state).

Unrealizable Cliff Walking: In the case where the expert walks along an unrealizable cliff, we assume that the policies in the learner's policy class cannot replicate the cliff walking behavior. We can consider the best *realizable* policy:

$$\pi^\star := \max_{\pi \in \Pi} J(\pi, r^*). \tag{9}$$

In other words, $\pi^\star$ the best policy *in* the learner's policy class under the ground-truth reward function. Since $\pi^\star \in \Pi$ and the expert walks along an unrealizable cliff, we assume that $\pi^\star$'s overlap with the expert's state distribution is small: $\text{Supp}(d_\mu^{\pi_E}) \cap \text{Supp}(d_\mu^{\pi^\star}) \approx 0$. We can measure the reset distribution's coverage of $\pi^\star$ via the standard concentrability coefficient:

$$C_S := \left\| \frac{d_\mu^{\pi^\star}}{\rho} \right\|_\infty < \infty \tag{10}$$

where ρ is the reset distribution for IRL. We theoretically analyze this setting in Appendix H.

4.2 Misspecified Setting 2: Finite Expert Data

Additionally, we consider the case of finite expert data. In this case, estimating the expert's true state distribution $d_\mu^{\pi_E}$ with finite samples from the expert policy π_E, may have a non-trivial finite sample error. To remedy this, we consider having a reset distribution that covers the expert data. In such a setting, it may be advantageous to consider access to some 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

$$C_B := \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty < \infty \tag{11}$$

This can be thought of intuitively as requiring that if the expert visits a state, the offline dataset does too, but not necessarily with an equal visitation frequency. In other words, the offline data covers the expert data. We theoretically analyze this setting in Section 5.

5 Incorporating Offline Data in Misspecified IRL

In the second misspecified setting from Section 4, we considered the case of finite expert data. How to use offline data in IRL remains an open question. Prior work has used sub-optimal data in IRL to learn reward functions (Brown et al., 2019; Brown & Niekum, 2019; Poiani et al., 2024), but such an approach requires strong assumptions about the data's structure. Without sufficient structure, incorporating the sub-optimal data into the discriminator update (i.e. the reward function) would result in the offline behavior being valued as optimal—an undesirable training outcome.

In our paper, we propose incorporating offline data into the reset distribution, thereby using it for policy optimization but not reward learning. Unlike prior work, our approach requires minimal assumptions on the offline data and maintains the strong performance guarantees from Section 3. We formalize our approach below.

Resetting to Offline Data. Our approach of using offline data for resets requires no change to the structure of Algorithm 2. We simply set PSDP's reset distribution to the mixture of offline and expert states, $\rho = \rho_{\text{mix}}$, where we define $D_{\text{mix}} = D_E \cup D_{\text{off}}$ and ρ_{mix} as the uniform distribution over D_{mix}. Let

$$\nu := \frac{N}{N+M} d_\mu^{\pi_E} + \frac{M}{N+M} d_\mu^{\pi_B}. \tag{12}$$

The reward update remains the same. 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}, which we explicitly denote by $\epsilon_\Pi^{\rho_{\text{mix}}}$ unless it is clear from context.

5.1 When is Offline Data Beneficial in IRL?

Next, we answer the natural question of when offline resets are beneficial to IRL.

Corollary 5.1 (Benefit of Offline Data)**.** *Incorporating offline data into the reset distribution improves the sample efficiency of IRL when*

$$\left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\pi_B}} \right\|_\infty \left(\epsilon_\Pi^{\rho_{mix}} + \epsilon_\Pi^{\rho_{mix}} \sqrt{\frac{C_{\Pi,\mathcal{R}}}{N+M}} \right) < \epsilon_\Pi^{\rho_{mix}} + \epsilon_\Pi^{\rho_{mix}} \sqrt{\frac{C_{\Pi,\mathcal{R}}}{N}} \tag{13}$$

where N is the number of expert state-action pairs, M is the number of offline state-action pairs, and $C_{\Pi,\mathcal{R}} = \ln \frac{|\Pi||\mathcal{R}|}{\delta}$.

Corollary 5.1 presents a sufficient condition for offline data improving GUITAR's sample efficiency *over* the algorithm with resets to strictly expert data. We observe that the benefit of offline data partly depends on the how well the offline data covers the expert data and the amount of expert and offline data. Intuitively, we can think of the coverage coefficient C_B as the "exchange rate," measuring how useful the offline data is in comparison to the expert data. When the offline data covers the expert data well, C_B is small, so the offline data may be beneficial. Considering the special case where the "offline" data is collected from the expert policy π_E, then $C_B = 1$. The bound becomes equivalent to the case of having $N + M$ number of expert data samples.

However, one should also observe the dependence on the policy completeness term, $\epsilon_\Pi^{\rho_{\text{mix}}}$. The policy completeness term measures how flexible or rich the policy class is on a given state distribution, in this case the mixture of the data ρ_{mix}. Thus, the condition can be interpreted as representing a trade-off between the offline data's coverage of the expert data and the richness of the policy class on the data's state distribution. We present the full finite sample analysis and sample complexity in Appendix F.

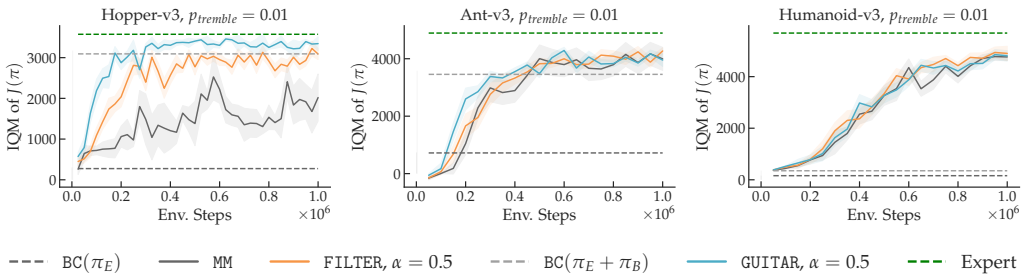

Figure 2: GUITAR, an IRL algorithm that uses resets to expert and offline data, shows improvement over IRL algorithms—FILTER (which resets to expert states) and MM (which resets to the starting state)—when the reset (roll-in) distribution has greater coverage of the expert's data. Standard errors are computed across 10 seeds. For all MuJoCo tasks, we use less than 1 full trajectory (600 expert state-action pairs for Hopper, 50 for Ant, and 100 for Humanoid). During evaluation, agents sample a random action with probability $p_{tremble}$.

6 Experiments

In Section 4, we proposed two misspecified settings where resetting directly to expert states may not be the optimal reset distribution. In this section, we aim to empirically corroborate our theoretical observations with experiments in continuous control tasks, specifically:

1. In settings with significant misspecification, where the expert's behavior is not imitable, can resetting to non-expert data better reduce the exploration problem of the RL subroutine than expert resets? We consider a setting with significant misspecification, where the learner must solve a maze through different path than the expert. We analyze the effects of varying the reset distribution, comparing resets to expert states, a realizable behavioral policy's states, and the true starting state.

2. In misspecified settings with finite expert data, does improving the reset distribution's coverage of the expert's state distribution improve IRL's sample efficiency? We consider the practical setting of having a small amount of expert data, such that it does not perfectly cover the expert's state distribution due to finite sample errors. We analyze the effects of improving the reset distribution's coverage of the expert's state distribution by incorporating offline data into the reset distribution.

