Accelerating Imitation Learning with Predictive Models

Ching-An Cheng Georgia Tech
Xinyan Yan Georgia Tech
Evangelos A. Theodorou Georgia Tech
Byron Boots Georgia Tech

Abstract

Sample efficiency is critical in solving real-world reinforcement learning problems where agent-environment interactions can be costly. Imitation learning from expert advice has proved to be an effective strategy for reducing the number of interactions required to train a policy. Online imitation learning, which interleaves policy evaluation and policy optimization, is a particularly effective technique with provable performance guarantees. In this work, we seek to further accelerate the convergence rate of online imitation learning, thereby making it more sample efficient. We propose two model-based algorithms inspired by Follow-the-Leader (FTL) with prediction: MoBIL-VI based on solving variational inequalities and MoBIL-Prox based on stochastic first-order updates. These two methods leverage a model to predict future gradients to speed up policy learning. When the model oracle is learned online, these algorithms can provably accelerate the best known convergence rate up to an order. Our algorithms can be viewed as a generalization of stochastic Mirror-Prox (Juditsky et al., 2011), and admit a simple constructive FTL-style analysis of performance.

1 INTRODUCTION

Imitation learning (IL) has recently received attention for its ability to speed up policy learning when solving reinforcement learning problems (RL) [1, 2, 3, 4, 5, 6]. Unlike pure RL techniques, which rely on uniformed random exploration to locally improve a policy, IL leverages prior knowledge about a problem in terms of *expert demonstrations*. At a high level, this additional information provides policy learning with an informed search direction toward the expert policy.

The goal of IL is to quickly learn a policy that can perform at least as well as the expert policy. Because the expert policy may be suboptimal with respect to the RL problem of interest, performing IL is often used to provide a good warm start to the RL problem, so that the number of interactions with the environment can be minimized [7]. Sample efficiency is especially critical when learning is deployed in applications like robotics, where every interaction incurs real-world costs.

By reducing IL to an online learning problem, *online IL* [2] provides a framework for convergence analysis and mitigates the covariate shift problem encountered in batch IL [8, 9]. In particular, under proper assumptions, the performance of a policy sequence updated by Follow-the-Leader (FTL) can converge on average to the performance of the expert policy [2]. Recently, it was shown that this rate is sufficient to make IL more efficient than solving an RL problem from scratch [7].

In this work, we further accelerate the convergence rate of online IL. Inspired by the observation of Cheng and Boots [10] that the online learning problem of IL is *not* truly adversarial, we propose two MOdel-Based IL (MoBIL) algorithms, MoBIL-VI and MoBIL-Prox, that can achieve a fast rate of convergence. Under the same assumptions of Ross et al. [2], these algorithms improve on-average convergence from $O(\log N/N)$ to $O(1/N^2)$, e.g., when a dynamics model is learned online, where N is the number of iterations of policy update.

The improved speed of our algorithms is attributed to using a model oracle to predict the gradient of the next per-round cost in online learning. This model can be realized, e.g., using a simulator based on a (learned) dynamics model, or using past demonstrations. We first conceptually show that this idea can be realized as a variational inequality problem in MoBIL-VI. Next, we propose a practical first-order stochastic algorithm MoBIL-Prox, which alternates between the steps of taking the true gradient and of taking the model gradient. MoBIL-Prox is a generalization of stochastic Mirror-Prox proposed by Juditsky et al. [11] to the

Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

case where the problem is weighted and the vector field is unknown but learned online. In theory, we show that having a *weighting* scheme is pivotal to speeding up the order of convergence rate, and this generalization is made possible by a new constructive FTL-style regret analysis, which greatly simplifies the original algebraic proof [11]. The performance of MoBIL-Prox is also empirically validated in simulation.

2 PRELIMINARIES

2.1 Problem Setup: RL and IL

Let $\mathbb{S}$ and $\mathbb{A}$ be the state and the action spaces, respectively. The objective of RL is to search for a stationary policy π inside a policy class Π with good performance. This can be characterized by the stochastic optimization problem with expected cost[1] $J(\pi)$ defined below:

$$\min_{\pi\in\Pi} J(\pi), \quad J(\pi) \coloneqq \mathbb{E}_{(s,t)\sim d_\pi}\mathbb{E}_{a\sim\pi_s}\left[c_t(s,a)\right], \quad (1)$$

in which $s \in \mathbb{S}$, $a \in \mathbb{A}$, c_t is the instantaneous cost at time t, d_π is a *generalized stationary distribution* induced by executing policy π, and π_s is the distribution of action a given state s of π. The policies here are assumed to be parametric. To make the writing compact, we will abuse the notation π to also denote its parameter, and assume Π is a compact convex subset of parameters in some normed space with norm $\|\cdot\|$.

Based on the abstracted distribution d_π, the formulation in (1) subsumes multiple discrete-time RL problems. For example, a γ-discounted infinite-horizon problem can be considered by setting $c_t = c$ as a time-invariant cost and defining the joint distribution $d_\pi(s,t) = (1-\gamma)\gamma^t d_{\pi,t}(s)$, in which $d_{\pi,t}(s)$ denotes the probability (density) of state s at time t under policy π. Similarly, a T-horizon RL problem can be considered by setting $d_\pi(s,t) = \frac{1}{T} d_{\pi,t}(s)$. Note that while we use the notation $\mathbb{E}_{a\sim\pi_s}$, the policy is allowed to be deterministic; in this case, the notation means evaluation. For notational compactness, we will often omit the random variable inside the expectation (e.g. we shorten (1) to $\mathbb{E}_{d_\pi}\mathbb{E}_\pi[c]$). In addition, we denote $Q_{\pi,t}$ as the Q-function[2] at time t with respect to π.

In this paper, we consider IL, which is an indirect approach to solving the RL problem. We assume there is a black-box oracle π^*, called the *expert* policy, from which demonstration $a^* \sim \pi^*_s$ can be queried for any state $s \in \mathbb{S}$. To satisfy the querying requirement, usually the expert policy is an algorithm; for example, it can represent a planning algorithm which solves a simplified version of (1), or some engineered, hard-coded policy (see e.g. [12]).

