Provably Efficient Imitation Learning from Observation Alone

Wen Sun [1] **Anirudh Vemula** [1] **Byron Boots** [2] **J. Andrew Bagnell** [3]

Abstract

We study Imitation Learning (IL) from Observations alone (ILFO) in large-scale MDPs. While most IL algorithms rely on an expert to directly provide actions to the learner, in this setting the expert only supplies sequences of observations. We design a new model-free algorithm for ILFO, *Forward Adversarial Imitation Learning* (FAIL), which learns a sequence of time-dependent policies by minimizing an Integral Probability Metric between the observation distributions of the expert policy and the learner. FAIL is the *first* provably efficient algorithm in ILFO setting, which learns a near-optimal policy with a number of samples that is polynomial in all relevant parameters but independent of the number of unique observations. The resulting theory extends the domain of provably sample efficient learning algorithms beyond existing results, which typically only consider tabular reinforcement learning settings or settings that require access to a near-optimal reset distribution. We also demonstrate the efficacy of FAIL on multiple OpenAI Gym control tasks.

1. Introduction

Imitation Learning (IL) is a sample efficient approach to policy optimization (Ross et al., 2011; Ross & Bagnell, 2014; Sun et al., 2017) that has been extensively used in real applications, including Natural Language Processing (Daumé III et al., 2009; Chang et al., 2015b;a), game playing (Silver et al., 2016; Hester et al., 2017), system identification (Venkatraman et al., 2015; Sun et al., 2016), and robotics control tasks (Pan et al., 2018). Most previous IL work considers settings where an expert can directly provide action signals to the learner. In these settings, a general strategy is to directly learn a policy that maps from state to action, via supervised learning approaches (e.g., DAgger (Ross et al., 2011), AggreVaTe (Ross & Bagnell, 2014), Behaviour Cloning (Syed & Schapire, 2010)). Another popular strategy is to learn a policy by minimizing some divergence between the policy's state-action distribution and the expert's state-action distribution. Popular divergences include Forward KL (i.e., Behaviour Cloning), Jensen Shannon Divergence (e.g., GAIL (Ho & Ermon, 2016)).

Here, we consider a more challenging IL setting, where experts' demonstrations consist only of observations, no action or reward signals are available to the learner, and no reset is allowed (e.g., a robot learns a task by just watching an expert performing the task). We call this setting *Imitation Learning from Observations alone* (ILFO). Under this setting, without access to expert actions, approaches like DAgger, AggreVaTe, GAIL, and Behaviour Cloning by definition cannot work. Although recently several model-based approaches, which learn an inverse model that predicts the actions taken by an expert (Torabi et al., 2018; Edwards et al., 2018) based on successive observations, have been proposed, these approaches can suffer from covariate shift (Ross et al., 2011). While we wish to train a predictor that can infer an expert's actions accurately *under the expert's observation distribution*, we do not have access to actions generated by the expert conditioned on the expert's observation (See Section 6 for a more detailed discussion). An alternative strategy is to handcraft cost functions that use some distance metric to penalize deviation from the experts' trajectories (e.g., Liu et al. (2018); Peng et al. (2018)), which is then optimized by Reinforcement Learning (RL). These methods typically involve hand-designed cost functions that sometimes require prior task-specific knowledge (Peng et al., 2018). The quality of the learned policy is therefore completely dependent on the hand-designed costs which could be widely different from the true cost. Ideally, we would like to learn a policy that minimizes the unknown true cost function of the underlying MDP.

In this work, we explicitly consider learning near-optimal policies in a sample and computationally efficient manner. Specifically, we focus on large-scale MDPs where the number of unique observations is extremely large (e.g., high-dimensional observations such as raw-pixel images). Such large-scale MDP settings immediately exclude most existing sample efficient RL algorithms, which are often designed

[1]Robotics Institute, Carnegie Mellon University, USA [2]College of Computing, Georgia Institute of Technology, USA [3]Aurora Innovation, USA. Correspondence to: Wen Sun <wensun@cs.cmu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

for small tabular MDPs, whose sample complexities have a polynomial dependency on the number of observations and hence cannot scale well. To solve large-scale MDPs, we need to design algorithms that leverage function approximation for generalization. Specifically, we are interested in algorithms with the following three properties: (1) *near-optimal performance guarantees*, i.e., we want to search for a policy whose performance is close to the expert's in terms of the expected total cost of the underlying MDP (and not a hand-designed cost function); (2) *sample efficiency*, we require sample complexity that scales polynomially with respect to all relevant parameters (e.g., horizon, number of actions, statistical complexity of function approximators) except the cardinality of the observation space—hence excluding PAC RL algorithms designed for small tabular MDPs; (3) *computational efficiency*: we rely on the notion of oracle-efficiency (Agarwal et al., 2014) and require the number of efficient oracle calls to scale polynomially—thereby excluding recently proposed algorithms for Contextual Decision Processes which are not computationally efficient (Jiang et al., 2016; Sun et al., 2018). To the best of our knowledge, the desiderata above requires designing new algorithms.

With access to experts' trajectories of observations we introduce a model-free algorithm, called Forward Adversarial Imitation Learning (FAIL), that decomposes ILFO into H independent two-player min-max games, where H is the horizon length. We aim to learn a sequence of time-dependent policies from $h = 1$ to H, where at any time step h, the policy π_h is learned such that the generated observation distribution at time step $h+1$, conditioned on $\pi_1, \ldots, \pi_{h-1}$ being fixed, matches the expert's observation distribution at time step $h+1$, in terms of an Integral Probability Metric (IPM) (Müller et al., 1997). IPM is a family of divergences that can be understood as using a set of discriminators to distinguish two distributions (e.g., Wasserstein distance is one such special instance). We analyze the sample complexity of FAIL and show that FAIL can learn a near-optimal policy in sample complexity that does not explicitly depend on the cardinality of observation space, but rather only depends on the complexity measure of the policy class and the discriminator class. Hence FAIL satisfies the above mentioned three properties. The resulting theory extends the domain of provably sample efficient learning algorithms beyond existing results, which typically only consider tabular reinforcement learning settings (e.g., Dann & Brunskill (2015)) or settings that require access to a near-optimal reset distribution (e.g., Kakade & Langford (2002); Bagnell et al. (2004); Munos & Szepesvári (2008)). We also demonstrate that learning under ILFO can be exponentially more sample efficient than pure RL.We also study FAIL under three specific settings: (1) Lipschitz continuous MDPs, (2) Interactive ILFO where one can query the expert at any time during training, and (3) state abstraction. Finally, we demonstrate the efficacy of FAIL on multiple continuous control tasks.