We implement GUITAR with Soft Actor Critic (Haarnoja et al., 2018) for the policy and critic updates in MuJoCo tasks and TD3 (Fujimoto et al., 2018) for D4RL tasks, with a discriminator network for reward labels in both. See Appendix G for a more detailed discussion of implementation details.

6.1 Empirical Analysis of Misspecified Setting 1: Expert Cliff Walking

We first consider a variant of the expert cliff walking setting from Section 4.1. In this experiment, an ant learns to solve a quadruped ant learns to solve a maze to reach a goal position (Fu et al., 2020). The expert can solve the maze by "walking along the cliff," a behavior that the learner is unable to replicate. More specifically, the expert takes a path through the maze that is "blocked" to the learner. Instead, the learner must find a different route through the maze to reach the goal.

In this setting, we have access to data from an unrealizable expert policy, π_E, and offline data from a realizable behavioral policy, π_B. We compare our algorithm, GUITAR, against two behavioral cloning baselines (Pomerleau, 1988) and two IRL baselines (Swamy et al., 2023; 2021a). 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 offline 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., 2021a; 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 offline states (i.e. $\rho = \rho_B$).

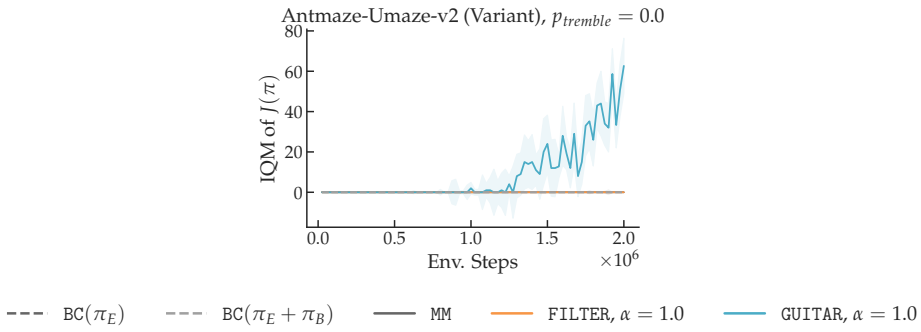

Figure 3: We train all IRL algorithms—MM, FILTER, GUITAR—with the same expert data. The only difference between the algorithms is their reset distributions. MM resets to the starting state, FILTER to expert data, and GUITAR to the offline data. GUITAR is the only imitation algorithm that solved the hard exploration problem of learning a different route through the maze than the expert policy's path. Standard errors are computed across 5 seeds.

We observe that GUITAR is the only algorithm capable of solving the hard exploration problem in the strongly misspecified setting, confirming that the offline data—states from a behavior policy—is a better reset distribution than expert states in the task.

6.2 Empirical Analysis of Misspecified Setting 2: Finite Expert Data

Next, we consider a variant of the finite expert data setting from Section 4.2. In this experiment, we consider the practical case of having a limited amount of expert data and a large amount of sub-optimal offline data. To ensure we train in the low-data regime, we used the minimum amount of expert data that allowed the baseline IRL algorithm (MM) to learn in each environment (notably, less than one full episode). The offline data was generated by rolling out the pretrained expert policy with a probability $p^{\pi_b}_{\text{tremble}}$ of sampling a random action.

Based on our theoretical analysis in Section 5, the sample efficiency of IRL should improve as the reset distribution's coverage of the expert's state distribution improves. More formally, this is when,

$$C_B = \left\| \frac{d_\mu^{\pi_E}}{\rho} \right\|_\infty \to 1, \tag{14}$$

where $d_\mu^{\pi_E}$ is the expert's state distribution and ρ is the reset distribution.

Notably, we consider the practical constraint of not having access to arbitrary learner resets, a setting common in real robotics applications, where the robot cannot be reset to an arbitrary state. Instead, we mimic resets by rolling in with a BC policy trained on the corresponding reset distribution. More specifically, FILTER's reset distribution are expert states, so FILTER rolls in with BC(π_E). GUITAR's reset distribution are a mixture of expert and offline states, so GUITAR rolls in with BC($\pi_E + \pi_b$). MM continues to reset to the environment's true starting state.

Since GUITAR reset to BC($\pi_E + \pi_b$), the performance of BC($\pi_E + \pi_b$) is an estimation of the GUITAR's reset distribution's coverage of the expert's state distribution, and correspondingly for FILTER's reset distribution and BC(π_E). From Figure 2, we see that as the reset distribution's coverage of the expert's states improves—as measured by the corresponding BC performance—so does the performance of the IRL algorithm.

References

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.

Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. *arXiv preprint arXiv:2311.02496*, 2023.

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.

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.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pp. 783–792. PMLR, 2019.

Daniel S Brown and Scott Niekum. Deep bayesian reward learning from preferences. *arXiv preprint arXiv:1912.04472*, 2019.

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

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

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.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

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.

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

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.

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.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

Ashish Kumar, Zhongyu Li, Jun Zeng, Deepak Pathak, Koushil Sreenath, and Jitendra Malik. Adapting rapid motor adaptation for bipedal robots. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1161–1168. IEEE, 2022.

Yichao Liang, Kevin Ellis, and João Henriques. Rapid motor adaptation for robotic manipulator arms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16404–16413, 2024.

Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 525–533. JMLR Workshop and Conference Proceedings, 2011.

V Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

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.

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, 2009.

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.

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

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.

Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

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.

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, 2008.

A 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; Song et al., 2022). Similar to Song et al. (2022), we consider access to offline data but differ by considering the imitation learning setting, while Song et al. (2022) considers the known-reward reinforcement learning setting. 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. We use Jia et al. (2024)'s lower bound on agnostic RL with expert feedback to show why agnostic IRL is hard.

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).

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 (i.e. queryable) expert and *recoverability* (Rajaraman et al., 2021; Swamy et al., 2021a), the former of 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. 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 offline data to improve the interaction efficiency of our algorithm. Incorporating structured offline data has been proposed to learn reward functions in IRL (Brown et al., 2019; Brown & Niekum, 2019; Poiani et al., 2024), but rely on stronger assumptions about the structure of the offline data. In contrast, we do not use offline data in learning a reward function, instead using it to accelerate policy optimization.

Inverse Reinforcement Learning. We build upon Swamy et al. (2023)'s technique of speeding up IRL by leveraging the expert's state distribution for learner resets. Our paper introduces the following key improvements to Swamy et al. (2023)'s work. First, while Swamy et al. (2023) relies on the impractical assumption of expert realizability, we tackle the more general, misspecified setting. Second, instead of assuming access to infinite expert data like Swamy et al. (2023), we consider the finite sample regime and further demonstrate how to incorporate offline data into IRL. Lastly, we extend Swamy et al. (2023)'s analysis of optimization error to the two other forms of error, finite sample and misspecification, and provide conditions under which quadratically compounding errors can be avoided in the misspecified setting.

B STATISTICALLY OPTIMAL IMITATION UNDER MISSPECIFICATION

We begin with the following question: ignoring computation efficiency, what is the statistically optimal algorithm (with respect to the number of expert samples) for imitation learning in the misspecified setting?