The purpose of incorporating the expert policy into solving (1) is to quickly obtain a policy π that has reasonable performance. Toward this end, we consider solving a surrogate problem of (1),

$$\min_{\pi\in\Pi} \mathbb{E}_{(s,t)\sim d_\pi}[D(\pi^*_s||\pi_s)], \quad (2)$$

where D is a function that measures the difference between two distributions over actions (e.g. KL divergence; see Appendix B). Importantly, the objective in (2) has the property that $D(\pi^*||\pi^*) = 0$ and there is constant $C_{\pi^*} \geq 0$ such that $\forall t \in \mathbb{N}, s \in \mathbb{S}, \pi \in \Pi$, it satisfies $\mathbb{E}_{a\sim\pi_s}[Q_{\pi^*,t}(s,a)] - \mathbb{E}_{a^*\sim\pi^*_s}[Q_{\pi^*,t}(s,a^*)] \leq C_{\pi^*} D(\pi^*_s||\pi_s)$, in which $\mathbb{N}$ denotes the set of natural numbers. By the Performance Difference Lemma [13], it can be shown that the inequality above implies [10],

$$J(\pi) - J(\pi^*) \leq C_{\pi^*}\mathbb{E}_{d_\pi}[D(\pi^*||\pi)]. \quad (3)$$

Therefore, solving (2) can lead to a policy that performs similarly to the expert policy π^*.

2.2 Imitation Learning as Online Learning

The surrogate problem in (2) is more structured than the original RL problem in (1). In particular, when the distance-like function D is given, and we know that $D(\pi^*||\pi)$ is close to zero when π is close to π^*. On the contrary, $\mathbb{E}_{a\sim\pi_s}[c_t(s,a)]$ in (1) generally can still be large, even if π is a good policy (since it also depends on the state). This *normalization* property is crucial for the reduction from IL to online learning [10].

The reduction is based on observing that, with the normalization property, the expressiveness of the policy class Π can be described with a constant ϵ_Π defined as,

$$\epsilon_\Pi \geq \max_{\{\pi_n\in\Pi\}} \min_{\pi\in\Pi} \frac{1}{N}\sum_{n=1}^N \mathbb{E}_{d_{\pi_n}}[D(\pi^*||\pi)], \quad (4)$$

for all $N \in \mathbb{N}$, which measures the average difference between Π and π^* with respect to D and the state distributions visited by a worst possible policy sequence. Ross et al. [2] make use of this property and reduce (2) into an online learning problem by distinguishing the influence of π on d_π and on $D(\pi^*||\pi)$ in (2). To make this transparent, we define a bivariate function

$$F(\pi',\pi) \coloneqq \mathbb{E}_{d_{\pi'}}[D(\pi^*||\pi)]. \quad (5)$$

Using this bivariate function F, the online learning setup can be described as follows: in round n, the learner applies a policy $\pi_n \in \Pi$ and then the environment reveals a per-round cost

$$f_n(\pi) \coloneqq F(\pi_n,\pi) = \mathbb{E}_{d_{\pi_n}}[D(\pi^*||\pi)]. \quad (6)$$

[1] Our definition of $J(\pi)$ corresponds to the average accumulated cost in the RL literature.

[2] For example, in a T-horizon problem, $Q_{\pi,t}(s,a) = c_t(s,a) + \mathbb{E}_{\rho_{\pi,t}}\left[\sum_{\tau=t}^{T-1} c_\tau(s_\tau,a_\tau)\right]$, where $\rho_{\pi,t}$ denotes the distribution of future trajectory $(s_t,a_t,s_{t+1},\ldots s_{T-1},a_{T-1})$ conditioned on $s_t = s, a_t = a$.

Ross et al. [2] show that if the sequence $\{\pi_n\}$ is selected by a *no-regret algorithm*, then it will have good performance in terms of (2). For example, DAGGER updates the policy by FTL, $\pi_{n+1} = \arg\min_{\pi\in\Pi} f_{1:n}(\pi)$ and has the following guarantee (cf. [10]), where we define the shorthand $f_{1:n} = \sum_{m=1}^{n} f_m$.

Theorem 2.1. *Let $\mu_f > 0$. If each f_n is μ_f-strongly convex and $\|\nabla f_n(\pi)\|_* \leq G, \forall \pi \in \Pi$, then* DAGGER *has performance on average satisfying*

$$\frac{1}{N}\sum_{n=1}^{N} J(\pi_n) \leq J(\pi^*) + C_{\pi^*}\left(\frac{G^2}{2\mu_f}\frac{\ln N+1}{N} + \epsilon_\Pi\right). \quad (7)$$

First-order variants of DAGGER based on Follow-the-Regularized-Leader (FTRL) have also been proposed by Sun et al. [5] and Cheng et al. [7], which have the same performance but only require taking a stochastic gradient step in each iteration without keeping all the previous cost functions (i.e. data) as in the original FTL formulation. The bound in Theorem 2.1 also applies to the expected performance of a policy randomly picked out of the sequence $\{\pi_n\}_{n=1}^N$, although it does not necessarily translate into the performance of the last policy π_{N+1} [10].

3 ACCELERATING IL WITH PREDICTIVE MODELS

The reduction-based approach to solving IL has demonstrated sucess in speeding up policy learning. However, because interactions with the environment are necessary to approximately evaluate the per-round cost, it is interesting to determine if the convergence rate of IL can be further improved. A faster convergence rate will be valuable in real-world applications where data collection is expensive.

We answer this question affirmatively. We show that, by modeling[3] $\nabla_2 F$ the convergence rate of IL can potentially be improved by up to an order, where ∇_2 denotes the derivative to the second argument. The improvement comes through leveraging the fact that the per-round cost f_n defined in (6) is not completely unknown or adversarial as it is assumed in the most general online learning setting. Because the *same* function F is used in (6) over different rounds, the online component actually comes from the reduction made by Ross et al. [2], which ignores information about how F changes with the left argument; in other words, it omits the variations of d_π when π changes [10]. Therefore, we argue that the original reduction proposed by Ross et al. [2], while allowing the use of (4) to characterize the performance, loses one critical piece of information present in the original RL problem: *both the system dynamics and the expert are the same across different rounds of online learning.*

[3]We define $\nabla_2 F$ as a vector field $\nabla_2 F : \pi \mapsto \nabla_2 F(\pi, \pi)$

We propose two model-based algorithms (MOBIL-VI and MOBIL-PROX) to accelerate IL. The first algorithm, MOBIL-VI, is conceptual in nature and updates policies by solving variational inequality (VI) problems [14]. This algorithm is used to illustrate how modeling $\nabla_2 F$ through a *predictive model* $\nabla_2 \hat{F}$ can help to speed up IL, where $\hat{F}$ is a model bivariate function.[4] The second algorithm, MOBIL-PROX is a first-order method. It alternates between taking stochastic gradients by interacting with the environment and querying the model $\nabla_2 \hat{F}$. We will prove that this simple and practical approach has the same performance as the conceptual one: when $\nabla_2 \hat{F}$ is learned online and $\nabla_2 F$ is realizable, e.g. both algorithms can converge in $O\left(\frac{1}{N^2}\right)$, in contrast to DAGGER's $O\left(\frac{\ln N}{N}\right)$ convergence. In addition, we show the convergence results of MOBIL under relaxed assumptions, e.g. allowing stochasticity, and provide several examples of constructing predictive models. (See Appendix A for a summary of notation.)

