Proximal Point Imitation Learning

Luca Viano
LIONS, EPFL
Lausanne, Switzerland
luca.viano@epfl.ch

Angeliki Kamoutsi
ETH Zurich
Zurich, Switzerland
kamoutsa@ethz.ch

Gergely Neu
Universitat Pompeu Fabra
Barcelona, Spain
gergely.neu@gmail.com

Igor Krawczuk
LIONS, EPFL
Lausanne, Switzerland
igor.krawczuk@epfl.ch

Volkan Cevher
LIONS, EPFL
Lausanne, Switzerland
volkan.cevher@epfl.ch

Abstract

This work develops new algorithms with rigorous efficiency guarantees for infinite horizon imitation learning (IL) with linear function approximation without restrictive coherence assumptions. We begin with the minimax formulation of the problem and then outline how to leverage classical tools from optimization, in particular, the proximal-point method (PPM) and dual smoothing, for online and offline IL, respectively. Thanks to PPM, we avoid nested policy evaluation and cost updates for online IL appearing in the prior literature. In particular, we do away with the conventional alternating updates by the optimization of a single convex and smooth objective over both cost and Q-functions. When solved inexactly, we relate the optimization errors to the suboptimality of the recovered policy. As an added bonus, by re-interpreting PPM as dual smoothing with the expert policy as a center point, we also obtain an offline IL algorithm enjoying theoretical guarantees in terms of required expert trajectories. Finally, we achieve convincing empirical performance for both linear and neural network function approximation.

1 Introduction

This work is concerned with the prototypical setting of imitation learning (IL) where

1. An expert provides demonstrations of state-action pairs in an environment. The expert could be optimal or suboptimal with respect to an unknown cost/reward function.
2. The learner chooses distance measure between its policy to be learned and the expert empirical distribution estimated from demonstrations.
3. The learner employs an algorithm, which additionally may or may not use interactions with the environment, to minimize the chosen distance.

In IL, the central goal of the learner is to recover a policy competitive with expert with respect to the underlying unknown cost function. IL is important for several real world applications like driving [62], robotics [88], and economics/finance [27] at the expense of following resources: (R1) expert demonstrations, (R2) (optional) interactions with the environment where the expert collected the demonstrations, and (R3) computational resources for solving the problem template.

Interestingly, while there is a vast amount of literature using optimization ideas on the IL problem template, i.e. Lagrangian duality [51, 38, 59, 63, 64], resource guarantees are still widely missing since the optimization literature focuses on the resource (R3) where IL literature mainly focuses on

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

the first two resources (R1) and (R2). Our work leverages deeper connections between optimization tools and IL by showing how classical optimization tools can be applied in a linear programming formulation of IL problem guaranteeing efficiency in all (R1), (R2), (R3).

Our contributions: This work aims at designing an algorithm enjoying both theoretical guarantees and convincing empirical performance. Our methodology is rooted in classical optimization tools and the LP approach to MDPs. More precisely, the method uses the recently repopularized overparameterization technique to obtain the Q-function as a Lagrangian multiplier [77, 14] and solves the associated program using a PPM update with appropriately chosen Bregman divergences. This results to an actor-critic algorithm, with the key feature that the policy evaluation step involves optimization of a single concave and smooth objective over both cost and Q-functions. In this way, we avoid instability or poor convergence due to adversarial training [51, 122, 70, 105], and can also recover an explicit cost along with Q-function. We further account for potential optimization errors, presenting an error propagation analysis that leads to rigorous guarantees for both online and offline setting. For the context of linear MDPs [14, 121, 55, 22, 116, 7, 84], we provide explicit convergence rates and error bounds for the suboptimality of the learned policy, under mild assumptions, significantly weaker than those found in the literature until now. To our knowledge, such guarantees in this setting are provided for the first time. Finally, we demonstrate that our approach achieves convincing empirical performance for both linear and neural network function approximation.

Related Literature. The first algorithm addressing the imitation learning problem is behavioral cloning [93]. Due to the covariate shift problem [98, 99], it has low efficiency in terms of expert trajectories (R1). To address this issue, [100, 87, 4, 95, 111, 85, 123, 5, 68, 69] proposed to cast the problem as inverse reinforcement learning (IRL). IRL improves the efficiency in terms of expert trajectories, at the cost of introducing the need of running reinforcement learning (RL) repetitively, which can be prohibitive in terms of environment samples (R2) and computation (R3). A successive line of work started with [112] highlights that repeated calls to an RL routine can be avoided. This work inspired generative adversarial imitation learning (GAIL) [51] and other follow-up works [38, 59, 63, 64] that leveraged optimization tools like primal-dual algorithms but did not try to deepen the optimization connections to derive efficiency guarantees in terms of all (R1),(R2),(R3). Finally, a recent line of work [40, 57] in IL bypasses the need of optimizing over cost functions and thus avoids instability due to adversarial training. Although these algorithms achieve impressive empirical performance in challenging high dimensional benchmark tasks, they are hampered by limited theoretical understanding. This is the fundamental difference from our work, which enjoys both favorable practical performance and strong theoretical guarantees.