We present ***Scheffé Tournament Imitation LEarning (STILE)***, a statistically optimal algorithm for the misspecified setting. For any two policies π and π', we denote by $f_{\pi,\pi'}$ the following witness function:

$$f_{\pi,\pi'} := \arg \max_{f:\|f\|_\infty \leq 1} \left[\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi'}} f(s,a) \right], \tag{15}$$

and the set of witness functions as

$$\mathcal{F} = \{ f_{\pi,\pi'} : \pi, \pi' \in \Pi, \pi \neq \pi' \}. \tag{16}$$

Note that $|\mathcal{F}| \leq |\Pi|^2$. STILE selects $\hat{\pi}$ using the following procedure:

$$\hat{\pi} \in \arg\min_{\pi\in\Pi} \left[\max_{f\in\mathcal{F}} \left(\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \frac{1}{N} \sum_{i=1}^{M} f(s_i^*, a_i^*) \right) \right], \tag{17}$$

where $(s_i^*, a_i^*) \in D_E$. Notably, running a tournament algorithm requires comparing every pair of policies, which is not feasible with policy classes like deep neural networks, making STILE impractical to implement.

We present the analysis in the infinite-horizon setting for convenience.

Theorem B.1 (Sample Complexity of STILE)**.** *Assume Π is finite and $\pi^\star \in \Pi$. With probability at least $1-\delta$, STILE finds a policy $\hat{\pi}$*

$$V^{\pi_E} - V^{\hat{\pi}} \leq \frac{4}{1-\gamma} \sqrt{\frac{2\ln(|\Pi|) + \ln\left(\frac{1}{\delta}\right)}{M}}. \tag{18}$$

Proof. The proof relies on a uniform convergence argument over $\mathcal{F}$ of which the size is $|\Pi|^2$. First, note that for all policies $\pi \in \Pi$:

$$\max_{f\in\mathcal{F}} \left(\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi_E}} f(s,a) \right) = \max_{f:\|f\|_\infty\leq 1} \left(\mathbb{E}_{s,a\sim d^\pi} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi_E}} f(s,a) \right) \tag{19}$$

$$= \|d^\pi - d^{\pi_E}\|_1 \tag{20}$$

where the first equality comes from the fact that $\mathcal{F}$ includes $\arg\max_{f:\|f\|_\infty\leq 1} \left[\mathbb{E}_{s,a,s'\sim d^\pi} f(s,a) - \mathbb{E}_{s,a,s'\sim d^{\pi_E}} f(s,a) \right]$

Via Hoeffding's inequality and a union bound over $\mathcal{F}$, we get that with probability at least $1-\delta$, for all $f \in \mathcal{F}$:

$$\left| \frac{1}{M} \sum_{i=1}^{M} f(s_i^*, a_i^*) - \mathbb{E}_{s,a\sim d^{\pi_E}} f(s,a) \right| \leq 2\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}} \tag{21}$$

$$:= \epsilon_{\text{stat}}. \tag{22}$$

Denote

$$\hat{f} := \arg\max_{f\in\mathcal{F}} \left[\mathbb{E}_{s,a\sim d^{\hat{\pi}}} f(s,a) - \mathbb{E}_{s,a\sim d^{\pi_E}} f(s,a) \right] \tag{23}$$

and

$$\tilde{f} := \arg\max_{f\in\mathcal{F}} \mathbb{E}_{s,a\sim d^{\hat{\pi}}} f(s,a) - \frac{1}{M} \sum_{i=1}^{M} f(s_i, a_i). \tag{24}$$

Hence, for $\hat{\pi}$, we have:

$$\left\|d^{\hat{\pi}} - d^{\pi_E}\right\|_1 = \mathbb{E}_{s,a\sim d^{\hat{\pi}}} \hat{f}(s,a) - \mathbb{E}_{s,a\sim d^{\pi_E}} \hat{f}(s,a) \tag{25}$$

$$\leq \mathbb{E}_{s,a\sim d^{\hat{\pi}}} \hat{f}(s,a) - \frac{1}{M}\sum_{i=1}^{M} \hat{f}(s_i^*, a_i^*) + \epsilon_{\text{stat}} \tag{26}$$

$$\leq \mathbb{E}_{s,a\sim d^{\pi_E}} \tilde{f}(s,a) - \frac{1}{M}\sum_{i=1}^{M} \tilde{f}(s_i^*, a_i^*) + \epsilon_{\text{stat}} \tag{27}$$

$$\leq \mathbb{E}_{s,a\sim d^{\pi_E}} \tilde{f}(s,a) - \mathbb{E}_{s,a\sim d^{\pi_E}} \tilde{f}(s,a) + 2\epsilon_{\text{stat}} \tag{28}$$

$$= 2\epsilon_{\text{stat}} \tag{29}$$

where we use the optimality of $\hat{\pi}$ in the third inequality.

Recall that $V^{\pi} = \mathbb{E}_{s,a\sim d^{\pi}} r(s,a)/(1-\gamma)$, so we have:

$$V^{\hat{\pi}} - V^{\pi_E} = \frac{1}{1-\gamma}\left(\mathbb{E}_{s,a\sim d^{\hat{\pi}}} r(s,a) - \mathbb{E}_{s,a\sim d^{\pi_E}} r(s,a)\right) \tag{30}$$

$$\leq \frac{\sup_{s,a}|r(s,a)|}{1-\gamma}\|d^{\hat{\pi}} - d^{\pi_E}\|_1 \tag{31}$$

$$\leq \frac{2}{1-\gamma}\epsilon_{\text{stat}} \tag{32}$$

This concludes the proof. □

B.1 Sample Complexity of STILE in the Misspecified Settting

Theorem B.2 (Sample Complexity of STILE in the Misspecified Setting)**.** *Assume Π is finite, but $\pi_E \notin \Pi$. With probability at least $1-\delta$, STILE learns a policy $\hat{\pi}$ such that:*

$$V^{\pi_E} - V^{\hat{\pi}} \leq \frac{1}{1-\gamma}\|d^{\pi_E} - d^{\hat{\pi}}\|_1 \tag{33}$$

$$\leq \frac{3}{1-\gamma}\min_{\pi\in\Pi}\|d^{\pi} - d^{\pi_E}\|_1 + \tilde{O}\left(\frac{1}{1-\gamma}\sqrt{\frac{\ln(|\Pi|)+\ln(1/\delta)}{M}}\right) \tag{34}$$

Proof. We first define some terms below. Denote $\tilde{\pi} := \arg\min_{\pi\in\Pi}\|d^{\pi} - d^{\pi_E}\|_1$. Let us denote:

$$\tilde{f} = \arg\max_{f\in\mathcal{F}}\left[\mathbb{E}_{s,a\sim d^{\hat{\pi}}} f(s,a) - \mathbb{E}_{s,a\sim d^{\tilde{\pi}}} f(s,a)\right], \tag{35}$$

$$\bar{f} = \arg\max_{f\in\mathcal{F}}\left[\mathbb{E}_{s,a\sim d^{\hat{\pi}}} f(s,a) - \frac{1}{M}\sum_{i=1}^{M} f(s_i^\star, a_i^\star)\right], \tag{36}$$

$$f' = \arg\max_{f\in\mathcal{F}}\left[\mathbb{E}_{s,a\sim d^{\tilde{\pi}}}\left[f(s,a)\right] - \frac{1}{M}\sum_{i=1}^{M} f(s_i^\star, a_i^\star)\right]. \tag{37}$$