3.1 Performance and Average Regret

Before presenting the two algorithms, we first summarize the core idea of the reduction from IL to online learning in a simple lemma, which builds the foundation of our algorithms (proved in Appendix C.1).

Lemma 3.1. *For arbitrary sequences $\{\pi_n \in \Pi\}_{n=1}^N$ and $\{w_n > 0\}_{n=1}^N$, it holds that*

$$\mathbb{E}\left[\sum_{n=1}^{N} \frac{w_n J(\pi_n)}{w_{1:N}}\right] \leq J(\pi^*) + C_{\pi^*}\left(\epsilon_\Pi^w + \mathbb{E}\left[\frac{\text{regret}^w(\Pi)}{w_{1:N}}\right]\right)$$

where $\tilde{f}_n$ is an unbiased estimate of f_n, $\text{regret}^w(\Pi) \coloneqq \max_{\pi\in\Pi}\sum_{n=1}^{N} w_n\tilde{f}_n(\pi_n) - w_n\tilde{f}_n(\pi)$, *$\epsilon_\Pi^w$ is given in Definition 4.1, and the expectation is due to sampling $\tilde{f}_n$.*

In other words, the on-average performance convergence of an online IL algorithm is determined by the rate of the expected weighted average regret $\mathbb{E}\left[\text{regret}^w(\Pi)/w_{1:N}\right]$. For example, in DAGGER, the weighting is uniform and $\mathbb{E}\left[\text{regret}^w(\Pi)\right]$ is in $O(\log N)$; by Lemma 3.1 this rate directly proves Theorem 2.1.

3.2 Algorithms

From Lemma 3.1, we know that improving the regret bound implies a faster convergence of IL. This leads to the main idea of MOBIL-VI and MOBIL-PROX: to use model information to *approximately* play Be-the-Leader (BTL) [15], i.e. $\pi_{n+1} \approx \arg\min_{\pi\in\Pi} f_{1:n+1}(\pi)$. To understand why playing BTL can minimize the regret, we recall a classical regret bound of online learning.[5]

[4]While we only concern predicting the vector field $\nabla_2 F$, we adopt the notation $\hat{F}$ to better build up the intuition, especially of MOBIL-VI; we will discuss other approximations that are not based on bivariate functions in Section 3.3.

[5]We use notation x_n and l_n to distinguish general online learning problems from online IL problems.

Lemma 3.2 (Strong FTL Lemma [16]). *For any sequence of decisions $\{x_n \in \mathcal{X}\}$ and loss functions $\{l_n\}$, $\text{regret}(\mathcal{X}) \leq \sum_{n=1}^{N} l_{1:n}(x_n) - l_{1:n}(x_n^\star)$, where $x_n^\star \in \arg\min_{x\in\mathcal{X}} l_{1:n}(x)$, where $\mathcal{X}$ is the decision set.*

Namely, if the decision π_{n+1} made in round n in IL is close to the best decision in round $n+1$ after the new per-round cost f_{n+1} is revealed (which depends on π_{n+1}), then the regret will be small.

The two algorithms are summarized in Algorithm 1, which mainly differ in the policy update rule (line 5). Like DAgger, they both learn the policy in an interactive manner. In round n, both algorithms execute the current policy π_n in the real environment to collect data to define the per-round cost functions (line 3): $\tilde{f}_n$ is an unbiased estimate of f_n in (6) for policy learning, and $\tilde{h}_n$ is an unbiased estimate of the per-round cost h_n for model learning. Given the current per-round costs, the two algorithms then update the model (line 4) and the policy (line 5) using the respective rules. Here we use the set $\hat{\mathcal{F}}$, abstractly, to denote the family of predictive models to estimate $\nabla_2 F$, and h_n is defined as an upper bound of the prediction error. For example, $\hat{\mathcal{F}}$ can be a family of dynamics models that are used to simulate the predicted gradients, and $\tilde{h}_n$ is the empirical loss function used to train the dynamics models (e.g. the KL divergence of prediction).

3.2.1 A Conceptual Algorithm: MoBIL-VI

We first present our conceptual algorithm MoBIL-VI, which is simpler to explain. We assume that f_n and h_n are given, as in Theorem 2.1. This assumption will be removed in MoBIL-Prox later. To realize the idea of BTL, in round n, MoBIL-VI uses a newly learned predictive model $\nabla_2 \hat{F}_{n+1}$ to estimate $\nabla_2 F$ in (5) and then updates the policy by solving the VI problem below: finding $\pi_{n+1} \in \Pi$ such that $\forall \pi' \in \Pi$,

$$\langle \Phi_n(\pi_{n+1}), \pi' - \pi_{n+1} \rangle \geq 0, \tag{8}$$

where the vector field Φ_n is defined as

$$\Phi_n(\pi) = w_{n+1}\nabla_2 \hat{F}_{n+1}(\pi,\pi) + \textstyle\sum_{m=1}^{n} w_m \nabla f_m(\pi)$$

Suppose $\nabla_2 \hat{F}_{n+1}$ is the partial derivative of some bivariate function $\hat{F}_{n+1}$. If $w_n = 1$, then the VI problem[6] in (8) finds a fixed point π_{n+1} satisfying $\pi_{n+1} = \arg\min_{\pi\in\Pi} f_{1:n}(\pi) + \hat{F}_{n+1}(\pi_{n+1}, \pi)$. That is, if $\hat{F}_{n+1} = F$ exactly, then π_{n+1} plays exactly BTL and by Lemma 3.2 the regret is non-positive. In general, we can show that, even with modeling errors, MoBIL-VI can still reach a faster convergence rate such as $O\left(\frac{1}{N^2}\right)$, if a non-uniform weighting scheme is used, the model is updated online, and $\nabla_2 F$ is realizable within $\hat{\mathcal{F}}$. The details will be presented in Section 4.2.

[6] Because Π is compact, the VI problem in (8) has at least one solution [14]. If f_n is strongly convex, the VI problem in line 6 of Algorithm 1 is strongly monotone for large enough n and can be solved e.g. by basic projection method [14]. Therefore, for demonstration purpose, we assume the VI problem of MoBIL-VI can be exactly solved.