2. Preliminaries

We consider an episodic finite horizon Decision Process that consists of $\{\mathcal{X}_h\}_{h=1}^H, \mathcal{A}, c, H, \rho, P$, where $\mathcal{X}_h$ for $h \in [H]$ is a time-dependent observation space,[1] $\mathcal{A}$ is a discrete action space such that $|\mathcal{A}| = K \in \mathbb{N}^+$, H is the horizon. We assume the cost function $c : \mathcal{X}_H \to \mathbb{R}$ is only defined at the last time step H (e.g., sparse cost), and $\rho \in \Delta(\mathcal{X}_1)$ is the initial observation distribution, and P is the transition model i.e., $P : \mathcal{X}_h \times \mathcal{A} \to \Delta(\mathcal{X}_{h+1})$ for $h \in [H-1]$. Note that here we assume the cost function only depends on observations. We assume that $\mathcal{X}_h$ for all $h \in [H]$ is discrete, but $|\mathcal{X}_h|$ is extremely large and hence any sample complexity that has polynomial dependency on $|\mathcal{X}_h|$ should be considered as sample inefficient, i.e., one cannot afford to visit every unique observation. We assume that the cost is bounded, i.e., for any sequence of observations, $c_H \leq 1$ (e.g., zero-one loss at the end of each episode). For a time-dependent policy $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_H\}$ with $\pi_h : \mathcal{X}_h \to \Delta(\mathcal{A})$, the value function $V_h^{\boldsymbol{\pi}} : \mathcal{X}_h \to [0,1]$ is defined as:

$$V_h^{\boldsymbol{\pi}}(x_h) = \mathbb{E}\left[c(x_H) | a_i \sim \pi_i(\cdot|x_i), x_{i+1} \sim P_{x_i,a_i}\right],$$

and state-action function $Q_h^{\boldsymbol{\pi}}(x_h, a_h)$ is defined as $Q_h^{\boldsymbol{\pi}}(x_h, a_h) = \mathbb{E}_{x_{h+1} \sim P_{x_h,a_h}}\left[V_{h+1}^{\boldsymbol{\pi}}(x_{h+1})\right]$ with $V_H^{\boldsymbol{\pi}}(x) = c(x)$. We denote $\mu_h^{\boldsymbol{\pi}}$ as the distribution over $\mathcal{X}_h$ at time step h following $\boldsymbol{\pi}$. Given H policy classes $\{\Pi_1, \ldots, \Pi_H\}$, the goal is to learn a $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_H\}$ with $\pi_h \in \Pi_h$, which minimizes the expected cost:

$$J(\boldsymbol{\pi}) = \mathbb{E}\left[c(x_H) | a_h \sim \pi_h(\cdot|x_h), x_{h+1} \sim P(\cdot|x_h, a_h)\right].$$

Denote $\mathcal{F}_h \subseteq \{f : \mathcal{X}_h \to \mathbb{R}\}$ for $h \in [H]$. We define a Bellman Operator Γ_h associated with the expert $\pi_h^\star$ at time step h as $\Gamma_h : \mathcal{F}_{h+1} \to \{f : \mathcal{X}_h \to \mathbb{R}\}$where for any $x_h \in \mathcal{X}_h, f \in \mathcal{F}_{h+1}$,

$$\left(\Gamma_h f\right)(x_h) \triangleq \mathbb{E}_{a_h \sim \pi_h^\star(\cdot|x_h), x_{h+1} \sim P_{x_h,a_h}}\left[f(x_{h+1})\right].$$

Notation For a function $f : \mathcal{X} \to \mathbb{R}$, we denote $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$, $\|f\|_L$ as the Lipschitz constant: $\|f\|_L = \sup_{x_1, x_2, x_1 \neq x_2} (f(x_1) - f(x_2))/d(x_1, x_2)$, with $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ being the metric in space $\mathcal{X}$.[2] We consider a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ defined with a positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to [0,1]$ such that

[1] we use the term observation throughout the paper instead of the term state as one would normally use in defining MDPs, for the purpose of sharply distinguishing our setting from tabular MDPs where $\mathcal{X}$ has very small number of states.

[2] A metric d (e.g., Euclidean distance) satisfies the following conditions: $d(x,y) \geq 0$, $d(x,y) = 0$ iff $x = y$, $d(x,y) = d(y,x)$ and d satisfies triangle inequality.

$\mathcal{H}$ is the span of $\{k(x,\cdot) : x \in \mathcal{X}\}$, and we have $f(x) = \langle f, k(x,\cdot)\rangle$, with $\langle k(x_1,\cdot), k(x_2,\cdot)\rangle \triangleq k(x_1,x_2)$. For any $f \in \mathcal{H}$, we define $\|f\|_{\mathcal{H}}^2 = \langle f, f\rangle$. We denote $U(\mathcal{A})$ as a uniform distribution over action set $\mathcal{A}$. For $N \in \mathbb{N}^+$, we denote $[N] \triangleq \{1, 2, \ldots, N\}$.

Integral Probability Metrics (IPM) (Müller, 1997) is a family of distance measures on distributions: given two distributions P_1 and P_2 over $\mathcal{X}$, and a function class $\mathcal{F}$ containing real-value functions $f : \mathcal{X} \to \mathbb{R}$ and symmetric (e.g., $\forall f \in \mathcal{F}$, we have $-f \in \mathcal{F}$), IPM is defined as:

$$\sup_{f\in\mathcal{F}} \left(\mathbb{E}_{x\sim P_1}[f(x)] - \mathbb{E}_{x\sim P_2}[f(x)]\right). \tag{1}$$

By choosing different class of functions $\mathcal{F}$, various popular distances can be obtained. For instance, IPM with $\mathcal{F} = \{f : \|f\|_\infty \le 1\}$ recovers Total Variation distance, IPM with $\mathcal{F} = \{f : \|f\|_L \le 1\}$ recovers Wasserstein distance, and IPM with $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \le 1\}$ with RKHS $\mathcal{H}$ reveals maximum mean discrepancy (MMD).

2.1. Assumptions

We first assume access to a Cost-Sensitive oracle and to a Linear Programming oracle.

Cost-Sensitive Oracle The Cost-Sensitive (CS) oracle takes a class of classifiers $\Pi \triangleq \{\pi : \mathcal{X} \to \Delta(\mathcal{A})\}$, a dataset consisting of pairs of feature x and cost vector $c \in \mathbb{R}^K$, i.e., $\mathcal{D} = \{x_i, c_i\}_{i=1}^N$, as inputs, and outputs a classifier that minimizes the average expected classification cost: $\sum_{i=1}^N \pi(\cdot|x_i)^\top c_i/N$.

Efficient cost sensitive classifiers exist (e.g., Beygelzimer et al. (2005)) and are widely used in sequential decision making tasks (e.g., Agarwal et al. (2014); Chang et al. (2015b)).

Linear Programming Oracle A Linear Programming (LP) oracle takes a class of functions $\mathcal{G}$ as inputs, optimizes a linear functional with respect to $g \in \mathcal{G}$: $\min_{g\in\mathcal{G}} \sum_{i=1}^N \alpha_i g(x_i)$.

When $\mathcal{G}$ is in RKHS with bounded norm, the linear functional becomes $\max_{g:\|g\|\le c}\langle g, \sum_{i=1}^n \alpha_i \phi(x_i)\rangle$, from which one can obtain the closed-form solution. Another example is when $\mathcal{G}$ consists of all functions with bounded Lipschitiz constant, i.e., $\mathcal{G} = \{g : \|g\|_L \le c\}$ for $c \in \mathbb{R}^+$, Sriperumbudur et al. (2012) showed that $\max_{g\in\mathcal{G}} \sum_{i=1}^n \alpha_i g(x_i)$ can be solved by Linear Programming with n many constraints, one for each pair (α_i, x_i). In Appendix E, we provide a new reduction to Linear Programming for $\mathcal{G} = \{g : \|g\|_L \le c_1, \|g\|_\infty \le c_2\}$ for $c_1, c_2 \in \mathbb{R}^+$.