Starting with triangle inequality, we have:

$$\left\|d^{\hat{\pi}} - d^{\pi^*}\right\|_1 \leq \left\|d^{\hat{\pi}} - d^{\tilde{\pi}}\right\|_1 + \left\|d^{\tilde{\pi}} - d^{\pi^*}\right\|_1 \tag{38}$$

$$= \mathbb{E}_{s,a\sim d^{\hat{\pi}}}\left[\tilde{f}(s,a)\right] - \mathbb{E}_{s,a\sim d^{\tilde{\pi}}}\left[\tilde{f}(s,a)\right] + \left\|d^{\tilde{\pi}} - d^{\pi^*}\right\|_1 \tag{39}$$

$$= \mathbb{E}_{s,a\sim d^{\hat{\pi}}}\left[\tilde{f}(s,a)\right] - \frac{1}{M}\sum_{i=1}^{M}\tilde{f}(s_i,a_i^*) + \frac{1}{M}\sum_{i=1}^{M}\tilde{f}(s_i,a_i^*) - \mathbb{E}_{s,a\sim d^{\tilde{\pi}}}\left[\tilde{f}(s,a)\right] + \left\|d^{\tilde{\pi}} - d^{\pi^*}\right\|_1 \tag{40}$$

$$\leq \mathbb{E}_{s,a\sim d^{\hat{\pi}}}\left[\tilde{f}(s,a)\right] - \frac{1}{M}\sum_{i=1}^{M}\bar{f}(s_i,a_i^*) + \frac{1}{M}\sum_{i=1}^{M}\tilde{f}(s_i,a_i^*) - \mathbb{E}_{s,a\sim d^{\tilde{\pi}}}\left[\tilde{f}(s,a)\right] + \left[\mathbb{E}_{s,a\sim d^{\pi_E}}\tilde{f}(s,a) - \mathbb{E}_{s,a\sim d^{\tilde{\pi}}}\tilde{f}(s,a)\right] + \left\|d^{\tilde{\pi}} - d^{\pi^*}\right\|_1 \tag{41}$$

$$\leq \mathbb{E}_{s,a\sim d^{\hat{\pi}}}\left[f'(s,a)\right] - \frac{1}{M}\sum_{i=1}^{M} f'(s_i,a_i^*) + 2\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}} + 2\left\|d^{\tilde{\pi}} - d^{\pi_E}\right\|_1 \tag{42}$$

$$\leq \mathbb{E}_{s,a\sim d^{\hat{\pi}}}\left[f'(s,a)\right] - \mathbb{E}_{s,a\sim d^{\pi_E}}\left[f'(s,a)\right] + 4\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}} + 2\left\|d^{\tilde{\pi}} - d^{\pi_E}\right\|_1 \tag{43}$$

$$\leq 3\left\|d^{\pi_E} - d^{\tilde{\pi}}\right\|_1 + 4\sqrt{\frac{\ln(|\mathcal{F}|/\delta)}{M}}. \tag{44}$$

where the first inequality uses the definition of $\bar{f}$, the second inequality uses the fact that $\hat{\pi}$ is the minimizer of $\max_{f\in\mathcal{F}}\mathbb{E}_{s,a\sim d^{\pi}} f(s,a) - \frac{1}{M}\sum_{i=1}^{M} f(s_i^*, a_i^*)$. We also use Hoeffding's inequality where $\forall f \in \mathcal{F}$,

$$\left|\mathbb{E}_{s,a\sim d^{\pi_E}} f(s,a) - \sum_{i=1}^{M} f(s_i^*, a_i^*)\right| \leq 2\sqrt{\frac{\ln\left(|\mathcal{F}|/\delta\right)}{M}} \tag{45}$$

with probability at least $1-\delta$. □

C MISSPECIFIED RL WITH EXPERT FEEDBACK

Theorem 2.1 establishes that polynomial sample complexity in the misspecified 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, even with access to a queryable expert policy like DAgger (Ross et al., 2011).

More specifically, Theorem 2.1 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 2.1 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.

Notably, this lower bound focuses on computational efficiency. Statistically efficient imitation *is* possible, and we present a statistically optimal imitation learning algorithm for the misspecified setting in Appendix B.

D FURTHER EXPLANATION OF GUITAR AND PSDP

Algorithm 3 Policy Search Via Dynamic Programming (Bagnell et al., 2003)

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

$$\pi_h \leftarrow \arg\max_{\pi' \in \Pi} \mathbb{E}_{s_h \sim \rho_h} \mathbb{E}_{a_h \sim \pi'(\cdot|s_h)} A_{r_i}^{\pi_{h+1}, \ldots, \pi_H}(s_h, a_h) \tag{46}$$

5: **end for**
6: **Return** $\pi = \{\pi_h\}_{h=1}^H$

Algorithm 4 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

```
// Loss function
```

$$\hat{L}(\pi, r) = \mathbb{E}_{(s,a)\sim\rho_E} r(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^\pi} r(s,a) \tag{47}$$

6: Optimize

```
// No-regret reward update
```

$$r_i \leftarrow \arg\max_{r \in \mathcal{R}} \hat{L}(\pi_{i-1}, r) + \eta^{-1} \Delta_R(r \mid r_{i-1}) \tag{48}$$

7: Optimize

```
// Expert-competitive response with RL
```

$$\pi_i \leftarrow \texttt{PSDP}(r = r_i, \rho = \rho_{\text{mix}}) \tag{49}$$

8: **end for**
9: **Return** π_i with lowest validation error

The full IRL procedure is outlined in Algorithm 4. It can be summarized as (1) a no-regret reward update using Online Mirror Descent, and (2) an expert-competitive policy update using Policy Search by Dynamic Programming (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 $\rho = \rho_E$). GUITAR can be seen as extending MMDP to a general reset distribution in the misspecified setting. We will focus on expert resets in the misspecified setting first, and we then consider other reset distributions in Section 4.

Policy Update. Following Ren et al. (2024)'s reduction of inverse RL to expert-competitive RL, we can use any RL algorithm to generate an expert-competitive response. We employ PSDP (Bagnell et al., 2003), shown in Algorithm 3, for its strong theoretical guarantees. In practice, any RL algorithm can be used, such as Soft Actor Critic (SAC, Haarnoja et al. (2018)).

Reward Update. We employ a no-regret update to the reward function. We employ Online Mirror Descent (Nemirovskij & Yudin, 1983; Beck & Teboulle, 2003; Srebro et al., 2011) for its strong theoretical guarantees, but in practice, any no-regret update can be used, such as gradient descent.

More specifically, the reward function is updated through Online Mirror Descent, such that

$$r_i \leftarrow \arg\max_{r \in \mathcal{R}} \hat{L}(\pi_{i-1}, r) + \eta^{-1} \Delta_R(r \mid r_{i-1}), \tag{50}$$

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

$$\hat{L}(\pi, r) = \mathbb{E}_{(s,a)\sim\rho_E} r(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^\pi} r(s,a), \tag{51}$$

with respect to the distribution of expert samples, ρ_E.