Existing model-free IL theoretical papers with global convergence guarantees assume either a finite horizon episodic MDP setting [70], or tabular MDPs [105], or the infinite horizon case but with restrictive assumptions, such as linear quadratic regulator setting [21], continuous kernelized nonlinear regulator [26, 56], access to a generative model and coherence assumption on the choice of features [58, 14], bounded strong concentrability coefficients [122] or a linear transition law that can be completely specified by a finite-dimensional matrix [70]. On the other hand, we provide convergence guarantees and error bounds for the context of linear MDPs [14, 121, 55, 22, 116, 7, 84] under a mild *feature excitation* condition assumption. Despite being linear, the transition law can still have infinite degrees of freedom. To our knowledge, such guarantees in this setting are provided for the first time.

Our work applies the technique known as regularization in the online learning literature [6, 103] and Bregman proximal-point or smoothing in optimization literature [97, 82] to the LP formulation for MDPs [73, 35, 36, 17, 48, 49, 33, 34, 102, 91, 92, 1, 65, 30, 79, 115, 67, 13, 31, 55, 106]. From this perspective, we can see Deep Inverse Q-Learning [57] and IQ-Learn [40] that consider entropy regularization in the objective as smoothing using uniform distribution as center point. In our case, we instead use as center point the previous iteration of the algorithm (for the online case) or the expert (for the offline case).

From the technical point of view, the most important related works are the analysis of REPS/Q-REPS [90, 14, 89] and O-REPS [124] that first pointed out the connection between REPS and PPM. We build on their techniques with some important differences. In particular, while in the LP formulation of RL, PPM and mirror descent [15, 47] are equivalent, recognizing that they are *not equivalent* in IL is critical for stronger empirical performance. As an independent interest, our techniques can be used to improve upon the best rate for REPS in the tabular setting [89] and to extend the guarantees to linear MDPs. In order to discuss in more detail our research questions and situate them among prior related theoretical and practical works, we provide in Appendix A an extended literature review.

2 Background

2.1 Markov Decision Processes

The RL environment and its underlying dynamics are typically abstracted as an MDP given by a tuple $(\mathcal{S}, \mathcal{A}, P, \boldsymbol{\nu}_0, \mathbf{c}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \rightarrow \Delta_{\mathcal{S}}$ is the transition law, $\boldsymbol{\nu}_0 \in \Delta_{\mathcal{S}}$ is the initial state distribution, $\mathbf{c} \in [0,1]^{|\mathcal{S}||\mathcal{A}|}$ is the cost, and $\gamma \in (0,1)$ is the discount factor. For simplicity, we focus on problems where $\mathcal{S}$ and $\mathcal{A}$ are finite but too large to be enumerated. A *stationary Markov policy* $\pi \colon \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$ interacts with the environment iteratively, starting with an initial state $s_0 \sim \boldsymbol{\nu}_0$. At round t, if the system is at state s_t, an action $a_t \sim \pi(\cdot|s_t)$ is sampled and applied to the environment. Then a cost $c(s,a)$ is incurred, and the system transitions to the next state $s_{t+1} \sim P(\cdot|s,a)$. The goal of RL is to solve the optimal control problem $\rho^\star_{\mathbf{c}} \triangleq \min_\pi \rho_{\mathbf{c}}(\pi)$, where $\rho_{\mathbf{c}}(\pi) \triangleq (1-\gamma) \langle \boldsymbol{\nu}_0, \mathbf{V}^\pi_{\mathbf{c}} \rangle$ is the *normalized total discounted expected cost* of π.