The second assumption is related to the richness of the function class. We simply consider a time-dependent policy class Π_h and $\mathcal{F}_h$ for $h \in [H]$, and we assume realizability:

Assumption 2.1 (Realizability and Capacity of Function Class)**.** *We assume Π_h and $\mathcal{F}_h$ contains $\pi_h^\star$ and $V_h^\star$, i.e., $\pi_h^\star \in \Pi_h$ and $V_h^\star \in \mathcal{F}_h$, $\forall h \in [H]$. Further assume that for all h, $\mathcal{F}_h$ is symmetric, and Π_h and $\mathcal{F}_h$ is finite in size.*

Note that we assume Π_h and $\mathcal{F}_h$ to be discrete (but could be extremely large) for analysis simplicity. As we will show later, our bound scales *only logarithmically* with respect to the size of function class.

3. Algorithm

Our algorithm, Forward Adversarial Imitation Learning (FAIL), aims to learn a sequence of policies $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_H\}$ such that its value $J(\boldsymbol{\pi})$ is close to $J(\boldsymbol{\pi}^\star)$. Note that $J(\boldsymbol{\pi}) \approx J(\boldsymbol{\pi}^\star)$ does not necessarily mean that the state distribution of $\boldsymbol{\pi}$ is close to $\boldsymbol{\pi}^\star$. FAIL learns a sequence of policies with this property in a forward training manner. The algorithm learns π_i starting from $i = 1$. When learning π_i, the algorithm fixes $\{\pi_1, \ldots, \pi_{i-1}\}$, and solves a min-max game to compute π_i; it then proceeds to time step $i + 1$. At a high level, FAIL decomposes a sequential learning problem into H-many independent two-player min-max games, where each game can be solved efficiently via no-regret online learning. Below, we first consider how to learn π_h conditioned on $\{\pi_1, \ldots, \pi_{h-1}\}$ being fixed. We then present FAIL by chaining H min-max games together.

3.1. Learning One Step Policy via a Min-Max Game

Throughout this section, we assume $\{\pi_1, \ldots, \pi_{h-1}\}$ are learned already and fixed. The sequence of policies $\{\pi_1, \ldots, \pi_{h-1}\}$ for $h \ge 2$ determines a distribution $\nu_h \in \Delta(\mathcal{X}_h)$ over observation space $\mathcal{X}_h$ at time step h. Also expert policy $\boldsymbol{\pi}^\star$ naturally induces a sequence of observation distributions $\mu_h^\star \in \Delta(\mathcal{X}_h)$ for $h \in [H]$. The problem we consider in this section is to learn a policy $\pi_h \in \Pi_h$, such that the resulting observation distribution from $\{\pi_1, \ldots, \pi_{h-1}, \pi_h\}$ at time step $h + 1$ is close to the expert's observation distribution $\mu_{h+1}^\star$ at time step $h + 1$.

We consider the following IPM minimization problem. Given the distribution $\nu_h \in \Delta(\mathcal{X}_h)$, and a policy $\pi \in \Pi_h$, via the Markov property, we have the observation distribution at time step $h + 1$ conditioned on ν_h and π as $\nu_{h+1}(x) \triangleq \sum_{x_h,a_h} \nu_h(x_h)\pi(a_h|x_h)P(x|x_h,a_h)$ for any $x \in \mathcal{X}_{h+1}$. Recall that the expert observation distribution at time step $h + 1$ is denoted as $\mu_{h+1}^\star$. The IPM with $\mathcal{F}_{h+1}$ between ν_{h+1} and $\mu_{h+1}^\star$ is defined as:

$$d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu_{h+1}^\star) \triangleq \max_{f\in\mathcal{F}_{h+1}} \left(\mathbb{E}_{x\sim\nu_{h+1}}[f(x)] - \mathbb{E}_{x\sim\mu_{h+1}^\star}[f(x)]\right).$$

Note that $d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu_{h+1}^\star)$ is parameterized by π, and our goal is to minimize $d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu_{h+1}^\star)$ with respect to

Algorithm 1 Min-Max Game $(\mathcal{D}^\star, \mathcal{D}, \Pi, \mathcal{F}, T)$

1: Initialize $\pi^0 \in \Pi$
2: **for** $n = 1$ to T **do**
3: $\quad f^n = \arg\max_{f \in \mathcal{F}} u(\pi^n, f)$ (LP Oracle)
4: $\quad u^n = u(\pi^n, f^n)$
5: $\quad \pi^{n+1} = \arg\min_{\pi \in \Pi} \sum_{t=1}^n u(\pi, f^t) + \phi(\pi)$ (Regularized CS Oracle)
6: **end for**
7: **Output**: $\pi^{n^\star}$ with $n^\star = \arg\min_{n \in [T]} u^n$

π over Π_h. However, $d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1})$ is not measurable directly as we do not have access to $\mu^\star_{h+1}$ but only samples from $\mu^\star_{h+1}$. To estimate $d_{\mathcal{F}_{h+1}}$, we draw a dataset $\mathcal{D} = \left\{(x^i_h, a^i_h, x^i_{h+1})\right\}_{i=1}^N$ such that $x^i_h \sim \nu_h$, $a^i_h \sim U(\mathcal{A})$, $x^i_{h+1} \sim P(\cdot|x^i_h, a^i_h)$, together with observation set resulting from expert $\mathcal{D}^\star = \{\tilde{x}^i_{h+1}\}_{i=1}^{N'} \overset{iid}{\sim} \mu^\star_{h+1}$, we form the following empirical estimation of $d_{\mathcal{F}_{h+1}}$ for any π, via importance weighting (recall $a^i_h \sim U(\mathcal{A})$):

$$\widehat{d}_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1}) \triangleq \tag{2}$$
$$\max_{f \in \mathcal{F}_{h+1}} \left(\frac{1}{N} \sum_{i=1}^N \frac{\pi(a^i_h|x^i_h)}{1/K} f(x^i_{h+1}) - \frac{1}{N'} \sum_{i=1}^{N'} f(\tilde{x}^i_{h+1}) \right),$$

where recall that $K = |\mathcal{A}|$ and the importance weight $K\pi(a^i_h|x^i_h)$ is used to account for the fact that we draw actions uniformly from $\mathcal{A}$ but want to evaluate π. Though due to the max operator, $\widehat{d}_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1})$ is not an unbiased estimate of $d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1})$, in Appendix A, we show that $\widehat{d}_{\mathcal{F}_{h+1}}$ indeed is a good approximation of $d_{\mathcal{F}_{h+1}}$ via an application of the standard Bernstein's inequality and a union bound over $\mathcal{F}_{h+1}$. Hence we can approximately minimize $\widehat{d}_{\mathcal{F}_{h+1}}$ with respect to π: $\min_{\pi \in \Pi} \widehat{d}_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1})$, resulting in a two-player min-max game. Intuitively, we can think of π as a generator, such that, conditioned on ν_h, it generates next-step samples x_{h+1} that are similar to the expert samples from $\mu^\star_{h+1}$, via fooling discriminators $\mathcal{F}_{h+1}$.