E PROOFS OF SECTION 3

E.1 PROOF OF THEOREM 3.3

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

$$V^{\pi_E} - V^{\overline{\pi}} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}_{\zeta\sim\pi_E}\sum_{h=1}^{H} r^*(s,a) - \mathbb{E}_{\zeta\sim\pi_i}\sum_{h=1}^{H} r^*(s,a)\right) \tag{52}$$

$$= H\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}_{(s,a)\sim d_\mu^{\pi_E}} r^*(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^{\pi_i}} r^*(s,a)\right) \tag{53}$$

$$= H\frac{1}{n}\sum_{i=1}^{n} L(\pi_i, r^*) \tag{54}$$

$$\leq H\frac{1}{n}\max_{r\in\mathcal{R}}\sum_{i=1}^{n} L(\pi_i, r) \tag{55}$$

$$\leq H\frac{1}{n}\max_{r\in\mathcal{R}}\sum_{i=1}^{n}\left(L(\pi_i, r) - L(\pi_i, r_i) + L(\pi_i, r_i)\right) \tag{56}$$

$$= H\frac{1}{n}\sum_{i=1}^{n} L(\pi_i, r_i) + H\frac{1}{n}\max_{r\in\mathcal{R}}\sum_{i=1}^{n}\left(L(\pi_i, r) - L(\pi_i, r_i)\right) \tag{57}$$

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

$$V^{\pi_E} - V^{\overline{\pi}} \leq H\frac{1}{n}\sum_{i=1}^{n} L(\pi_i, r_i) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{58}$$

$$= H\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{H}\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_E}} r_i(s_h,a_h) - \frac{1}{H}\sum_{h=1}^{H}\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_i}} r_i(s_h,a_h)\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{59}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{E}_{s\sim\mu} V_{r_i}^{\pi_E} - \mathbb{E}_{s\sim\mu} V_{r_i}^{\pi_i}\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{60}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\left(\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_E}} A_{r_i,h}^{\pi_i}(s_h,a_h)\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{61}$$

Focusing on the interior summation, we have

$$\sum_{h=0}^{H-1}\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_E}} A_h^{\pi_i}(s_h,a_h) \leq \sum_{h=0}^{H-1}\mathbb{E}_{s_h\sim d_h^{\pi_E}}\max_{a\in\mathcal{A}} A_h^{\pi_i}(s_h,a) \tag{62}$$

$$= \sum_{h=0}^{H-1}\mathbb{E}_{s_h\sim d_h^{\pi_E}}\max_{a\in\mathcal{A}} A_h^{\pi_i}(s_h,a) - \epsilon_{\Pi,h} + \epsilon_{\Pi,h} \tag{63}$$

$$= \sum_{h=0}^{H-1}\max_{\pi'\in\Pi}\mathbb{E}_{s_h\sim d_h^{\pi_E}}\mathbb{E}_{a\sim\pi'(\cdot|s)} A_h^{\pi_i}(s_h,a) + \epsilon_{\Pi,h} \tag{64}$$

$$\leq H^2\epsilon + H\epsilon_\Pi \tag{65}$$

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

Applying Equation 65 to Equation 61, we have

$$V^{\pi_E} - V^{\overline{\pi}} \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\left(\mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_E}} A_{r_i,h}^{\pi_i}(s_h,a_h)\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{66}$$

$$\leq \frac{1}{n}\sum_{i=1}^{n}\left(H^2\epsilon + H\epsilon_\Pi\right) + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{67}$$

$$\leq H^2\epsilon + H\epsilon_\Pi + H\sqrt{\frac{\ln|\mathcal{R}|}{n}} \tag{68}$$

which completes the proof. □

F PROOFS OF SECTION 4

F.1 LEMMAS OF THEOREM F.5

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

$$\hat{L}(\pi, r) = \mathbb{E}_{(s,a)\sim\rho_E} r(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^\pi} r(s,a). \tag{69}$$

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 I.2 with* $B = 1$*, the regret is given by*

$$\lambda_n = \sup_{r\in\mathcal{R}} \frac{1}{n}\sum_{i=1}^n \hat{L}(\pi_i, r) - \frac{1}{n}\sum_{i=1}^n \hat{L}(\pi_i, r_i) \tag{70}$$

$$\le \sqrt{\frac{2\ln|\mathcal{R}|}{n}} \tag{71}$$

$$= \sqrt{\frac{C_1}{n}}, \tag{72}$$

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

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

$$L(\pi, r) \le \hat{\mathrm{L}}(\pi, r) + \sqrt{\frac{C}{N}}, \tag{73}$$

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

$$\begin{aligned}
\left|L(\pi, r) - \hat{\mathrm{L}}(\pi, r)\right| &= \Big|\mathbb{E}_{(s,a)\sim d_\mu^{\pi_E}} r(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^\pi} r(s,a) \\
&\quad - \Big(\mathbb{E}_{(s,a)\sim\rho_E} r(s,a) - \mathbb{E}_{(s,a)\sim d_\mu^\pi} r(s,a)\Big)\Big| && (74)\\
&= \left|\mathbb{E}_{(s,a)\sim d_\mu^{\pi_E}} r(s,a) - \mathbb{E}_{(s,a)\sim\rho_E} r(s,a)\right| && (75)\\
&= \left|\mathbb{E}_{(s,a)\sim d_\mu^{\pi_E}} r(s,a) - \frac{1}{N}\sum_{(s_i,a_i)\in D_E}^{N} r(s_i,a_i)\right| && (76)\\
&\le \sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{R}|}{\delta}} && (77)\\
&\le \sqrt{\frac{C}{N}}, && (78)
\end{aligned}$$

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 I.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 F.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* $1-\delta$*,*

$$\mathbb{E}_{s\sim d^{\pi_E}} \max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \le \min\left\{\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N}}, C_B\left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right)\right\} \tag{79}$$

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

$$H\epsilon \geq \max_{\pi' \in \Pi} \mathbb{E}_{s_h \sim \rho_{mix,h}} \mathbb{E}_{a \sim \pi'(\cdot|s)} A_h^{\pi_i}(s_h, a) \tag{80}$$

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

$$\begin{aligned} H\epsilon &\geq \max_{\pi' \in \Pi} \mathbb{E}_{s \sim \rho_{\text{mix}}} \mathbb{E}_{a \sim \pi'(\cdot|s)} A^{\pi_i}(s, a) && (81)\\ &= \max_{\pi' \in \Pi} \mathbb{E}_{s \sim \rho_{\text{mix}}} \mathbb{E}_{a \sim \pi'(\cdot|s)} A^{\pi_i}(s, a) + \epsilon_{\Pi, r_i} - \epsilon_{\Pi, r_i} && (82)\\ &= \mathbb{E}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \epsilon_{\Pi, r_i} && (83) \end{aligned}$$

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

$$H\epsilon \geq \mathbb{E}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \epsilon_\Pi. \tag{84}$$

Rearranging the terms gives us

$$\begin{aligned} \mathbb{E}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) &\leq H\epsilon + \epsilon_\Pi && (85)\\ &= \epsilon_\Pi, && (86) \end{aligned}$$

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},

$$\mathbb{E}_{s \sim \rho_{\text{mix}}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) = \mathbb{E}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) + \mathbb{E}_{s \sim \rho_b} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a). \tag{87}$$

Consequently, we know that

$$\begin{aligned} \epsilon_\Pi &\geq \mathbb{E}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) && (88)\\ &= \frac{1}{N} \sum_{s_i \in D_E}^{N} \max_{a \in \mathcal{A}} A^{\pi_i}(s_i, a) && (89) \end{aligned}$$

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 I.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

$$\begin{aligned} \left| \mathbb{E}_{s \sim d_\mu^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) - \mathbb{E}_{s \sim \rho_E} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \right| &= \left| \mathbb{E}_{s \sim d_\mu^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \right. && (90)\\ &\quad \left. - \frac{1}{N} \sum_{s_i \in D_E}^{N} \max_{a \in \mathcal{A}} A^{\pi_i}(s_i, a) \right| && (91)\\ &\leq \sqrt{\epsilon_\Pi^2 \frac{1}{2N} \ln \frac{|\Pi||\mathcal{R}|}{\delta}} && (92)\\ &\leq \epsilon_\Pi \sqrt{\frac{C_0}{N}}, && (93) \end{aligned}$$