Algorithm 1 MoBIL

Input: π_1, N, p
Output: $\bar{\pi}_N$
1: Set weights $w_n = n^p$ for $n = 1, \ldots, N$ and sample integer K with $P(K = n) \propto w_n$
2: **for** $n = 1 \ldots K-1$ **do**
3: Run π_n in the real environment to collect data to define $\tilde{f}_n$ and $\tilde{h}_n$[7]
4: Update the predictive model to $\nabla_2 \hat{F}_{n+1}$; e.g., using FTL $\hat{F}_{n+1} = \arg\min_{\hat{F}\in\hat{\mathcal{F}}} \sum_{m=1}^{n} \frac{w_m}{m} \tilde{h}_m(\hat{F})$
5: Update policy to π_{n+1} by (8) (MoBIL-VI) or by (9) (MoBIL-Prox)
6: **end for**
7: Set $\bar{\pi}_N = \pi_K$

3.2.2 A Practical Algorithm: MoBIL-Prox

While the previous conceptual algorithm achieves a faster convergence, it requires solving a nontrivial VI problem in each iteration. In addition, it assumes f_n is given as a function and requires keeping all the past data to define $f_{1:n}$. Here we relax these unrealistic assumptions and propose MoBIL-Prox. In round n of MoBIL-Prox, the policy is updated from π_n to π_{n+1} by *taking two gradient steps*:

$$\begin{aligned} \hat{\pi}_{n+1} &= \arg\min_{\pi\in\Pi} \textstyle\sum_{m=1}^{n} w_m\big(\langle g_m, \pi\rangle + r_m(\pi)\big), \\ \pi_{n+1} &= \arg\min_{\pi\in\Pi} w_{n+1}\langle \hat{g}_{n+1}, \pi\rangle + \\ &\qquad \textstyle\sum_{m=1}^{n} w_m\big(\langle g_m, \pi\rangle + r_m(\pi)\big) \end{aligned} \tag{9}$$

We define r_n as an $\alpha_n\mu_f$-strongly convex function (with $\alpha_n \in (0, 1]$; we recall μ_f is the strongly convexity modulus of f_n) such that π_n is its global minimum and $r_n(\pi_n) = 0$ (e.g. a Bregman divergence). And we define g_n and $\hat{g}_{n+1}$ as estimates of $\nabla f_n(\pi_n) = \nabla_2 F(\pi_n, \pi_n)$ and $\nabla_2 \hat{F}_{n+1}(\hat{\pi}_{n+1}, \hat{\pi}_{n+1})$, respectively. Here we only require $g_n = \nabla \tilde{f}_n(\pi_n)$ to be unbiased, whereas $\hat{g}_n$ could be a biased estimate of $\nabla_2 \hat{F}_{n+1}(\hat{\pi}_{n+1}, \hat{\pi}_{n+1})$.

MoBIL-Prox treats $\hat{\pi}_{n+1}$, which plays FTL with g_n from the real environment, as a rough estimate of the next policy π_{n+1} and uses it to query an gradient estimate $\hat{g}_{n+1}$ from the model $\nabla_2 \hat{F}_{n+1}$. Therefore, the learner's decision π_{n+1} can approximately play BTL. If we compare the update rule of π_{n+1} and the VI problem in (8), we can see that MoBIL-Prox linearizes the problem and attempts to approximate $\nabla_2 \hat{F}_{n+1}(\pi_{n+1}, \pi_{n+1})$ by $\hat{g}_{n+1}$. While the above approximation is crude, interestingly it is sufficient to speed up the convergence rate to be as fast as MoBIL-VI under mild assumptions, as shown later in Section 4.3.

[7] MoBIL-VI assumes $\tilde{f}_n = f_n$ and $\tilde{h}_n = h_n$

3.3 Predictive Models

MoBIL uses $\nabla_2 \hat{F}_{n+1}$ in the update rules (8) and (9) at round n to predict the unseen gradient at round $n+1$ for speeding up policy learning. Ideally $\hat{F}_{n+1}$ should approximate the unknown bivariate function F so that $\nabla_2 F$ and $\nabla_2 \hat{F}_{n+1}$ are close. This condition can be seen from (8) and (9), in which MoBIL concerns only $\nabla_2 \hat{F}_{n+1}$ instead of $\hat{F}_{n+1}$ directly. In other words, $\nabla_2 \hat{F}_{n+1}$ is used in MoBIL as a first-order oracle, which leverages all the past information (up to the learner playing π_n in the environment at round n) to predict the future gradient $\nabla_2 F_{n+1}(\pi_{n+1}, \pi_{n+1})$, which depends on the decision π_{n+1} the learner is about to make. Hence, we call it a predictive model.

To make the idea concrete, we provide a few examples of these models. By definition of F in (5), one way to construct the predictive model $\nabla_2 \hat{F}_{n+1}$ is through a *simulator* with an (online learned) dynamics model, and define $\nabla_2 \hat{F}_{n+1}$ as the simulated gradient (computed by querying the expert along the simulated trajectories visited by the learner). If the dynamics model is exact, then $\nabla_2 \hat{F}_{n+1} = \nabla_2 F$. Note that a stochastic/biased estimate of $\nabla_2 \hat{F}_{n+1}$ suffices to update the policies in MoBIL-Prox.

Another idea is to construct the predictive model through $\tilde{f}_n$ (the stochastic estimate of f_n) and indirectly define $\hat{F}_{n+1}$ such that $\nabla_2 \hat{F}_{n+1} = \nabla \tilde{f}_n$. This choice is possible, because the learner in IL collects *samples* from the environment, as opposed to, literally, gradients. Specifically, we can define $g_n = \nabla \tilde{f}_n(\pi_n)$ and $\hat{g}_{n+1} = \nabla \tilde{f}_n(\hat{\pi}_{n+1})$ in (9). The approximation error of setting $\hat{g}_{n+1} = \nabla \tilde{f}_n(\hat{\pi}_{n+1})$ is determined by the convergence and the stability of the learner's policy. If π_n visits similar states as $\hat{\pi}_{n+1}$, then $\nabla \tilde{f}_n$ can approximate $\nabla_2 F$ well at $\hat{\pi}_{n+1}$. Note that this choice is different from using the previous gradient (i.e. $\hat{g}_{n+1} = g_n$) in optimistic mirror descent/FTL [17], which would have a larger approximation error due to additional linearization.

Finally, we note that while the concept of predictive models originates from estimating the partial derivatives $\nabla_2 F$, a predictive model does not necessarily have to be in the same form. A parameterized vector-valued function can also be directly learned to approximate $\nabla_2 F$, e.g., using a neural network and the sampled gradients $\{g_n\}$ in a supervised learning fashion.