Note that the above formulation is a two-player game, with the utility function for π and f defined as:

$$u(\pi, f) \triangleq \sum_{i=1}^N K\pi(a^i_h|x^i_h) f(x^i_{h+1})/N - \sum_{i=1}^{N'} f(\tilde{x}^i_{h+1})/N'. \tag{3}$$

Algorithm 1 solves the minmax game $\min_\pi \max_f u(\pi, f)$ using no-regret online update on both f and π. At iteration n, player f plays the best-response via $f_n = \arg\max_f u(\pi^n, f)$ (Line 3) and player π plays the Follow-the-Regularized Leader (FTRL) (Shalev-Shwartz et al., 2012) as $\pi^{n+1} = \sum_{t=1}^n u(\pi, f^t) + \phi(\pi)$ with ϕ being convex regularization (Line 5). Note that other no-regret online learning algorithms (e.g., replacing FTRL by incremental online learning algorithms like OGD (Zinkevich, 2003) can also be used to approximately optimize the above min-max formulation. After the end of Algorithm 1, we output a policy π among all computed policies $\{\pi^i\}_{i=1}^T$ such that $\widehat{d}_{\mathcal{F}_{h+1}}(\pi|\nu_n, \mu^\star_{n+1})$ is minimized (Line 7).

Regarding the computation efficiency of Algorithm 1, the best response computation on f in Line 3 can be computed by a call to the LP Oracle, while FTRL on π can be implemented by a call to the regularized CS Oracle. Regarding the statistical performance, we have the following theorem:

Theorem 3.1. *Given* $\epsilon \in (0, 1], \delta \in (0, 1]$, *set* $T = \Theta\left(\frac{4K^2}{\epsilon^2}\right)$, $N = N' = \Theta\left(\frac{K \log(|\Pi_h||\mathcal{F}_{h+1}|/\delta)}{\epsilon^2}\right)$, *Algorithm 1 outputs* π *such that with probability at least* $1 - \delta$,

$$\left| d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1}) - \min_{\pi' \in \Pi_h} d_{\mathcal{F}_{h+1}}(\pi'|\nu_h, \mu^\star_{h+1}) \right| \leq O(\epsilon).$$

The proof of the above theorem is included in Appendix A, which combines standard min-max theorem and uniform convergence analysis. The above theorem essentially shows that Algorithm 1 successfully finds a policy π whose resulting IPM is close to the smallest possible IPM one could achieve if one had access to $d_{\mathcal{F}_{h+1}}(\pi|\nu_h, \mu^\star_{h+1})$ directly, up to ϵ error. Intuitively, from Theorem 3.1, we can see that if ν_h—the observation distribution resulting from fixed policies $\{\pi_1, \ldots, \pi_{h-1}\}$, is similar to $\mu^\star_h$, then we guarantee to learn a policy π, such that the new sequence of policies $\{\pi_1, \ldots, \pi_{h-1}, \pi\}$ will generate a new distribution ν_{h+1} that is close to $\mu^\star_{h+1}$, in terms of IPM with $\mathcal{F}_{h+1}$. The algorithm introduced below is based on this intuition.

3.2. Forward Adversarial Imitation Learning

Theorem 3.1 indicates that conditioned on $\{\pi_1, \ldots, \pi_{h-1}\}$ being fixed, Algorithm 1 finds a policy $\pi \in \Pi_h$ such that it approximately minimizes the divergence—measured under IPM with $\mathcal{F}_{h+1}$, between the observation distribution ν_{h+1} resulting from $\{\pi_1, \ldots, \pi_{h-1}, \pi\}$, and the corresponding distribution $\mu^\star_{h+1}$ from expert.

With Algorithm 1 as the building block, we now introduce our model-free algorithm—Forward Adversarial Imitation Learning (FAIL) in Algorithm 2. Algorithm 2 integrates Algorithm 1 into the Forward Training framework (Ross & Bagnell, 2010), by decomposing the sequential learning problem into H many independent distribution matching problems where each one is solved using Algorithm 1 independently. Every time step h, FAIL assumes that $\pi_1, \ldots, \pi_{h-1}$ have been correctly learned in the sense that the resulting observation distribution ν_h from $\{\pi_1, \ldots, \pi_{h-1}\}$ is close to $\mu^\star_h$ from expert. Therefore, FAIL is only required to focus on learning π_h correctly conditioned on $\{\pi_1, \ldots, \pi_{h-1}\}$ being fixed, such that ν_{h+1} is close to $\mu^\star_{h+1}$, in terms of the IPM with $\mathcal{F}_{h+1}$. Intuitively, when one has a strong class of discriminators, and the two-

Algorithm 2 FAIL($\{\Pi_h\}_h, \{\mathcal{F}_h\}_h, \epsilon, n, n', T$)

Set $\boldsymbol{\pi} = \emptyset$
for $h = 1$ to $H - 1$ **do**
  Extract expert's data at $h+1$: $\tilde{\mathcal{D}} = \{\tilde{x}^i_{h+1}\}_{i=1}^{n'}$
  $\mathcal{D} = \emptyset$
  for $i = 1$ to n **do**
    Reset $x_1^{(i)} \sim \rho$
    Execute $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_{h-1}\}$ to generate state x^i_h
    Execute $a^i_h \sim U(\mathcal{A})$ to generate x^i_{h+1} and add $(x^i_h, a^i_h, x^i_{h+1})$ to $\mathcal{D}$
  end for
  Set π_h to be the return of Algorithm 1 with inputs $\left(\tilde{\mathcal{D}}, \mathcal{D}, \Pi_h, \mathcal{F}_{h+1}, T\right)$
  Append π_h to $\boldsymbol{\pi}$
end for

player game in each time step is solved near optimally, then by induction from $h = 1$ to H, FAIL should be able to learn a sequence of policies such that ν_h is close to $\mu^\star_h$ for all $h \in [H]$ (for the base case, we simply have $\nu_1 = \mu^\star_1 = \rho$).

3.3. Analysis of Algorithm 2

The performance of FAIL crucially depends on the capacity of the discriminators. Intuitively, discriminators that are too strong cause overfitting (unless one has extremely large number of samples). Conversely, discriminators that are too weak will not be able to distinguish ν_h from $\mu^\star_h$. This dilemma was studied in the Generative Adversarial Network (GAN) literature already by Arora et al. (2017). Below we study this tradeoff explicitly in IL.

To quantify the power of discriminator class $\mathcal{F}_h$ for all h, we use *inherent Bellman Error* (IBE) with respect to $\boldsymbol{\pi}^\star$:

$$\epsilon_{\text{be}} = \max_h \left(\max_{g \in \mathcal{F}_{h+1}} \min_{f \in \mathcal{F}_h} \|f - \Gamma_h g\|_\infty \right). \tag{4}$$

The Inherent Bellman Error is commonly used in approximate value iteration literature (Munos, 2005; Munos & Szepesvári, 2008; Lazaric et al., 2016) and policy evaluation literature (Sutton, 1988). It measures the worst possible projection error when projecting $\Gamma_h g$ to function space $\mathcal{F}_h$. Intuitively increasing the capacity of $\mathcal{F}_h$ reduces ϵ_{be}.