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 Equation 88 yields

$$\mathbb{E}_{s \sim d_\mu^{\pi_E}} \max_{a \in \mathcal{A}} A^{\pi_i}(s, a) \leq \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}. \tag{94}$$

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 I.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

$$\left|\mathbb{E}_{s\sim\nu}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) - \mathbb{E}_{s\sim\rho_{\text{mix}}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a)\right| = \left|\mathbb{E}_{s\sim\nu}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) - \frac{1}{N+M}\sum_{s_i\in D_{mix}}^{N+M}\max_{a\in\mathcal{A}} A^{\pi_i}(s_i,a)\right| \tag{95}$$

$$\leq\sqrt{\epsilon_\Pi\frac{1}{2(N+M)}\ln\frac{|\Pi||\mathcal{R}|}{\delta}} \tag{96}$$

$$\leq\epsilon_\Pi\sqrt{\frac{C_0}{N+M}} \tag{97}$$

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 Equation 85 yields

$$\mathbb{E}_{s\sim\nu}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a)\leq\epsilon_\Pi+\epsilon_\Pi\sqrt{\frac{C_0}{N+M}}. \tag{98}$$

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

$$\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) = \frac{N}{N+M}\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + \frac{M}{N+M}\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \tag{99}$$

$$\leq \frac{N}{N+M}\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + C_B\frac{M}{N+M}\mathbb{E}_{s\sim d^{\pi_B}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \tag{100}$$

$$\leq C_B\frac{N}{N+M}\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + C_B\frac{M}{N+M}\mathbb{E}_{s\sim d^{\pi_B}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \tag{101}$$

$$= C_B\left(\frac{N}{N+M}\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) + \frac{M}{N+M}\mathbb{E}_{s\sim d^{\pi_B}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a)\right) \tag{102}$$

$$\leq C_B\mathbb{E}_{s\sim\nu}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a). \tag{103}$$

Applying Equation 103 to Equation 98, we have

$$\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a)\leq C_B\mathbb{E}_{s\sim\nu}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a) \tag{104}$$

$$\leq C_B\left(\epsilon_\Pi+\epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right) \tag{105}$$

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

$$\mathbb{E}_{s\sim d^{\pi_E}}\max_{a\in\mathcal{A}} A^{\pi_i}(s,a)\leq\min\left\{\epsilon_\Pi+\epsilon_\Pi\sqrt{\frac{C_0}{N}}, C_B\left(\epsilon_\Pi+\epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right)\right\} \tag{106}$$

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 F.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 $1 - \delta$,*

$$\hat{\mathrm{L}}(\pi_i, r_i) \leq \min \left\{ \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}, C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right) \right\} + \sqrt{\frac{C}{N}}, \tag{107}$$

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 F.2, we have

$$\begin{aligned}
\hat{\mathrm{L}}(\pi_i, r_i) &\leq L(\pi_i, r_i) + \sqrt{\frac{C}{N}} && (108)\\
&= \mathbb{E}_{(s,a)\sim d_\mu^{\pi_E}} [r_i(s,a)] - \mathbb{E}_{(s,a)\sim d_\mu^{\pi_i}} [r_i(s,a)] + \sqrt{\frac{C}{N}} && (109)\\
&= \frac{1}{H} \left(V_{r_i}^{\pi_E} - V_{r_i}^{\pi_i} \right) + \sqrt{\frac{C}{N}} && (110)\\
&= \frac{1}{H} \left(\sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim d_h^{\pi_E}} A_{r_i,h}^{\pi_i}(s_h, a_h) \right) + \sqrt{\frac{C}{N}} && (111)\\
&\leq \frac{1}{H} \left(\sum_{h=1}^{H} \mathbb{E}_{s_h \sim d_h^{\pi_E}} \max_{a \in \mathcal{A}} A_{r_i,h}^{\pi_i}(s_h, a) \right) + \sqrt{\frac{C}{N}} && (112)\\
&= \frac{1}{H} \left(H \mathbb{E}_{s\sim d^{\pi_E}} \max_{a \in \mathcal{A}} A_{r_i}^{\pi_i}(s, a) \right) + \sqrt{\frac{C}{N}} && (113)
\end{aligned}$$

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 F.3, we have

$$\hat{\mathrm{L}}(\pi_i, r_i) \leq \min \left\{ \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}, C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right) \right\} + \sqrt{\frac{C}{MN}}, \tag{114}$$

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 $C = \ln \frac{2|\mathcal{R}|}{\delta}$. □

F.2 Finite Sample Analysis of Algorithm 2

Theorem F.5 (Sample Complexity of Algorithm 2)**.** *Suppose that PSDP's accuracy parameter is set to $\epsilon = 0$. Then, upon termination of Algorithm 2, with probability at least $1 - \delta$, we have*

$$V^{\pi_E} - V^{\overline{\pi}} \leq \underbrace{H \min\left\{\epsilon_\Pi^{\rho_{mix}} + \epsilon_\Pi^{\rho_{mix}}\sqrt{\frac{C_{\Pi,\mathcal{R}}}{N}}, C_B\left(\epsilon_\Pi^{\rho_{mix}} + \epsilon_\Pi^{\rho_{mix}}\sqrt{\frac{C_{\Pi,\mathcal{R}}}{N+M}}\right)\right\}}_{\textit{Misspecification Error}} + \underbrace{H\sqrt{\frac{C_{\mathcal{R}}}{N}}}_{\textit{Statistical Error}} + \underbrace{H\sqrt{\frac{\ln|\mathcal{R}|}{n}}}_{\textit{Reward Regret}} \tag{115}$$

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 F.5 upper bounds the sample complexity of Algorithm 2 in the offline data setting. The bound differs from Theorem 3.3 in the following ways. First, the policy optimization error term vanishes by the assumption that $\epsilon = 0$. Importantly, the assumption of $\epsilon = 0$ is not necessary but rather convenient, as the $\epsilon > 0$ case was presented in Theorem 3.3. Second, offline data is incorporated into the reset distribution, resulting in a modified misspecification error. Finally, the finite expert sample regime is considered, resulting in statistical error of estimating the expert policy's state distribution $d_\mu^{\pi_E}$ with the distribution over samples ρ_E.

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

$$V^{\pi_E} - V^{\overline{\pi}} = \frac{1}{n}\sum_{i=0}^{n}\left(\mathbb{E}_{\zeta\sim\pi_E}\left[\sum_{h=1}^{H} r^*(s_h, a_h)\right] - \mathbb{E}_{\zeta\sim\pi_i}\left[\sum_{h=1}^{H} r^*(s_h, a_h)\right]\right) \tag{116}$$

$$= \frac{1}{n}H\sum_{i=0}^{n}\left(\mathbb{E}_{(s,a)\sim d_\mu^{\pi_E}}\left[r^*(s,a)\right] - \mathbb{E}_{(s,a)\sim d_\mu^{\pi_i}}\left[r^*(s,a)\right]\right) \tag{117}$$

$$= \frac{1}{n}H\sum_{i=0}^{n} L(\pi_i, r^*) \tag{118}$$

$$\leq \frac{1}{n}H\max_{r\in\mathcal{R}}\sum_{i=0}^{n} L(\pi_i, r) \tag{119}$$

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 F.2), we have

$$V^{\pi_E} - V^{\overline{\pi}} \leq \frac{1}{n}H\max_{r\in\mathcal{R}}\sum_{i=0}^{n}\left(\hat{\mathrm{L}}(\pi_i, r) + \sqrt{\frac{C}{N}}\right) \tag{120}$$

$$= \frac{1}{n}H\max_{r\in\mathcal{R}}\sum_{i=0}^{n}\left(\hat{\mathrm{L}}(\pi_i, r) - \hat{\mathrm{L}}(\pi_i, r_i) + \hat{\mathrm{L}}(\pi_i, r_i) + \sqrt{\frac{C}{N}}\right) \tag{121}$$