The *state value function* $\mathbf{V}^\pi_{\mathbf{c}} \in \mathbb{R}^{|\mathcal{S}|}$ of π, given cost $\mathbf{c}$, is defined by $V^\pi_{\mathbf{c}}(s) \triangleq \mathbb{E}^\pi_s \left[\sum_{t=0}^\infty \gamma^t c(s_t, a_t) \right]$, where $\mathbb{E}^\pi_s$ denotes the expectation with respect to the trajectories generated by π starting from $s_0 = s$. The *optimal value function* $\mathbf{V}^\star_{\mathbf{c}} \in \mathbb{R}^{|\mathcal{S}|}$ is defined by $V^\star_{\mathbf{c}}(s) \triangleq \min_\pi V^\pi_{\mathbf{c}}(s)$. The *optimal state-action value function* $\mathbf{Q}^\star_{\mathbf{c}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, given by $Q^\star_{\mathbf{c}}(s,a) \triangleq c(s,a) + \gamma \sum_{s'} V^\star_{\mathbf{c}}(s') P(s'|s,a)$, is known to characterize optimal behaviors. Indeed $\mathbf{V}^\star_{\mathbf{c}}$ is the unique solution to the *Bellman optimality equation* $V^\star_{\mathbf{c}}(s) = \min_a Q^\star_{\mathbf{c}}(s,a)$. In addition, any deterministic policy $\pi^\star_{\mathbf{c}}(s) = \arg\min_a Q^\star_{\mathbf{c}}(s,a)$ is known to be optimal.

For every policy π, we define the *normalized state-action occupancy measure* $\boldsymbol{\mu}_\pi \in \Delta_{\mathcal{S}\times\mathcal{A}}$, by $\mu_\pi(s,a) \triangleq (1-\gamma) \sum_{t=0}^\infty \gamma^t \, \mathbb{P}^\pi_{\boldsymbol{\nu}_0} \left[s_t = s, a_t = a \right]$, where $\mathbb{P}^\pi_{\boldsymbol{\nu}_0}[\cdot]$ denotes the probability of an event when following π starting from $s_0 \sim \boldsymbol{\nu}_0$. The occupancy measure can be interpreted as the discounted visitation frequency of state-action pairs. This allows us to write $\rho_{\mathbf{c}}(\pi) = \langle \boldsymbol{\mu}_\pi, \mathbf{c} \rangle$.

2.2 Imitation Learning

Similarly to RL, the IL problem is posed in the MDP formalism, with the critical difference that the true cost $\mathbf{c}_{\text{true}}$ is unknown. Instead, we have access to a finite set of truncated trajectories sampled i.i.d. by executing an expert policy π_{E} in the environment. The goal is to learn a policy that performs better than π_{E} with respect to the unknown $\mathbf{c}_{\text{true}}$. To this end, we adopt the *apprenticeship learning* formalism [4, 112, 50, 51, 105], which carries the assumption that $\mathbf{c}_{\text{true}}$ belongs to a class of cost functions $\mathcal{C}$. We then seek an *apprentice policy* π_{A} that outperforms the expert across $\mathcal{C}$ by solving the following optimization problem

$$\zeta^\star \triangleq \min_\pi d_{\mathcal{C}}(\pi, \pi_{\text{E}}), \tag{1}$$

where $d_{\mathcal{C}}(\pi, \pi_{\text{E}}) \triangleq \max_{\mathbf{c} \in \mathcal{C}} \left(\rho_{\mathbf{c}}(\pi) - \rho_{\mathbf{c}}(\pi_{\text{E}}) \right)$ defines the $\mathcal{C}$-distance between π and π_{E} [51, 28, 122, 70]. Then, π_{A} satisfies the goal of IL, since it holds that $\rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{A}}) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{E}}) \leq \zeta^\star \leq 0$. Intuitively, the cost class $\mathcal{C}$ distinguishes the expert from other policies. The maximization in (1) assigns high total cost to non-expert policies and low total cost to π_{E} [51], while the minimization aims to find the policy that matches the expert as close as possible with respect to $d_{\mathcal{C}}$.

By writing $d_{\mathcal{C}}$ in its *dual* form $\bar{d}_{\mathcal{C}}(\boldsymbol{\mu}_\pi, \boldsymbol{\mu}_{\pi_{\text{E}}}) \triangleq \max_{\mathbf{c} \in \mathcal{C}} \left(\langle \boldsymbol{\mu}_\pi, \mathbf{c} \rangle - \langle \boldsymbol{\mu}_{\pi_{\text{E}}}, \mathbf{c} \rangle \right)$, it can be interpreted as an *integral probability metric* [80, 60] between the occupancy measures $\boldsymbol{\mu}_\pi$ and $\boldsymbol{\mu}_{\pi_{\text{E}}}$. Depending on how $\mathcal{C}$ is chosen, $d_{\mathcal{C}}$ turns to a different metric of probability measures like the 1-Wasserstein distance [117, 32] for $\mathcal{C} = \text{Lip}_1(\mathcal{S} \times \mathcal{A})$, the total variation for $\mathcal{C} = \{\mathbf{c} \mid \|\mathbf{c}\|_\infty \leq 1\}$, or the maximum mean discrepancy for $\mathcal{C} = \{\mathbf{c} \mid \|\mathbf{c}\|_{\mathcal{H}} \leq 1\}$, where $\text{Lip}_1(\mathcal{S} \times \mathcal{A})$ denotes the space of 1-Lipschitz functions on $\mathcal{S} \times \mathcal{A}$, and $\|\cdot\|_{\mathcal{H}}$ denotes the norm of a reproducing kernel Hilbert space $\mathcal{H}$ [104].