Using a restricted function class $\mathcal{F}$ potentially introduces ϵ_{be}, hence one may tend to set $\mathcal{F}_h$ to be infinitely powerful discriminator class such as function class consisting of all bounded functions $\{f : \|f\|_\infty \le c\}$ (recall IPM becomes total variation in this case). However, using $\mathcal{F}_h \triangleq \{f : \|f\|_\infty \le c\}$ makes efficient learning impossible. The following proposition excludes the possibility of sample efficiency with discriminator class being $\{f : \|f\|_\infty \le c\}$.

Theorem 3.2 (Infinite Capacity $\mathcal{F}$ does not generalize)**.** *There exists a MDP with $H = 2$, a policy set $\Pi = \{\boldsymbol{\pi}, \boldsymbol{\pi}'\}$, an expert policy $\boldsymbol{\pi}^\star$ with $\boldsymbol{\pi} = \boldsymbol{\pi}^\star$ (i.e., Π is realizable), such that for datasets $\mathcal{D}^\star = \{\tilde{x}^i_2\}_{i=1}^M$ with $\tilde{x}^i_2 \sim \mu^\star_2$, $\mathcal{D} = \{x^i_2\}_{i=1}^M$ with $x^i_2 \sim \mu^{\boldsymbol{\pi}}_2$, and $\mathcal{D}' = \{x'^{(i)}_2\}_{i=1}^M$ with $x'^{(i)}_2 \sim \mu^{\boldsymbol{\pi}'}_2$, as long as $M = O(\log(|\mathcal{X}|))$, we must have:*

$$\lim_{|\mathcal{X}|\to\infty} \mathrm{P}(\mathcal{D}^\star \cap \mathcal{D} = \emptyset) = 1, \quad \lim_{|\mathcal{X}|\to\infty} \mathrm{P}(\mathcal{D}^\star \cap \mathcal{D}' = \emptyset) = 1.$$

Namely, denote $\hat{\mathcal{D}}$ as the empirical distribution of a dataset $\mathcal{D}$ by assigning probability $1/|\mathcal{D}|$ to any sample, we have:

$$\lim_{|\mathcal{X}|\to\infty} \|\hat{\mathcal{D}}^\star - \hat{\mathcal{D}}\|_1 = 2, \lim_{|\mathcal{X}|\to\infty} \|\hat{\mathcal{D}}^\star - \hat{\mathcal{D}}'\|_1 = 2.$$

The above theorem shows by just looking at the samples generated from $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, and comparing them to the samples generated from the expert policy $\boldsymbol{\pi}^\star$ using $\{f : \|f\|_\infty \le c\}$ (IPM becomes Total variation here), we cannot distinguish $\boldsymbol{\pi}$ from $\boldsymbol{\pi}'$, as they both look similar to $\boldsymbol{\pi}^\star$, i.e., none of the three datasets overlap with each other, resulting the TV distances between the empirical distributions become constants, *unless* the sample size scales $\Omega(\text{poly}(|\mathcal{X}|))$.

Theorem 3.2 suggests that one should explicitly regularize discriminator class so that it has finite capacity (e.g., bounded VC or Rademacher Complexity). The restricted discriminator class $\mathcal{F}$ has been widely used in practice as well such as learning generative models (i.e., Wasserstain GANs (Arjovsky et al., 2017)). Denote $|\Pi| = \max_h |\Pi_h|$ and $|\mathcal{F}| = \max_h |\mathcal{F}_h|$. The following theorem shows that the learned time-dependent policies $\boldsymbol{\pi}$'s performance is close to the expert's performance:

Theorem 3.3 (Sample Complexity of FAIL)**.** *Under Assumption 2.1, for any $\epsilon, \delta \in (0, 1]$, set $T = \Theta(\frac{K}{\epsilon^2})$, $n = n' = \Theta(\frac{K \log(|\Pi||\mathcal{F}|H/\delta)}{\epsilon^2})$, with probability at least $1 - \delta$,* FAIL *(Algorithm 2) outputs $\boldsymbol{\pi}$, such that,*

$$J(\boldsymbol{\pi}) - J(\boldsymbol{\pi}^\star) \le H^2 \epsilon'_{\text{be}} + H^2 \epsilon,$$

by using $\tilde{O}\left(\frac{HK}{\epsilon^2} \log\left(\frac{|\Pi||\mathcal{F}|}{\delta}\right)\right)$ [3] many trajectories with an average inherent Bellman Error ϵ'_{be}:

$$\epsilon'_{\text{be}} \triangleq \max_h \max_{g \in \mathcal{F}_{h+1}} \min_{f \in \mathcal{F}_h} \mathbb{E}_{x \sim (\mu^{\boldsymbol{\pi}}_h + \mu^\star_h)/2} [|f(x) - (\Gamma_h g)(x)|].$$

Note that the average inherent Bellman error ϵ'_{be} defined above is averaged over the state distribution of the learned policy $\boldsymbol{\pi}$ and the state distribution of the expert, which is guaranteed to be smaller than the classic inherent Bellman error used in RL literature (i.e., (4)) which uses infinity norm over $\mathcal{X}$. The proof of Theorem 3.3 is included in Appendix C. Regarding computational complexity of Algorithm 2, we can see that it requires polynomial number of

[3] In $\tilde{O}$, we drop log terms that does not dependent on $|\Pi|$ or $|\mathcal{F}|$. In Θ we drop constants that do not depend on $H, K, |\mathcal{X}|, |\Pi|, |\mathcal{F}|, 1/\epsilon, 1/\delta$. Details can be found in Appendix.

calls (with respect to parameters $H, K, 1/\epsilon$) to the efficient oracles (Regularized CS oracle and LP oracle). Since our analysis only uses uniform convergence analysis and standard concentration inequalities, extension to continuous Π and $\mathcal{F}$ with complexity measure such as VC-dimension, Rademacher complexity, and covering number is standard. We give an example in Section 5.1.

4. The Gap Between ILFO and RL

To quantify the gap between RL and ILFO, below we present an exponential separation between ILFO and RL in terms of sample complexity to learn a near-optimal policy. We assume that the expert policy is optimal.

Proposition 4.1 (Exponential Separation Between RL and ILFO)**.** *Fix $H \in \mathbb{N}^+$ and $\epsilon \in (0, \sqrt{1/8})$. There exists a family of MDP with deterministic dynamics, with horizon H, $2^H - 1$ many states, two different actions, such that for any RL algorithm, the probability of outputting a policy $\hat{\boldsymbol{\pi}}$ with $J(\hat{\boldsymbol{\pi}}) \leq J(\boldsymbol{\pi}^\star) + \epsilon$ after collecting T trajectories is at most $2/3$ for all $T \leq O(2^H/\epsilon^2)$. On the other hand, for the same MDP, given one trajectory of observations $\{\tilde{x}_h\}_{h=1}^H$ from the expert policy $\boldsymbol{\pi}^\star$, there exists an algorithm that deterministically outputs $\boldsymbol{\pi}^\star$ after collecting $O(H)$ trajectories.*

Proposition 4.1 shows having access to expert's trajectories of observations allows us to efficiently solve some MDPs that are otherwise provably intractable for any RL algorithm (i.e., requiring exponentially many trajectories to find a near optimal policy). This kind of exponential gap previously was studied in the interactive imitation learning setting where the expert also provides action signals (Sun et al., 2017) and one can query the expert's action at any time step during training. To the best of our knowledge, this is the *first exponential gap* in terms of sample efficiency between ILFO and RL. Note that our theorem applies to a specific family of purposefully designed MDPs, which is standard for proving information-theoretical lower bounds.