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 F.1), we have

$$V^{\pi_E} - V^{\overline{\pi}} \leq \frac{1}{n}H\sum_{i=0}^{n}\left(\hat{\mathrm{L}}(\pi_i, r_i) + \sqrt{\frac{C}{N}}\right) + H\sqrt{\frac{C_1}{n}} \tag{122}$$

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

$$V^{\pi_E} - V^{\overline{\pi}} \leq \frac{1}{n}H\sum_{i=0}^{n}\left(\min\left\{\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N}}, C_B\left(\epsilon_\Pi + \epsilon_\Pi\sqrt{\frac{C_0}{N+M}}\right)\right\} + \sqrt{\frac{C}{N}}\right) + H\sqrt{\frac{C_1}{n}}, \tag{123}$$

which simplifies to

$$V^{\pi_E} - V^{\overline{\pi}} \leq H \min \left\{ \epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N}}, C_B \left(\epsilon_\Pi + \epsilon_\Pi \sqrt{\frac{C_0}{N+M}} \right) \right\} + H\sqrt{\frac{C}{N}}, + H\sqrt{\frac{C_1}{n}}, \quad (124)$$

where $C_B = \left\| \frac{d_\mu^{\pi_E}}{d_\mu^{\overline{\pi}_B}} \right\|_\infty$, 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 $C_1 = 2 \ln |\mathcal{R}|$. □

G 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 offline 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 offline 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.

We calculate the inter-quartile mean (IQM) and standard errors across 10 seeds for all experiments.

G.1 MuJoCo Tasks

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

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, this was 50 expert state-action pairs. For Humanoid, this was 100 expert state-action pairs. For Hopper, this was 600 expert state-action pairs.

Offline Data. We generate the offline 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 environment and $p^{\pi_B}_{\text{tremble}} = 0.05$ for the Hopper and Humanoid environments.

Discriminator. For our discriminator, we start with a learning rate of $8\text{e}{-4}$ 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. 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 offline 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.

G.2 D4RL Tasks

G.2.1 Antmaze-Large Tasks

For the two `Antmaze-Large` tasks, we use the data provided by Fu et al. (2020) as the expert demonstrations. We append goal information to the observation for all algorithms following Ren et al. (2024); Swamy et al. (2023). For our policy optimizer in every algorithm, we build upon the TD3+BC implementation of Fujimoto & Gu (2021) with the default hyperparameters.

Expert Data and Discriminator. We use the relevant D4RL dataset to learn the discriminator. For our discriminator, we start with a learning rate of $8e-3$ and decay it linearly over outer-loop iterations. We update the discriminator every 5,000 actor steps.

Baselines. For behavioral cloning, we run the TD3+BC optimizer for 500,000 steps while zeroing out the component of the actor update that depends on rewards. We use a reset proportion of $\alpha = 1.0$. We provide all runs with the same expert data. All IRL algorithms are pretrained with 10,000 steps of behavioral cloning on the expert dataset.

Reset Distributions. We reset `GUITAR` to various reset distributions. We consider the expert dataset, short trajectories (of length less than 500) in the expert dataset, and long trajectories (of length greater than or equal to 500) in the expert dataset. We use each of these datasets as different reset distributions. For the D4RL tasks, we perform true state resets (i.e. reset the learner to states in the reset distribution), rather than perform BC roll-ins as done in the MuJoCo tasks. Notably, IRL with resets to the true starting state distribution (i.e. no selective reset distribution) has been well studied by prior work (Swamy et al., 2023; Ren et al., 2024) and observed to not solve the Antmaze-Large tasks.

G.2.2 Antmaze-Umaze Tasks

For the two `Antmaze-Umaze` tasks, we train an RL expert using TD3+BC. We append goal information to the observation for all algorithms following Ren et al. (2024); Swamy et al. (2023). For our policy optimizer in every algorithm, we build upon the TD3+BC implementation of Fujimoto & Gu (2021) with the default hyperparameters.

Expert Data and Offline Data. We collect expert data by first training an RL expert using TD3+BC on the following maze map:

$$M_{\pi_E} := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & \mathbf{R} & 1 & \mathbf{G} & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

where **R** is the starting state and **G** is the goal state. The expert policy is then rolled out for 1,000,000 state-action samples.

We collect the offline data by training an RL agent using TD3+BC on the following maze map:

$$M_{\pi_B} := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & \mathbf{R} & 1 & \mathbf{G} & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

and then rolling out the policy for 1,000,000 state-action samples.

The learner is trained and evaluated on the map M_{π_B}. Notably, this difference ensures that the expert policy solves the maze via one path, and the learner must solve it in a different path, where the only shared states can be the start and goal states.

Discriminator. We use the expert data to learn the discriminator. For our discriminator, we start with a learning rate of $8e-3$ and decay it linearly over outer-loop iterations. We update the discriminator every 5,000 actor steps. Since this is a strongly misspecified setting, we only use the last state in a trajectory as the input to the discriminator for all IRL algorithms. The explanation for this design choice is that the learner's and the expert's behaviors may differ throughout the maze, but the only important criterion is whether they end in the same state (i.e. the goal state).

Baselines and Reset Distributions. For behavioral cloning, we run the TD3+BC optimizer for 500,000 steps while zeroing out the component of the actor update that depends on rewards. We use a reset proportion of $\alpha = 1.0$. We provide all runs with the same expert data. Due to the strong misspecification in this task, we do not pretrain the IRL algorithms with behavioral cloning.

`MM` is reset to the true starting state, while `FILTER` is reset to the expert data. `GUITAR` is reset to the offline data (i.e. π_B's data).

It should be noted that this is not the typical Antmaze-Umaze task, which can typically be solved by IRL approaches. In this setting, we consider the strongly misspecified setting, where the expert policy solves the maze one way, but the learner must solve it differently due to differences in the maze at "test-time." The degree of exploration difficulty due to the misspecification is likely the reason for the poor performance of the baselines.

H Further Analysis of Misspecification Settings

H.1 Misspecified Setting 1: Expert Cliff Walks

We theoretically analyze the setting of expert cliff walks below.

In the case of a perfect reset distribution on the best realizable policy $\pi^\star$, where $\rho = d_\mu^{\pi^\star}$ and $C_S = 1$, we can present a straightforward bound on the performance gap, focusing on misspecification error and optimization error.

Corollary H.1 (Performance Gap under Resets to $\pi^\star$)**.** *Consider the case of infinite expert data samples, such that $\rho_E = d_\mu^{\pi_E}$, perfect reset distribution coverage, such that $\rho = d_\mu^{\pi^\star}$, and perfect reward learning, such that the reward regret error is negligible. 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 2. Then, the performance gap is given by*

$$V^{\pi_E} - V^{\overline{\pi}} \leq \underbrace{V^{\pi_E} - V^{\pi^\star}}_{\textit{Misspecification Error}} + \underbrace{H^2\epsilon}_{\textit{Optimization Error}} \tag{125}$$

where H is the horizon and $\overline{\pi}$ is the trajectory-level average of the learned policies (i.e. π_i at each iteration $i \in [n]$ of Algorithm 2).