4 THEORETICAL ANALYSIS

Now we prove that using predictive models in MoBIL can accelerate convergence, when proper conditions are met. Intuitively, MoBIL converges faster than the usual adversarial approach to IL (like DAgger), when the predictive models have smaller errors than not predicting anything at all (i.e. setting $\hat{g}_{n+1} = 0$). In the following analyses, we will focus on bounding the expected weighted average regret, as it directly translates into the average performance bound by Lemma 3.1. We define, for $w_n = n^p$,

$$\mathcal{R}(p) \coloneqq \mathbb{E}\left[\text{regret}^w(\Pi)/w_{1:N}\right] \tag{10}$$

Note that the results below assume that the predictive models are updated using FTL as outlined in Algorithm 1. This assumption applies, e.g., when a dynamics model is learned online in a simulator-oracle as discussed above. We provide full proofs in Appendix C and provide a summary of notation in Appendix A.

4.1 Assumptions

We first introduce several assumptions to more precisely characterize the online IL problem.

Predictive models Let $\hat{\mathcal{F}}$ be the class of predictive models. We assume these models are Lipschitz continuous in the following sense.

Assumption 4.1. *There is $L \in [0, \infty)$ such that $\|\nabla_2 \hat{F}(\pi, \pi) - \nabla_2 \hat{F}(\pi', \pi')\|_* \le L\|\pi - \pi'\|$, $\forall \hat{F} \in \hat{\mathcal{F}}$ and $\forall \pi, \pi' \in \Pi$.*

Per-round costs The per-round cost f_n for policy learning is given in (6), and we define $h_n(\hat{F})$ as an upper bound of $\|\nabla_2 F(\pi_n, \pi_n) - \nabla_2 \hat{F}(\pi_n, \pi_n)\|_*^2$ (see e.g. Appendix D). We make structural assumptions on $\tilde{f}_n$ and $\tilde{h}_n$, similar to the ones made by Ross et al. [2] (cf. Theorem 2.1).

Assumption 4.2. *Let $\mu_f, \mu_h > 0$. With probability 1, $\tilde{f}_n$ is μ_f-strongly convex, and $\|\nabla \tilde{f}_n(\pi)\|_* \le G_f$, $\forall \pi \in \Pi$; $\tilde{h}_n$ is μ_h-strongly convex, and $\|\nabla \tilde{h}_n(\hat{F})\|_* \le G_h$, $\forall \hat{F} \in \hat{\mathcal{F}}$.*

By definition, these properties extend to f_n and h_n. We note they can be relaxed to solely *convexity* and our algorithms still improve the best known convergence rate (see Table 1 and Appendix E).

Expressiveness of hypothesis classes We introduce two constants, ϵ_Π^w and $\epsilon_{\hat{\mathcal{F}}}^w$, to characterize the policy class Π and model class $\hat{\mathcal{F}}$, which generalize the idea of (4) to stochastic and general weighting settings. When $\tilde{f}_n = f_n$ and θ_n is constant, Definition 4.1 agrees with (4). Similarly, we see that if $\pi^* \in \Pi$ and $F \in \hat{\mathcal{F}}$, then ϵ_Π^w and $\epsilon_{\hat{\mathcal{F}}}^w$ are zero.

[8] The rates here assume $\sigma_{\hat{g}}, \sigma_g, \epsilon_{\hat{\mathcal{F}}}^w = 0$. In general, the rate of MoBIL-Prox becomes the improved rate in the table plus the ordinary rate multiplied by $C = \sigma_g^2 + \sigma_{\hat{g}}^2 + \epsilon_{\hat{\mathcal{F}}}^w$. For example, when $\tilde{f}$ is convex and $\tilde{h}$ is strongly convex, MoBIL-Prox converges in $O(1/N + C/\sqrt{N})$, whereas DAgger converges in $O(G_f^2/\sqrt{N})$.

Table 1: Convergence Rate Comparison[8]

	$\tilde{h}_n$ convex	$\tilde{h}_n$ strongly convex	Without model
$\tilde{f}_n$ convex	$O(N^{-3/4})$	$O(N^{-1})$	$O(N^{-1/2})$
$\tilde{f}_n$ strongly convex	$O(N^{-3/2})$	$O(N^{-2})$	$O(N^{-1})$

Definition 4.1. A policy class Π is ϵ_Π^w-*close to* π^*, if for all $N \in \mathbb{N}$ and weight sequence $\{\theta_n > 0\}_{n=1}^N$ with $\theta_{1:N} = 1$, $\mathbb{E}\big[\max_{\{\pi_n \in \Pi\}} \min_{\pi \in \Pi} \sum_{n=1}^N \theta_n \tilde{f}_n(\pi)\big] \le \epsilon_\Pi^w$. Similarly, a model class $\hat{\mathcal{F}}$ is $\epsilon_{\hat{\mathcal{F}}}^w$-*close to* F, if $\mathbb{E}\big[\max_{\{\pi_n \in \Pi\}} \min_{\hat{F} \in \hat{\mathcal{F}}} \sum_{n=1}^N \theta_n \tilde{h}_n(\hat{F})\big] \le \epsilon_{\hat{\mathcal{F}}}^w$. The expectations above are due to sampling $\tilde{f}_n$ and $\tilde{h}_n$.

4.2 Performance of MoBIL-VI

Here we show the performance for MoBIL-VI when there is prediction error in $\nabla_2 \hat{F}_n$. The main idea is to treat MoBIL-VI as online learning with prediction [17] and take $\hat{F}_{n+1}(\pi_{n+1}, \cdot)$ obtained after solving the VI problem (8) as an *estimate* of f_{n+1}.

Proposition 4.1. *For* MoBIL-VI *with* $p = 0$, $\mathcal{R}(0) \le \frac{G_f^2}{2\mu_f \mu_h} \frac{1}{N} + \frac{\epsilon_{\hat{\mathcal{F}}}^w}{2\mu_f} \frac{\ln N + 1}{N}$.

By Lemma 3.1, this means that if the model class is expressive enough (i.e $\epsilon_{\hat{\mathcal{F}}}^w = 0$), then by adapting the model online with FTL, we can improve the original convergence rate in $O(\ln N/N)$ of Ross et al. [2] to $O(1/N)$. While removing the $\ln N$ factor does not seem like much, we will show that running MoBIL-VI can improve the convergence rate to $O(1/N^2)$, when a *non-uniform* weighting is adopted.

Theorem 4.1. *For* MoBIL-VI *with* $p > 1$, $R(p) \le C_p \left(\frac{pG_h^2}{2(p-1)\mu_h} \frac{1}{N^2} + \frac{\epsilon_{\hat{\mathcal{F}}}^w}{pN} \right)$, *where* $C_p = \frac{(p+1)^2 e^{p/N}}{2\mu_f}$.