In our theoretical analysis, we focus on linearly parameterized cost classes [111, 112, 51, 70, 105] of the form $\mathcal{C} \triangleq \{\mathbf{c}_{\mathbf{w}} \triangleq \sum_{i=1}^m w_i \boldsymbol{\phi}_i \mid \mathbf{w} \in \mathcal{W}\}$, where $\{\boldsymbol{\phi}_i\}_{i=1}^m \subset \mathbb{R}_+^{|\mathcal{S}||\mathcal{A}|}$ are fixed feature vectors, such that $\|\boldsymbol{\phi}_i\|_1 \leq 1$ for all $i \in [m]$, and $\mathcal{W}$ is a a convex constraint set for the cost weights $\mathbf{w}$. This assumption is not necessarily restrictive as usually in practice the true cost depends on just a few key properties, but the desirable weighting that specifies how different desiderata should be traded-off is unknown [4]. Moreover, the cost features can be complex nonlinear functions that can be obtained via unsupervised learning from raw state observations [20, 29]. The matrix $\mathbf{\Phi} \triangleq [\boldsymbol{\phi}_1 \quad \dots \quad \boldsymbol{\phi}_m]$ gives rise a *feature expectation vector* (FEV) $\boldsymbol{\rho}_{\mathbf{\Phi}}(\pi) \triangleq (\rho_{\boldsymbol{\phi}_1}(\pi_{\mathrm{E}}), \dots, \rho_{\boldsymbol{\phi}_m}(\pi_{\mathrm{E}}))^\intercal \in \mathbb{R}^m$ of a policy π. Then, by choosing $\mathcal{W}$ to be the ℓ_2 unit ball $B_1^m \triangleq \{\mathbf{w} \in \mathbb{R}^m \mid \|\mathbf{w}\|_2 \leq 1\}$ [4], we get a *feature expectation matching* objective $d_{\mathcal{C}}(\pi, \pi_{\pi_{\mathrm{E}}}) = \|\boldsymbol{\rho}_{\mathbf{\Phi}}(\pi) - \boldsymbol{\rho}_{\mathbf{\Phi}}(\pi_{\mathrm{E}})\|_2$, while for $\mathcal{W}$ being the probability simplex $\Delta_{[m]}$ [111, 112] we have a worst-case excess cost objective $d_{\mathcal{C}}(\pi, \pi_{\pi_{\mathrm{E}}}) = \max_{i \in [m]} \big(\rho_{\boldsymbol{\phi}_i}(\pi) - \rho_{\boldsymbol{\phi}_i}(\pi_{\mathrm{E}})\big)$. For clarity, we will replace $\mathbf{c}$ by $\mathbf{w}$ in the notation of the quantities defined in Section 2.1.

3 A Q-Convex-Analytic Viewpoint

Our methodology builds upon the convex-analytic approach to AL, first introduced by [112], with the key difference that we consider a different convex formulation that introduces Q-functions as slack variables. This allows to design a practical scalable model-free algorithm with theoretical guarantees.

Let $\mathfrak{F} \triangleq \{\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|} \mid (\mathbf{B} - \gamma\mathbf{P})^\intercal \boldsymbol{\mu} = (1-\gamma)\boldsymbol{\nu}_0,\ \boldsymbol{\mu} \geq \mathbf{0}\}$ be the *state-action polytope*, where $\mathbf{P}$ is the vector form of P, i.e., $P_{(s,a),s'} \triangleq P(s'|s,a)$, and $\mathbf{B}$ is a binary matrix defined by $B_{(s,a),s'} \triangleq 1$ if $s = s'$, and $B_{(s,a),s'} \triangleq 0$ otherwise. The linear constraints that define the set $\mathfrak{F}$, also known as *Bellman flow constraints*, precisely characterize the set of state-action occupancy measures.