5. Case Study

In this section, we study three settings where inherent Bellman Error will disappear even under restricted discriminator class : (1) Lipschitz Continuous MDPs (e.g., (Kakade et al., 2003)), (2) Interactive Imitation Learning from Observation where expert is available to query during training, and (3) state abstraction.

5.1. Lipschitz Continuous MDPs

We consider a setting where cost functions, dynamics and $\pi_h^\star$ are Lipschitz continuous in metric space $(\mathcal{X}, d)$:

$$\|P(\cdot|x,a) - P(\cdot|x',a)\|_1 \leq L_P d(x,x'),$$
$$\|\pi_h^\star(\cdot|x) - \pi_h^\star(\cdot|x')\|_1 \leq L_\pi d(x,x'),$$

for the known metric d and Lipschitz constants L_P, L_π. Under this condition, the Bellman operator with respect to $\boldsymbol{\pi}^\star$ is Lipschitz continuous in the metric space $(\mathcal{X}, d)$: $|\Gamma_h f(x_1) - \Gamma_h f(x_2)| \leq (\|f\|_\infty(L_P + L_\pi))d(x_1, x_2)$, where we applied Holder's inequality. Hence, we can design the function class $\mathcal{F}_h$ for all $h \in [H]$ as follows:

$$\mathcal{F}_h = \{f : \|f\|_L \leq (L_P + L_\pi), \|f\|_\infty \leq 1\}, \quad (5)$$

which will give us $\epsilon_{\text{be}} = 0$ and $V_h^\star \in \mathcal{F}_h$ due to the assumption on the cost function. Namely $\mathcal{F}_h$ is the class of functions with bounded Lipschitz constant and bounded value. This class of functions is widely used in practice for learning generative models (e.g., Wasserstain GAN). Note that this setting was also studied in (Munos & Szepesvári, 2008) for the Fitted Value Iteration algorithm.

Denote $L \triangleq L_P + L_\pi$. For $\mathcal{F} = \{f : \|f\|_L \leq L, \|f\|_\infty \leq 1\}$ we show that we can evaluate the empirical IPM $\sup_{f\in\mathcal{F}} \left(\sum_{i=1}^N f(x_i)/N - \sum_{i=1}^{N'} f(x_i')/N'\right)$ by reducing it to Linear Programming, of which the details are deferred to Appendix E. Regarding the generalization ability, note that our function class $\mathcal{F}$ is a subset of all functions with bounded Lipschitz constant, i.e., $\mathcal{F} \subset \{f : \|f\|_L \leq L\}$. The Rademacher complexity for bounded Lipschitz function class grows in the order of $O(N^{-1/\text{cov}(\mathcal{X})})$ (e.g., see (Luxburg & Bousquet, 2004; Sriperumbudur et al., 2012)), with $\text{cov}(\mathcal{X})$ being the covering dimension of the metric space $(\mathcal{X}, d)$.[4] Extending Theorem 3.3 to Lipschitz continuous MDPs, we have the following corollary.

Corollary 5.1 (Sample Complexity of FAIL for Lipschitz Continuous MDPs)**.** *With the above set up on Lipschitz continuous MDP and $\mathcal{F}_h$ for $h \in [H]$ (5), given $\epsilon, \delta \in (0,1]$, set $T = \Theta(\frac{K}{\epsilon^2})$, $n = n' = \tilde{\Theta}(\frac{K(LK)^{\text{cov}(\mathcal{X})}\log(|\Pi|/\delta)}{\epsilon^{2+\text{cov}(\mathcal{X})}})$, then with probability at least $1-\delta$,* FAIL *(Algorithm 2) outputs a policy with $J(\boldsymbol{\pi}) - J(\boldsymbol{\pi}^\star) \leq O(H^2\epsilon)$ using at most $\tilde{O}\left(\frac{HK(KL)^{\text{cov}(\mathcal{X})}}{\epsilon^{(2+\text{cov}(\mathcal{X}))}}\log\left(\frac{|\Pi|}{\delta\epsilon}\right)\right)$ many trajectories.*

The proof of the above corollary is deferred to Appendix F which uses a standard covering number argument over $\mathcal{F}_h$ with norm $\|\cdot\|_\infty$. Note that we get rid of IBE here and hence as the number of sample grows, FAIL approaches to the global optimality. Though the bound has an exponential dependency on the covering dimension, note that the covering dimension $\text{cov}(\mathcal{X})$ is completely dependent on the underlying metric space $(\mathcal{X}, d)$ and could be much smaller than the real dimension of $\mathcal{X}$. Note that the above theorem also serves an example regarding how we can extend Theorem 3.3 to settings where $\mathcal{F}$ contains infinitely many functions but with bounded statistical complexity (similar techniques can be used for Π as well).

[4] Covering dimension is defined as $\text{cov}(\mathcal{X}) \triangleq \inf_{d>0}\{N_\epsilon(\mathcal{X}) \leq \epsilon^{-d}, \forall \epsilon > 0\}$, where $N_\epsilon(\mathcal{X})$ is the size of the minimum ϵ-net of metric space $(\mathcal{X}, \mathcal{D})$.

5.2. Interactive Imitation Learning from Observations

We can avoid IBE in an interactive learning setting, where we assume that we can query expert during training. But different from previous interactive imitation learning setting such as AggreVaTe, LOLS (Ross & Bagnell, 2014; Chang et al., 2015b), and DAgger (Ross et al., 2011), here we do not assume that expert provides actions nor cost signals. Given any observation x, we simply ask expert to take over for just one step, and observe the observation at the next step, i.e., $x' \sim P(\cdot|x,a)$ with $a \sim \pi^\star(\cdot|x)$. Note that compared to the non-interactive setting, interactive setting assumes a much stronger access to expert. In this setting, we can use arbitrary class of discriminators with bounded complexity. Due to space limit, we defer the detailed description of the interactive version of FAIL (Algorithm 5) to Appendix G. The following theorem states that we can avoid IBE:

Theorem 5.2. *Under Assumption 2.1 and the existence of an interactive expert, for any $\epsilon \in (0,1]$ and $\delta \in (0,1]$, set $T = \Theta(\frac{K}{\epsilon^2})$, $n = \Theta(\frac{K\log(|\Pi||\mathcal{F}|H)/\delta}{\epsilon^2})$, with probability at least $1-\delta$, Algorithm 5) outputs a policy $\boldsymbol{\pi}$ such that:*

$$J(\boldsymbol{\pi}) - J(\boldsymbol{\pi}^\star) \leq H\epsilon,$$

by using at most $\tilde{O}\left(\frac{HK}{\epsilon^2}\log\left(\frac{|\Pi||\mathcal{F}|}{\delta}\right)\right)$ many trajectories.

Compare to the non-interactive setting, we get rid of IBE, at the cost of a much stronger expert.