Corollary H.1 presents the performance gap with resets to the best realizable policy's state distribution. Notably, by considering the perfect scenario of infinite expert data, perfect reset distribution coverage, and perfect reward learning, our bound does not include the policy completeness error ϵ_Π and instead is an extension of Swamy et al. (2023)'s NRMM algorithm to the misspecified setting.

H.2 Misspecified Setting 3: Multiple Optimal Paths

In addition to the two misspecification settings presented in 4, we present a third one below.

An adaption of the first misspecification setting from 4 is the setting where the expert demonstrations contain multiple optimal approaches to solving the task. In this setting, it may be beneficial to the subset of the expert demonstrations where the policy class is rich, and, therefore, the expert's behavior in that state distribution is realizable. We can formalize this setting as:

$$C_E := \left\| \frac{\rho}{d_\mu^{\pi_E}} \right\|_\infty < \infty \tag{126}$$

In other words, the reset distribution ρ is a subset of the expert's state distribution.

H.3 Empirical Analysis of Misspecified Setting 3

Next, we consider the opposite approach to constructing the offline data and reset distribution. Rather than use an offline distribution that covers the expert, we instead select reset distributions that are subsets of the expert's distribution, in settings with access to a generative model (i.e. full state resets). More specifically, we isolate all short trajectories (D_{short}) and all long trajectories (D_{long}) in the expert dataset and use them for different reset distributions. We aim to isolate the effects of varying the reset distribution. To this end, we use the full D4RL dataset for discriminator learning to avoid finite sample errors in the reward update. We perform all runs with `GUITAR` using the varied reset distributions.

In Figure 5, we observe that using the short-trajectory dataset as the reset distribution performs comparably to the full expert dataset, from which we establish that full coverage of the expert's state distribution is not necesssary in the reset distribution. However, as demonstrated by the long-trajectory dataset failing to solve the problem, there are states in the expert's state distribution that are not beneficial to reset to. Crucially, we observe a difference in the coverage of the reset distributions when visualized (Figure **??**). Surprisingly, the short-trajectory datasets do not have greater coverage of states; they simply have more focused coverage (i.e. closer to goals).

The results suggest a larger takeaway for practioners: while the theoretical analysis mandates a reset distribution that covers the expert's state distribution, in practice, the concentrability coefficient is a conservative condition for good performance. In other words, not all expert states are equally

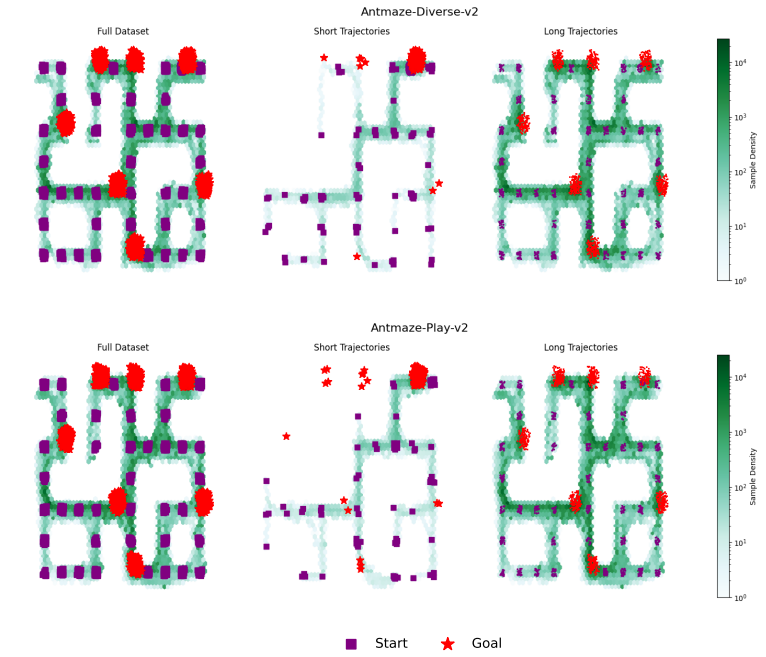

Figure 4: We train GUITAR with expert data from the full D4RL dataset (Fu et al., 2020) and vary the reset distribution, considering exclusively long trajectories (D_{long}, episodes longer than 500 steps), exclusively short trajectories (D_{short}, episodes shorter than 500 steps), the full dataset (D_{full}), and the true starting state (D_{start}). We observe that using a subset of the expert's data distribution (i.e. a reset distribution with imperfect coverage of the expert's data) can still match the performance of the best reset distribution. Standard errors are computed across 10 seeds.

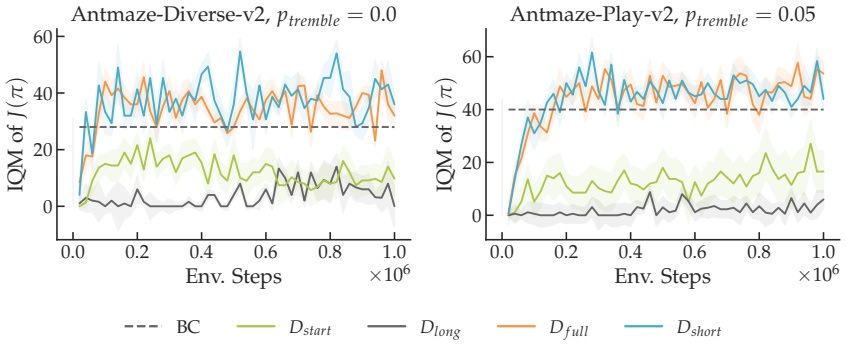

Figure 5: We train GUITAR with expert data from the full D4RL dataset (Fu et al., 2020) and vary the reset distribution, considering exclusively long trajectories (D_{long}, episodes longer than 500 steps), exclusively short trajectories (D_{short}, episodes shorter than 500 steps), the full dataset (D_{full}), and the true starting state (D_{start}). We observe that using a subset of the expert's data distribution (i.e. a reset distribution with imperfect coverage of the expert's data) can still match the performance of the best reset distribution. Standard errors are computed across 10 seeds.

valuable for the reset distribution. For the Antmaze tasks, this seems to be states that are close to accomplishing the goal. Moreover, the efficiency improvement gained by resets to expert states can be attained by resetting to a focused subset of the expert states. The empirical results mark a gap with the theory, motivating future research into alternative assumptions and conditions on the offline data and reset distributions.

I USEFUL LEMMAS

Theorem I.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 $1 - \delta$,*

$$|\overline{Z}_M - \mu| \leq \sqrt{\frac{c}{2M} \ln \frac{2}{\delta}} \tag{127}$$

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

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

$$\lambda_N = \frac{1}{N} \sum_{t=1}^{N} \ell(\hat{\mathbf{y}}_t, z_t) - \inf_{\mathbf{f} \in \mathcal{F}} \frac{1}{N} \sum_{t=1}^{N} \ell(\mathbf{f}, z_t). \tag{128}$$

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 $\eta = \sqrt{\frac{2 \sup_{\mathbf{f} \in \mathcal{F}} R(\mathbf{f})}{N B^2}}$, then,

$$\lambda_n \leq \sqrt{\frac{2B^2 \sup_{\mathbf{f} \in \mathcal{F}} R(\mathbf{f})}{N}}. \tag{129}$$

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