Proposition 1 (94)**.** *We have that $\boldsymbol{\mu} \in \mathfrak{F}$ if and only if there exists a unique stationary Markov policy π such that $\boldsymbol{\mu} = \boldsymbol{\mu}_\pi$. If $\boldsymbol{\mu} \in \mathcal{F}$ then the policy $\pi_{\boldsymbol{\mu}}(a|x) \triangleq \frac{\boldsymbol{\mu}(x,a)}{\sum_{a' \in \mathcal{A}} \boldsymbol{\mu}(x,a')}$ has occupancy measure $\boldsymbol{\mu}$.*

Using Proposition 1 and the dual form of the $\mathcal{C}$-distance $\bar{d}_{\mathcal{C}}(\boldsymbol{\mu}, \boldsymbol{\mu}_{\pi_{\mathrm{E}}}) = \max_{\mathbf{w} \in \mathcal{W}} \big\langle \boldsymbol{\mu} - \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_{\mathbf{w}} \big\rangle$, it follows that (1) is equivalent to the primal convex program $\zeta^\star = \min_{\boldsymbol{\mu}} \{\bar{d}_{\mathcal{C}}(\boldsymbol{\mu}, \boldsymbol{\mu}_{\pi_{\mathrm{E}}}) \mid \boldsymbol{\mu} \in \mathfrak{F}\}$. In particular for $\mathcal{W} = \Delta_{[m]}$ and by using an epigraphic transformation, we end up with an LP program [112], while for $\mathcal{W} = B_1^m$ we get a quadratic objective with linear constraints [4].

A slight variation of the above reasoning is to introduce a mirror variable $\mathbf{d}$ and split the Bellman flow constraints in the definition of $\mathfrak{F}$. We then get the primal convex program

$$\zeta^\star = \min_{(\boldsymbol{\mu}, \mathbf{d})} \{\bar{d}_{\mathcal{C}}(\boldsymbol{\mu}, \boldsymbol{\mu}_{\pi_{\mathrm{E}}}) \mid (\boldsymbol{\mu}, \mathbf{d}) \in \mathfrak{M}\}, \tag{Primal}$$

where the new polytope is given by $\mathfrak{M} \triangleq \{(\boldsymbol{\mu}, \mathbf{d}) \mid \mathbf{B}^\intercal \mathbf{d} = \gamma \mathbf{P}^\intercal \boldsymbol{\mu} + (1-\gamma)\boldsymbol{\nu}_0,\ \boldsymbol{\mu} = \mathbf{d},\ \mathbf{d} \geq \mathbf{0}\}$. This overparameterization trick has been first introduced by Mehta and Meyn [76] and has been recently revisited by [14, 84, 67, 83, 77, 71]. A salient feature of this equivalent formulation is that it introduces a Q-function as Lagrange multiplier to the equality constraint $\mathbf{d} = \boldsymbol{\mu}$, and so lends itself to data-driven algorithms. To motivate further this new formulation, in Appendix C, we shed light to its dual and provide an interpretation of the dual optimizers. In particular, when $\mathcal{W} = B_1^m$, we show that $(\mathbf{V}^\star_{\mathbf{w}_{\text{true}}}, \mathbf{Q}^\star_{\mathbf{w}_{\text{true}}}, \mathbf{w}_{\text{true}})$ is a dual optimizer.

For our theoretical analysis we focus on the linear MDP setting [55], i.e., we assume that the transition law is linear in the feature mapping. We denote by $\boldsymbol{\phi}(s,a)$ the (s,a)-th row of $\mathbf{\Phi}$.

Assumption 1 (Linear MDP)**.** *There exists a collection of m probability measures $\boldsymbol{\omega} = (\omega_1, \dots, \omega_m)$ on $\mathcal{S}$, such that $P(\cdot|s,a) = \langle \boldsymbol{\omega}(\cdot), \phi(s,a) \rangle$, for all (s,a). Moreover $\boldsymbol{\phi}(s,a) \in \Delta_{[m]}$, for all (s,a).*

Assumption 1 essentialy says that the transition matrix $\mathbf{P}$ has rank at most m, and $\mathbf{P} = \mathbf{\Phi}\mathbf{M}$ for some matrix $\mathbf{M} \in \mathbb{R}^{m \times |\mathcal{S}|}$. It is worth noting that in the case of continuous MDPs, despite being linear, the transition law $P(\cdot|s,a)$ can still have infinite degrees of freedom. This is a substantial difference from the recent theoretical works on IL [70, 105] which consider either a linear quadratic regulator, or a transition law that can be completely specified by a finite-dimensional matrix such that the degrees of freedom are bounded.