5.3. State Abstraction

Denote $\phi : \mathcal{X} \to \mathcal{S}$ as the abstraction that maps $\mathcal{X}$ to a discrete set $\mathcal{S}$. We assume that the abstraction satisfies the Bisimulation property (Givan et al., 2003): for any $x, x' \in \mathcal{X}$, if $\phi(x) = \phi(x')$, we have that $\forall s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$:

$$c(x) = c(x'), \quad \pi_h^\star(a|x) = \pi_h^\star(a|x')$$
$$\sum_{x'' \in \phi^{-1}(s)} P(x''|x,a) = \sum_{x'' \in \phi^{-1}(s)} P(x''|x',a). \quad (6)$$

In this case, one can show that the $V^\star$ is piece-wise constant under the partition induced from ϕ, i.e., $V^\star(x) = V^\star(x')$ if $\phi(x) = \phi(x')$. Leveraging the abstraction ϕ, we can then design $\mathcal{F}_h = \{f : \|f\|_\infty \leq 1, f(x) = f(x'), \forall x, x', \text{s.t.}, \phi(x) = \phi(x')\}$. Namely, $\mathcal{F}_h$ contains piece-wise constant functions with bounded values. Under this setting, we can show that the inherent Bellman Error is zero as well (see Proposition 9 from Chen & Jiang (2019)). Also $\sup_{f\in\mathcal{F}_h} u(\pi, f)$ can be again computed via LP and $\mathcal{F}_h$ has Rademacher complexity scales $O(\sqrt{|\mathcal{S}|/N})$ with N being the number of samples. Details are in Appendix I.

6. Discussion on Related Work

Some previous works use the idea of learning an inverse model to predict actions (or latent causes) (Nair et al., 2017; Torabi et al., 2018) from two successive observations and then use the learned inverse model to generate actions using expert observation demonstrations. With the inferred actions, it reduces the problem to normal imitation learning. We note here that learning an inverse model is ill-defined. Specifically, simply by the Bayes rule, the inverse model $P(a|x_h, x_{h+1})$—the probability of action a was executed at x_h such that the system generated x_{h+1}, is equivalent to $P(a|x_h, x_{h+1}) \propto P(x_{h+1}|x_h, a)P(a|x_h)$, i.e., an inverse model $P(a|x_h, x_{h+1})$ is explicitly dependent on the action generation policy $P(a|x_h)$. Unlike $P(x_{h+1}|x_h, a)$, without the policy $P(a|x_h)$, the inverse model is ill-defined by itself alone. This means that if one wants to learn an inverse model that predicts expert actions along the trajectory of observations generated by the expert, one would need to learn an inverse model, denoted as $P^\star(a|x_h, x_{h+1})$, such that $P^\star(a|x_h, x_{h+1}) \propto P(x_{h+1}|x_h, a)\pi_h^\star(a|x_h)$, which indicates that one needs to collect actions from $\pi_h^\star$. An inverse model makes sense when the dynamics is deterministic and bijective. Hence replying on learning such an inverse model $P^\star(a|x, x')$ will not provide any performance guarantees in general, unless we have actions collected from $\pi^\star$.

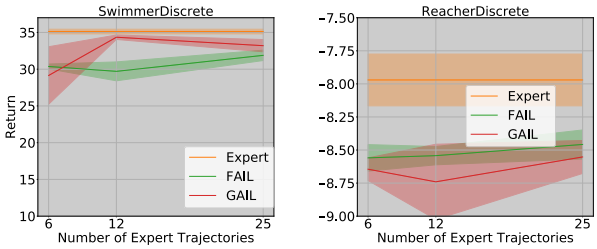
7. Simulation

We test FAIL on three simulations from openAI Gym (Brockman et al., 2016): Swimmer, Reacher, and the Fetch Robot Reach task (FetchReach). For Swimmer we set H to be 100 while for Reacher and FetchReach, H is 50 in default. The Swimmer task has dense reward (i.e.,., reward at every time step). For reacher, we try both dense reward and sparse reward (i.e., success if it reaches to the goal within a threshold). FetchReach is a sparse reward task. As our algorithm is presented for discrete action space, for all three tasks, we discrete the action space via discretizing each dimension into 5 numbers and applying categorical distribution independently for each dimension.[5] [6]

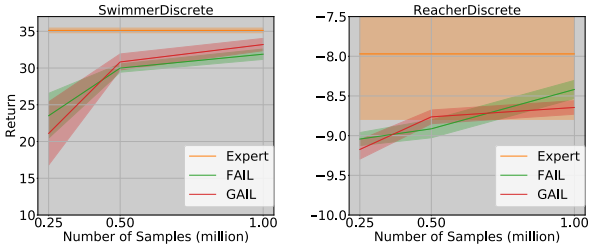
For baseline, we modify GAIL (Ho & Ermon, 2016), a model-free IL algorithm, based on the implementation from OpenAI Baselines, to make it work for ILFO. We delete the input of actions to discriminators in GAIL to make it work for ILFO. Hence the modified version can be understood as using RL (the implementation from OpenAI uses TRPO (Schulman et al., 2015)) methods to minimize the divergence between the learner's average state distribution

[5] i.e., $\pi(a|x) = \prod_{i=1}^d \pi_i(a[i]|x)$, with $a[i]$ stands for the i-th dimension. Note that common implementation for continuous control often assumes such factorization across action dimensions as the covariance matrix of the Gaussian distribution is often diagonal. See comments in Appendix J for extending FAIL to continuous control setting in practice.

[6] Implementation and scripts for reproducing results can be found at https://github.com/wensun/Imitation-Learning-from-Observation

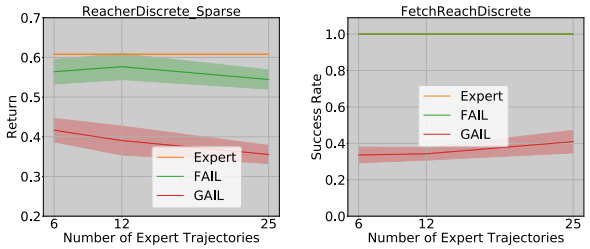


(a) Swimmer (1m) (b) Reacher (1m)



(c) swimmer (25) (d) Reacher (25)

Figure 1. Performance of expert, FAIL, and GAIL (without actions) on dense reward tasks (Reacher and Hopper). For (a) and (b), we fix the number of training samples while varying the number of expert demonstrations (6, 12, 25). For (c) and (d), we fix the number of expert trajectories, while varying the training samples.



(a) Reacher Sparse (1m) (b) FetchReach (1m)

Figure 2. Performance of expert, FAIL, and GAIL (without actions) on two sparse control tasks (Reacher Sparse and FetchReach). We fix the number of training samples while varying the number of expert demonstrations (6, 12, 25).

and the expert's average state distribution.

For FAIL implementation, we use MMD as a special IPM, where we use RBF kernel and set the width using the common median trick (e.g.,(Fukumizu et al., 2009)) without any future tuning. All policies are parameterized by one-layer neural networks. Instead of using FTRL, we use ADAM as an incremental no-regret learner, with all default parameters (e.g., learning rate) (Kingma & Ba, 2014). The total number of iteration T in Algorithm 1 is set to 1000 without any future tuning. During experiments, we stick to default hyperparameters for the purpose of best reflecting the algorithmic contribution of FAIL. All the results below are averaged over ten random trials with seeds randomly generated between $[1, 1e6]$. The experts are trained via a reinforcement learning algorithm (TRPO (Schulman et al., 2015)) with multiple millions of samples till convergence.