The key is that $\text{regret}^w(\Pi)$ can be upper bounded by the regret of the online learning for models, which has per-round cost $\frac{w_n}{n} h_n$. Therefore, if $\epsilon_{\hat{\mathcal{F}}}^w \approx 0$, randomly picking a policy out of $\{\pi_n\}_{n=1}^N$ proportional to weights $\{w_n\}_{n=1}^N$ has expected convergence in $O\left(\frac{1}{N^2}\right)$ if $p > 1$.[9]

4.3 Performance of MoBIL-Prox

As MoBIL-Prox uses gradient estimates, we additionally define two constants σ_g and $\sigma_{\hat{g}}$ to characterize the estimation error, where $\sigma_{\hat{g}}$ also entails potential bias.

Assumption 4.3. $\mathbb{E}[\|g_n - \nabla_2 F(\pi_n, \pi_n)\|_*^2] \le \sigma_g^2$ *and* $\mathbb{E}[\|\hat{g}_n - \nabla_2 \hat{F}_n(\hat{\pi}_n, \hat{\pi}_n)\|_*^2] \le \sigma_{\hat{g}}^2$

We show this simple first-order algorithm achieves similar performance to MoBIL-VI. Toward this end, we introduce a stronger lemma than Lemma 3.2.

[9] If $p = 1$, it converges in $O\left(\frac{\ln N}{N^2}\right)$; if $p \in [0, 1)$, it converges in $O\left(\frac{1}{N^{1+p}}\right)$. See Appendix C.2.

Lemma 4.1 (Stronger FTL Lemma)**.** *Let* $x_n^\star \in \arg\min_{x \in \mathcal{X}} l_{1:n}(x)$. *For any sequence of decisions* $\{x_n\}$ *and losses* $\{l_n\}$, $\text{regret}(\mathcal{X}) = \sum_{n=1}^N l_{1:n}(x_n) - l_{1:n}(x_n^\star) - \Delta_n$, *where* $\Delta_{n+1} := l_{1:n}(x_{n+1}) - l_{1:n}(x_n^\star) \ge 0$.

The additional $-\Delta_n$ term in Lemma 4.1 is pivotal to prove the performance of MoBIL-Prox.

Theorem 4.2. *For* MoBIL-Prox *with* $p > 1$ *and* $\alpha_n = \alpha \in (0, 1]$, *it satisfies*

$$\mathcal{R}(p) \le \frac{(p+1)^2 e^{\frac{p}{N}}}{\alpha \mu_f} \left(\frac{G_h^2}{\mu_h} \frac{p}{p-1} \frac{1}{N^2} + \frac{2}{p} \frac{\sigma_g^2 + \sigma_{\hat{g}}^2 + \epsilon_{\hat{\mathcal{F}}}^w}{N} \right) + \frac{(p+1)\nu_p}{N^{p+1}},$$

where $\nu_p = O(1)$ *and* $n_{\text{ceil}} = \lceil \frac{2e^{\frac{1}{2}}(p+1)LG_f}{\alpha \mu_f} \rceil$.

Proof sketch. Here we give a proof sketch in big-O notation (see Appendix C.3 for the details). To bound $\mathcal{R}(p)$, recall the definition $\text{regret}^w(\Pi) = \sum_{n=1}^N w_n \tilde{f}_n(\pi_n) - \min_{\pi \in \Pi} \sum_{n=1}^N w_n \tilde{f}_n(\pi)$. Now define $\bar{f}_n(\pi) := \langle g_n, \pi \rangle + r_n(\pi)$. Since $\tilde{f}_n$ is μ_f-strongly convex, r_n is $\alpha\mu_f$-strongly convex, and $r(\pi_n) = 0$, we know that $\bar{f}_n$ satisfies that $\tilde{f}_n(\pi_n) - \tilde{f}_n(\pi) \le \bar{f}_n(\pi_n) - \bar{f}_n(\pi)$, $\forall \pi \in \Pi$. This implies $\mathcal{R}(p) \le \mathbb{E}[\text{regret}_{\text{path}}^w(\Pi)/w_{1:N}]$, where $\text{regret}_{\text{path}}^w(\Pi) := \sum_{n=1}^N w_n \bar{f}_n(\pi_n) - \min_{\pi \in \Pi} \sum_{n=1}^N w_n \bar{f}_n(\pi)$.

The following lemma upper bounds $\text{regret}_{\text{path}}^w(\Pi)$ by using Stronger FTL lemma (Lemma 4.1).

Lemma 4.2. $\text{regret}_{\text{path}}^w(\Pi) \le \frac{p+1}{2\alpha\mu_f} \sum_{n=1}^N n^{p-1} \|g_n - \hat{g}_n\|_*^2 - \frac{\alpha\mu_f}{2(p+1)} \sum_{n=1}^N (n-1)^{p+1} \|\pi_n - \hat{\pi}_n\|^2$.

Since the second term in Lemma 4.2 is negative, we just need to upper bound the expectation of the first item. Using the triangle inequality, we bound the model's prediction error of the next per-round cost.

Lemma 4.3. $\mathbb{E}[\|g_n - \hat{g}_n\|_*^2] \le 4(\sigma_g^2 + \sigma_{\hat{g}}^2 + L^2 \mathbb{E}[\|\pi_n - \hat{\pi}_n\|^2] + \mathbb{E}[\tilde{h}_n(\hat{F}_n)])$.

With Lemma 4.3 and Lemma 4.2, it is now clear that $\mathbb{E}[\text{regret}_{\text{path}}^w(\Pi)] \le \mathbb{E}[\sum_{n=1}^N \rho_n \|\pi_n - \hat{\pi}_n\|^2] + O(N^p)(\sigma_g^2 + \sigma_{\hat{g}}^2) + O(\mathbb{E}[\sum_{n=1}^N n^{p-1} \tilde{h}_n(\hat{F}_n)])$, where $\rho_n = O(n^{p-1} - n^{p+1})$. When n is large enough, $\rho_n \le 0$, and hence the first term is $O(1)$. For the third term, because the model is learned online using, e.g., FTL with strongly convex cost $n^{p-1}\tilde{h}_n$ we can show that $\mathbb{E}[\sum_{n=1}^N n^{p-1} \tilde{h}_n(\hat{F}_n)] = O(N^{p-1} + N^p \epsilon_{\hat{\mathcal{F}}}^w)$. Thus, $\mathbb{E}[\text{regret}_{\text{path}}^w(\Pi)] \le O(1 + N^{p-1} +$

$(\epsilon^w_{\hat{\mathcal{F}}} + \sigma_g^2 + \sigma_{\hat{g}}^2)N^p)$. Substituting this bound into $\mathcal{R}(p) \leq \mathbb{E}[\text{regret}^w_{\text{path}}(\Pi)/w_{1:N}]$ and using that the fact $w_{1:N} = \Omega(N^{p+1})$ proves the theorem. ■

The main assumption in Theorem 4.2 is that $\nabla_2 \hat{F}$ is L-Lipschitz continuous (Assumption 4.1). It does not depend on the continuity of $\nabla_2 F$. Therefore, this condition is practical as we are free to choose $\hat{\mathcal{F}}$. Compared with Theorem 4.1, Theorem 4.2 considers the inexactness of $\tilde{f}_n$ and $\tilde{h}_n$ explicitly; hence the additional term due to σ_g^2 and $\sigma_{\hat{g}}^2$. Under the same assumption of MoBIL-VI that f_n and h_n are directly available, we can actually show that the simple MoBIL-Prox has the same performance as MoBIL-VI, which is a corollary of Theorem 4.2.