Assumption 1 enables us to consider a relaxation of (Primal). In particular, we aggregate the constraints $\boldsymbol{\mu} = \mathbf{d}$ by imposing $\mathbf{\Phi}^\intercal \boldsymbol{\mu} = \mathbf{\Phi}^\intercal \mathbf{d}$ instead, and introduce a variable $\boldsymbol{\lambda} = \mathbf{\Phi}^\intercal \boldsymbol{\mu}$. It follows that $\boldsymbol{\lambda}$ lies in the m-dimensional simplex $\Delta_{[m]}$. Then, we get the following convex program

$$\zeta^\star = \min_{(\boldsymbol{\lambda}, \mathbf{d})} \{\max_{\mathbf{w} \in \mathcal{W}} \langle \boldsymbol{\lambda}, \mathbf{w} \rangle - \big\langle \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_{\mathbf{w}} \big\rangle \mid (\boldsymbol{\lambda}, \mathbf{d}) \in \mathfrak{M}_{\mathbf{\Phi}}\}, \tag{Primal$'$}$$

where $\mathfrak{M}_{\mathbf{\Phi}} \triangleq \{(\boldsymbol{\lambda}, \mathbf{d}) \mid \mathbf{B}^\intercal \mathbf{d} = \gamma \mathbf{M}^\intercal \boldsymbol{\lambda} + (1-\gamma)\boldsymbol{\nu}_0, \ \boldsymbol{\lambda} = \mathbf{\Phi}^\intercal \mathbf{d}, \ \boldsymbol{\lambda} \in \Delta_{[m]}, \ \mathbf{d} \in \Delta_{\mathcal{S}\times\mathcal{A}}\}$. As shown in [84, 14, 83], for linear MDPs, the set of occupancy measures $\mathfrak{F}$ can be completely characterized by the set $\mathfrak{M}_{\mathbf{\Phi}}$ (c.f., Proposition 2). While the number of constraints and variables in (Primal′) is intractable for large scale MDPs, in the next paragraph, we show how this problem can be solved using a proximal point scheme.

4 Proximal Point Imitation Learning

By using a Lagrangian decomposition, we have that (Primal′) is equivalent to the following bilinear saddle-point problem

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \langle \mathbf{y}, \mathbf{A}\mathbf{x} + \mathbf{b} \rangle, \tag{SPP}$$

where $\mathbf{A} \in \mathbb{R}^{(2m+|\mathcal{S}|)\times(m+|\mathcal{S}||\mathcal{A}|)}$, and $\mathbf{b} \in \mathbb{R}^{(m+|\mathcal{S}|+|\mathcal{S}||\mathcal{A}|)}$ are appropriately defined (see Appendix D), $\mathbf{x} \triangleq [\boldsymbol{\lambda}^\intercal, \mathbf{d}^\intercal]^\intercal$, $\mathbf{y} \triangleq [\mathbf{w}^\intercal, \mathbf{V}^\intercal, \boldsymbol{\theta}^\intercal]^\intercal$, $\mathcal{X} \triangleq \Delta_{[m]} \times \Delta_{\mathcal{S}\times\mathcal{A}}$, and $\mathcal{Y} \triangleq \mathcal{W} \times \mathbb{R}^{|\mathcal{S}|} \times \mathbb{R}^m$.

Since in practice we do not have access to the whole policy π_{E}, but instead can observe a finite set of i.i.d. sample trajectories $\mathcal{D}_{\mathrm{E}} \triangleq \{(x_0^{(l)}, a_0^{(l)}, x_1^{(l)}, a_1^{(l)}, \ldots, x_H^{(l)}, a_H^{(l)})\}_{l=1}^{n_{\mathrm{E}}} \sim \pi_{\mathrm{E}}$, we define the vector $\widehat{\mathbf{b}}$ by replacing $\boldsymbol{\rho}_{\mathbf{\Phi}}(\pi_{\mathrm{E}})$ with its empirical counterpart $\boldsymbol{\rho}_{\mathbf{\Phi}}(\widehat{\pi_{\mathrm{E}}})$ (by taking sample averages) in the definition of $\mathbf{b}$. We then consider the empirical objective $f(\mathbf{x}) \triangleq \max_{\mathbf{y}\in\mathcal{Y}} \left\langle \mathbf{y}, \mathbf{A}\mathbf{x} + \widehat{\mathbf{b}} \right\rangle$ and apply PPM on the decision variable $\mathbf{x}$. For the $\boldsymbol{\lambda}$-variable we use the relative entropy $D(\boldsymbol{\lambda}||\boldsymbol{\lambda}') \triangleq \sum_{i=1}^m \lambda(i) \log \frac{\lambda(i)}{\lambda'(i)}$, while for the occupancy measure $\mathbf{d}$ we use the conditional relative entropy $H(\mathbf{d}||\mathbf{d}') \triangleq \sum_{s,a} d(s,a) \log \frac{\pi_{\mathbf{d}}(a|s)}{\pi_{\mathbf{d}'}(a|s)}$. With this choice we can rewrite the PPM update as