Figure 1 shows the comparison of expert, FAIL, and GAIL on two dense reward tasks with different number of expert demonstrations, under fixed total number of training samples (one million). We report mean and standard error in Figure 1. We observe GAIL outperforms FAIL in Swimmer on some configurations, while FAIL outperforms GAIL on Reacher (Dense reward) for all configuration.

Figure 2 shows the comparison of expert, FAIL, and GAIL on two sparse reward settings. We observe that FAIL significantly outperforms GAIL on these two sparse reward tasks. For sparse reward, note that what really matters is to reach the target at the end, FAIL achieves that by matching expert's state distribution and learner's state distribution one by one at every time step till the end, while GAIL (without actions) loses the sense of ordering by focusing on the average state distributions.

The above experiments also indicates that FAIL in general can work well for shorter horizon (e.g., $H = 50$ for Reacher and Fetch), while shows much less improvement over GAIL on longer horizon task. We think this is because of the nature of FAIL which has to learn a sequence of time-dependent policies along the entire horizon H. Long horizon requires larger number of samples. While method like GAIL learns a single stationary policy with all training data, and hence is less sensitive to horizon increase. We leave extending FAIL to learning a single stationary policy as a future work.

8. Conclusion and Future Work

We study Imitation Learning from Observation (ILFO) setting and propose an algorithm, Forward Adversarial Imitation Learning (FAIL), that achieves sample efficiency and computational efficiency. FAIL decomposes the sequential learning tasks into independent two-player min-max games of which is solved via general no-regret online learning. In addition to the algorithmic contribution, we present the first exponential gap in terms of sample complexity between ILFO and RL, demonstrating the potential benefit from expert's observations. A key observation is that one should explicitly regularize the class of discriminators to achieve sample efficiency and design discriminators to decrease the inherent Bellman Error. Experimentally, while GAIL can be used to solve the ILFO problem by removing action inputs to the discriminators, FAIL works just as well in problems with dense reward. Our analysis of FAIL provides the first strong theoretical guarantee for solving ILFO, and FAIL significantly outperforms GAIL on sparse reward MDPs, which are common in practice.

Acknowledgement

WS is supported in part by Office of Naval Research contract N000141512365. WS thanks Nan Jiang and Akshay Krishnamurthy for valuable discussions. We thank the first anonymous reviewer for carefully reviewing the proofs.

References

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pp. 1638–1646, 2014.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Arora, S., Ge, R., Liang, Y., Ma, T., and Zhang, Y. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.

Bagnell, J. A., Kakade, S. M., Schneider, J. G., and Ng, A. Y. Policy search by dynamic programming. In *Advances in neural information processing systems*, pp. 831–838, 2004.

Beygelzimer, A., Dani, V., Hayes, T., Langford, J., and Zadrozny, B. Error limiting reductions between classification tasks. In *Proceedings of the 22nd international conference on Machine learning*, pp. 49–56. ACM, 2005.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Chang, K.-W., He, H., Daumé III, H., and Langford, J. Learning to search for dependencies. *arXiv preprint arXiv:1503.05615*, 2015a.

Chang, K.-w., Krishnamurthy, A., Agarwal, A., Daume, H., and Langford, J. Learning to search better than your teacher. In *ICML*, 2015b.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. *arXiv preprint arXiv:1905.00360*, 2019.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2818–2826, 2015.

Daumé III, H., Langford, J., and Marcu, D. Search-based structured prediction. *Machine learning*, 2009.

Edwards, A. D., Sahni, H., Schroeker, Y., and Isbell, C. L. Imitating latent policies from observation. *arXiv preprint arXiv:1805.07914*, 2018.

Fukumizu, K., Gretton, A., Lanckriet, G. R., Schölkopf, B., and Sriperumbudur, B. K. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *Advances in neural information processing systems*, pp. 1750–1758, 2009.

Givan, R., Dean, T., and Greig, M. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Dulac-Arnold, G., et al. Deep q-learning from demonstrations. *arXiv preprint arXiv:1704.03732*, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *NIPS*, 2016.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low bellman rank are pac-learnable. *arXiv preprint arXiv:1610.09512*, 2016.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *ICML*, 2002.

Kakade, S., Kearns, M. J., and Langford, J. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 306–312, 2003.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krishnamurthy, A., Agarwal, A., and Langford, J. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pp. 1840–1848, 2016.

Lazaric, A., Ghavamzadeh, M., and Munos, R. Analysis of classification-based policy iteration algorithms. *The Journal of Machine Learning Research*, 17(1):583–612, 2016.

Liu, Y., Gupta, A., Abbeel, P., and Levine, S. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1118–1125. IEEE, 2018.

Luxburg, U. v. and Bousquet, O. Distance-based classification with lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29 (2):429–443, 1997.

Müller, K., Smola, A., and Rätsch, G. Predicting time series with support vector machines. *Artificial Neural Networks ICANN'9*, 1327:999–1004, 1997. doi: 10.1007/BFb0020283. URL http://link.springer.com/chapter/10.1007/BFb0020283.

Munos, R. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pp. 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9 (May):815–857, 2008.

Nair, A., Chen, D., Agrawal, P., Isola, P., Abbeel, P., Malik, J., and Levine, S. Combining self-supervised learning and imitation for vision-based rope manipulation. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 2146–2153. IEEE, 2017.

Pan, Y., Cheng, C.-A., Saigol, K., Lee, K., Yan, X., Theodorou, E., and Boots, B. Agile autonomous driving using end-to-end deep imitation learning. *Proceedings of Robotics: Science and Systems. Pittsburgh, Pennsylvania*, 2018.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *arXiv preprint arXiv:1804.02717*, 2018.

Ross, S. and Bagnell, J. A. Efficient reductions for imitation learning. In *AISTATS*, pp. 661–668, 2010.

Ross, S. and Bagnell, J. A. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

Ross, S., Gordon, G. J., and Bagnell, J. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. Trust region policy optimization. In *ICML*, pp. 1889–1897, 2015.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 2012.

Silver, D. et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G. R., et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Sun, W., Venkatraman, A., Boots, B., and Bagnell, J. A. Learning to filter with predictive state inference machines. In *ICML*, 2016.

Sun, W., Venkatraman, A., Gordon, G. J., Boots, B., and Bagnell, J. A. Deeply aggrevated: Differentiable imitation learning for sequential prediction. *arXiv preprint arXiv:1703.01030*, 2017.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A., and Langford, J. Model-based reinforcement learning in contextual decision processes. *arXiv preprint arXiv:1811.08540*, 2018.

Sutton, R. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

Syed, U. and Schapire, R. E. A reduction from apprenticeship learning to classification. In *Advances in neural information processing systems*, pp. 2253–2261, 2010.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

Venkatraman, A., Hebert, M., and Bagnell, J. A. Improving multi-step prediction of learned time series models. *AAAI*, 2015.

Zinkevich, M. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *ICML*, 2003.