Corollary 4.1. *If $\tilde{f}_n = f_n$ and $\tilde{h}_n = h_n$, for* MoBIL-Prox *with $p > 1$, $\mathcal{R}(p) \leq O(\frac{1}{N^2} + \frac{\epsilon^w_{\hat{\mathcal{F}}}}{N})$.*

The proof of Theorem 4.1 and 4.2 are based on assuming the predictive models are updated by FTL (see Appendix D for a specific bound when online learned dynamics models are used as a simulator). However, we note that these results are essentially based on the property that model learning also has no regret; therefore, the FTL update rule (line 4) can be replaced by a no-regret first-order method without changing the result. This would make the algorithm even simpler to implement. The convergence of other types of predictive models (like using the previous cost function discussed in Section 3.3) can also be analyzed following the major steps in the proof of Theorem 4.2, leading to a performance bound in terms of prediction errors. Finally, it is interesting to note that the accelerated convergence is made possible when model learning puts more weight on costs in later rounds (because $p > 1$).

4.4 Comparison

We compare the performance of MoBIL in Theorem 4.2 with that of DAgger in Theorem 2.1 in terms of the constant on the $\frac{1}{N}$ factor. MoBIL has a constant in $O(\sigma_g^2 + \sigma_{\hat{g}}^2 + \epsilon^w_{\hat{\mathcal{F}}})$, whereas DAgger has a constant in $G_f^2 = O(G^2 + \sigma_g^2)$, where we recall G_f and G are upper bounds of $\|\nabla \tilde{f}_n(\pi)\|_*$ and $\|\nabla f_n(\pi)\|_*$, respectively.[10] Therefore, in general, MoBIL-Prox has a better upper bound than DAgger when the model class is expressive (i.e. $\epsilon_{\hat{\mathcal{F}}} \approx 0$), because $\sigma_{\hat{g}}^2$ (the variance of the sampled gradients) can be made small as we are free to design the model. Note that, however, the improvement of MoBIL may be smaller when the problem is noisy, such that the large σ_g^2 becomes the dominant term.

An interesting property that arises from Theorems 4.1 and 4.2 is that the convergence of MoBIL is not biased by using an imperfect model (i.e. $\epsilon^w_{\hat{\mathcal{F}}} > 0$). This is shown in the term $\epsilon^w_{\hat{\mathcal{F}}}/N$. In other words, in the worst case of using an extremely wrong predictive model, MoBIL would just converge more slowly but still to the performance of the expert policy.

MoBIL-Prox is closely related to stochastic Mirror-Prox [18, 11]. In particular, when the exact model is known (i.e. $\nabla_2 \hat{F}_n = \nabla_2 F$) and MoBIL-Prox is set to convex-mode (i.e. $r_n = 0$ for $n > 1$, and $w_n = 1/\sqrt{n}$; see Appendix E), then MoBIL-Prox gives the same update rule as stochastic Mirror-Prox with step size $O(1/\sqrt{n})$ (See Appendix F for a thorough discussion). Therefore, MoBIL-Prox can be viewed as a generalization of Mirror-Prox: 1) it allows non-uniform weights; and 2) it allows the vector field $\nabla_2 F$ to be estimated online by alternately taking stochastic gradients and predicted gradients. The design of MoBIL-Prox is made possible by our Stronger FTL lemma (Lemma 4.1), which greatly simplifies the original algebraic proof in [18, 11]. Using Lemma 4.1 reveals more closely the interactions between model updates and policy updates. In addition, it more clearly shows the effect of non-uniform weighting, which is essential to achieving $O(\frac{1}{N^2})$ convergence. To the best of our knowledge, even the analysis of the original (stochastic) Mirror-Prox from the FTL perspective is new.

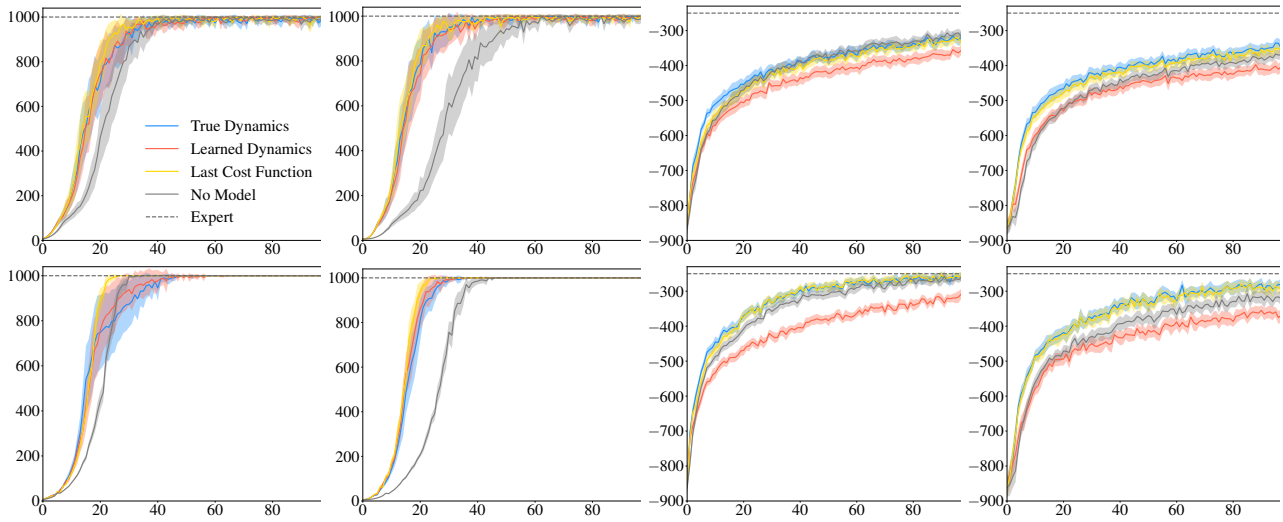
5 EXPERIMENTS

We experimented with MoBIL-Prox in simulation to study how weights $w_n = n^p$ and the choice of model oracles affect the learning. We used two weight schedules: $p = 0$ as baseline, and $p = 2$ suggested by Theorem 4.2. And we considered several predictive models: (a) a simulator with the true dynamics (b) a simulator with online-learned dynamics (c) the last cost function (i.e. $\hat{g}_{n+1} = \nabla \tilde{f}_n(\hat{\pi}_{n+1})$ (d) no model (i.e. $\hat{g}_{n+1} = 0$; in this case MoBIL-Prox reduces to the first-order version of DAgger [7], which is considered as a baseline here).