$$(\boldsymbol{\lambda}_{k+1}, \mathbf{d}_{k+1}) = \operatorname*{arg\,min}_{\boldsymbol{\lambda}\in\Delta_{[m]}, \mathbf{d}\in\Delta_{\mathcal{S}\times\mathcal{A}}} \max_{\mathbf{y}\in\mathcal{Y}} \left\langle \mathbf{y}, \mathbf{A}\begin{bmatrix}\boldsymbol{\lambda}\\ \mathbf{d}\end{bmatrix} + \widehat{\mathbf{b}} \right\rangle + \frac{1}{\eta} D(\boldsymbol{\lambda}||\mathbf{\Phi}^\intercal \mathbf{d}_k) + \frac{1}{\alpha} H(\mathbf{d}||\mathbf{d}_k), \tag{2}$$

where we used primal feasibility to replace $\boldsymbol{\lambda}_k$ with $\mathbf{\Phi}^\intercal \mathbf{d}_k$ as the center point of the relative entropy. PPM is implicit, meaning that it requires the evaluation of the gradient at the next iterate $\mathbf{x}_{k+1}$. Such a requirement makes it not implementable in general. However, in the following, we describe a procedure to apply proximal point to our specific $f(\mathbf{x})$. The following Proposition summarizes the result.

Proposition 2. *For a parameter $\boldsymbol{\theta} \in \mathbb{R}^m$, we define the logistic state-action value function $\mathbf{Q}_{\boldsymbol{\theta}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ by $\mathbf{Q}_{\boldsymbol{\theta}} \triangleq \mathbf{\Phi}\boldsymbol{\theta}$, and the k-step logistic state value function $\mathbf{V}_{\boldsymbol{\theta}}^k \in \mathbb{R}^{|\mathcal{S}|}$ by*

$$V_{\boldsymbol{\theta}}^k(s) \triangleq -\frac{1}{\alpha} \log\left(\sum_a \pi_{\mathbf{d}_{k-1}}(a|s) e^{-\alpha Q_{\boldsymbol{\theta}}(s,a)}\right).$$

Moreover, we define the k-step reduced Bellman error function $\boldsymbol{\delta}_{\mathbf{w},\boldsymbol{\theta}}^k \in \mathbb{R}^m$ by $\boldsymbol{\delta}_{\mathbf{w},\boldsymbol{\theta}}^k \triangleq \mathbf{w} + \gamma \mathbf{M}\mathbf{V}_{\boldsymbol{\theta}}^k - \boldsymbol{\theta}$. Then, the PPM update $(\boldsymbol{\lambda}_k^\star, \mathbf{d}_k^\star)$ in 2 is given by

$$\lambda_k^\star(i) \propto (\mathbf{\Phi}^\intercal \mathbf{d}_{k-1})(i)\, e^{-\eta \delta^k_{\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star}(i)}, \tag{3}$$

$$\pi_{\mathbf{d}_k^\star}(a|s) \propto \pi_{\mathbf{d}_{k-1}}(a|s)\, e^{-\alpha Q_{\boldsymbol{\theta}_k^\star}(s,a)}, \tag{4}$$

where $(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star)$ is the maximizer over $\mathcal{W} \times \mathbb{R}^m$ of the k-step logistic policy evaluation objective

$$\mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta}) \triangleq -\frac{1}{\eta} \log \sum_{i=1}^m (\mathbf{\Phi}^\intercal \mathbf{d}_{k-1})(i) e^{-\eta \delta^k_{\mathbf{w},\boldsymbol{\theta}}(i)} + (1-\gamma)\left\langle \boldsymbol{\nu}_0, \mathbf{V}_{\boldsymbol{\theta}}^k \right\rangle - \left\langle \boldsymbol{\rho}_{\mathbf{\Phi}}(\widehat{\pi_{\mathrm{E}}}), \mathbf{w} \right\rangle. \tag{5}$$

Moreover, it holds that $\mathcal{G}_k(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) = \langle \boldsymbol{\lambda}_k^\star, \mathbf{w}_k^\star \rangle - \langle \boldsymbol{\rho}_{\mathbf{\Phi}}(\widehat{\pi_{\mathrm{E}}}), \mathbf{w}_k^\star \rangle + \frac{1}{\eta} D(\boldsymbol{\lambda}_k^\star || \mathbf{\Phi}^\intercal \boldsymbol{\lambda}_{k-1}) + \frac{1}{\alpha} H(\mathbf{d}_k^\star || \mathbf{d}_{k-1})$. If in addition Assumption 1 holds, then $\mathbf{d}_k^\star$ is a valid occupancy measure, i.e., $\mathbf{d}_k^\star \in \mathfrak{F}$ and so $\mathbf{d}_k^\star = \boldsymbol{\mu}_{\pi_{\mathbf{d}_k^\star}}$.