5.1 Setup and Results

Two robot control tasks (CartPole and Reacher3D) powered by the DART physics engine [19] were used as the task environments. The learner was either a linear policy or a small neural network. For each IL problem, an expert policy that shares the same architecture as the learner was used, which was trained using policy gradients. While sharing the same architecture is not required in IL, here we adopted this constraint to remove the bias due to the mismatch between policy class and the expert policy to clarify the experimental results. For MoBIL-Prox, we set $r_n(\pi) = \frac{\mu_f \alpha_n}{2}\|\pi - \pi_n\|^2$ and set α_n such that $\sum w_n \alpha_n \mu_f = (1 + cn^{p+1/2})/\eta_n$, where $c = 0.1$ and η_n was adaptive to the norm of the prediction error. This leads to an effective learning rate

[10] Theorem 2.1 was stated by assuming $f_n = \tilde{f}_n$. In the stochastic setup here, DAgger has a similar convergence rate in expectation but with G replaced by G_f.



(a) CartPole $p = 0$ (b) CartPole $p = 2$ (c) Reacher3D $p = 0$ (d) Reacher3D $p = 2$

Figure 1: Experimental results of MoBIL-Prox with neural network (1st row) and linear policies (2nd row). The shaded regions represent 0.5 standard deviation over 24 seeds.

$\eta_n w^p/(1 + cn^{p+1/2})$ which is optimal in the convex setting (cf. Table 1). For the dynamics model, we used a neural network and trained it using FTL. The results reported are averaged over 24 seeds. Figure 1 shows the results of MoBIL-Prox. While the use of neural network policies violates the convexity assumptions in the analysis, it is interesting to see how MoBIL-Prox performs in this more practical setting. We include the experiment details in Appendix G for completeness.

5.2 Discussions

We observe that, when $p = 0$, having model information does not improve the performance much over standard online IL (i.e. no model), as suggested in Proposition 4.1. By contrast, when $p = 2$ (as suggested by Theorem 4.2), MoBIL-Prox improves the convergence and performs better than not using models.[11] It is interesting to see that this trend also applies to neural network policies.

From Figure 1, we can also study how the choice of predictive models affects the convergence. As suggested in Theorem 4.2, MoBIL-Prox improves the convergence only when the model makes non-trivial predictions. If the model is very incorrect, then MoBIL-Prox can be slower. This can be seen from the performance of MoBIL-Prox with online learned dynamics models. In the low-dimensional case of CartPole, the simple neural network predicts the dynamics well, and MoBIL-Prox with the learned dynamics performs similarly as MoBIL-Prox with the true dynamics. However, in the high-dimensional Reacher3D problem, the learned dynamics model generalizes less well, creating a performance gap between MoBIL-Prox using the true dynamics and that using the learned dynamics. We note that MoBIL-Prox would still converge at the end despite the model error. Finally, we find that the performance of MoBIL with the last-cost predictive model is often similar to MoBIL-Prox with the simulated gradients computed through the true dynamics.

[11] We note that the curves between $p = 0$ and $p = 2$ are not directly comparable; we should only compare methods within the same p setting as the optimal step size varies with p. The multiplier on the step size was chosen such that MoBIL-Prox performs similarly in both settings.

6 CONCLUSION

We propose two novel model-based IL algorithms MoBIL-Prox and MoBIL-VI with strong theoretical properties: they are provably up-to-and-order faster than the state-of-the-art IL algorithms and have unbiased performance even when using imperfect predictive models. Although we prove the performance under convexity assumptions, we empirically find that MoBIL-Prox improves the performance even when using neural networks. In general, MoBIL accelerates policy learning when having access to an predictive model that can predict future gradients non-trivially. While the focus of the current paper is theoretical in nature, the design of MoBIL leads to several interesting questions that are important to reliable application of MoBIL-Prox in practice, such as end-to-end learning of predictive models and designing adaptive regularizations for MoBIL-Prox.

Acknowledgements

This work was supported in part by NSF NRI Award 1637758 and NSF CAREER Award 1750483.

References

[1] Pieter Abbeel and Andrew Y Ng. Exploration and apprenticeship learning in reinforcement learning. In *International conference on Machine learning*, pages 1–8. ACM, 2005.

[2] 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*, pages 627–635, 2011.

[3] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[4] Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume III, and John Langford. Learning to search better than your teacher. 2015.

[5] Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. *arXiv preprint arXiv:1703.01030*, 2017.

[6] Hoang M Le, Nan Jiang, Alekh Agarwal, Miroslav Dudík, Yisong Yue, and Hal Daumé III. Hierarchical imitation and reinforcement learning. *arXiv preprint arXiv:1803.00590*, 2018.

[7] Ching-An Cheng, Xinyan Yan, Nolan Wagener, and Byron Boots. Fast policy learning using imitation and reinforcement. *Conference on Uncertainty in Artificial Intelligence*, 2018.

[8] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[9] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*, 2017.

[10] Ching-An Cheng and Byron Boots. Convergence of value aggregation for imitation learning. In *International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1801–1809, 2018.

[11] Anatoli Juditsky, Arkadi Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[12] Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos Theodorou, and Byron Boots. Agile off-road autonomous driving using end-to-end deep imitation learning. *arXiv preprint arXiv:1709.07174*, 2017.

[13] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

[14] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[15] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[16] H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

[17] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. 2013.

[18] Arkadi Nemirovski. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[19] Jeongseok Lee, Michael X Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S Srinivasa, Mike Stilman, and C Karen Liu. Dart: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, 3(22):500, 2018.

[20] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

[21] M Bianchi and S Schaible. Generalized monotone bifunctions and equilibrium problems. *Journal of Optimization Theory and Applications*, 90(1): 31–43, 1996.

[22] IV Konnov and S Schaible. Duality for equilibrium problems under generalized monotonicity. *Journal of Optimization Theory and Applications*, 104(2): 395–408, 2000.

[23] Sándor Komlósi. On the Stampacchia and Minty variational inequalities. *Generalized Convexity and Optimization for Economic and Financial Decisions*, pages 231–260, 1999.

[24] Nam Ho-Nguyen and Fatma Kilinc-Karzan. Exploiting problem structure in optimization under uncertainty via online convex optimization. *arXiv preprint arXiv:1709.02490*, 2017.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.