The proof of Proposition 2 is broken down into a sequence of lemmas and is presented in Appendix E. It employs an `analytical-oracle` $\mathbf{g} : \mathcal{Y} \to \mathcal{X}$ given by

$$\mathbf{g}(\mathbf{y}; \mathbf{x}_k) \triangleq \operatorname*{arg\,min}_{\boldsymbol{\lambda}\in\Delta_{[m]}, \mathbf{d}\in\Delta_{\mathcal{S}\times\mathcal{A}}} \left\langle \mathbf{y}, \mathbf{A}\begin{bmatrix}\boldsymbol{\lambda}\\ \mathbf{d}\end{bmatrix} + \widehat{\mathbf{b}} \right\rangle + \frac{1}{\eta} D(\boldsymbol{\lambda}||\mathbf{\Phi}^\intercal \mathbf{d}_k) + \frac{1}{\alpha} H(\mathbf{d}||\mathbf{d}_k),$$

and a `max-oracle` $\mathbf{h}:\mathcal{X}\to\mathcal{Y}$ given by $\mathbf{h}(\mathbf{x})\triangleq\arg\max_{\mathbf{y}\in\mathcal{Y}}\langle\mathbf{y},\mathbf{A}\mathbf{g}(\mathbf{y};\mathbf{x})\rangle+\frac{1}{\tau}D_{\Omega}(\mathbf{g}(\mathbf{y};\mathbf{x})||\mathbf{x})$, where we used D_{Ω} to compact the two divergences. By noting that the PPM update Equation (2) can be rewritten as $\mathbf{x}_{k+1}=\mathbf{g}(\mathbf{h}(\mathbf{x}_k);\mathbf{x}_k)$, its analytical computation is reduced to the characterization of the two aforementioned oracles. In particular, the updates (3)–(4) come from the `analytical-oracle` while (5) is the objective of the `max-oracle`.

The choice of conditional entropy as Bregman divergence for the $\boldsymbol{\lambda}$ variable living in the probability simplex is standard in the optimization literature and is known to mitigate the effect of dimension. In particular, as noted in [85], the classic REPS algorithm [90] can be seen as mirror descent with relative entropy regularization. On the other hand, the choice of conditional entropy as Bregman divergence for the $\mathbf{d}$ variable is less standard and has been popularized by Q-REPS [14]. Such particular divergence leads to an actor-critic algorithm that comes with several merits. By Proposition 2, it is apparent that we get analytical softmin updates for the policy $\pi_{\mathbf{d}}$ rather than the occupancy measure $\mathbf{d}$. Moreover, these softmin updates are expressed in terms of the logistic Q-function and do not involve the unknown transition matrix $\mathbf{P}$. Consequently, we avoid the problematic occupancy measure approximation and the restrictive coherence assumption on the choice of features needed in [13, 58], as well as the biased policy updates appearing in REPS [90, 89]. In addition, the newly introduced logistic policy evaluation objective $\mathcal{G}_k(\mathbf{w},\boldsymbol{\theta})$ has several desired properties. It is concave and smooth in $(\mathbf{w},\boldsymbol{\theta})$ and has bounded gradients. Therefore, it does not suffer from the pathologies of the squared Bellman error [78] and does not require heuristic gradient clipping techniques. Moreover, unlike [58] it allows a model-free implementation without the need for a generative model (see Section 4.1)

We stress the fact that the `max-oracle` of our proximal point scheme performs the cost update and policy evaluation phases jointly. This is a rather novel feature of our algorithm that differs from the separate cost update and policy evaluation step used in recent theoretical imitation learning works [122, 105, 70]. Our joint optimization over cost and Q-functions avoids instability due to adversarial training and can also recover an explicit cost along with the Q-function without requiring knowledge or additional interaction with the environment (see Section 5). It is worth noting that application of primal-dual mirror descent to (SPP) does not have this favorable property. While in the standard MDP setting, proximal point and mirror descent coincide because of the linear objective, in imitation learning proximal point optimization makes a difference. In Appendix K, we include a more detailed discussion and numerical comparison between PPM and mirror descent updates.

4.1 Practical Implementation

Exact optimization of the logistic policy evaluation objective is infeasible in practical scenarios, due to unknown dynamics and limited computation power. In this section, we design a practical algorithm that uses only sample transitions by obtaining stochastic (albeit biased) gradient estimators.

Proposition 2 gives rise to Proximal Point Imitation Learning (P^{2}IL), a model-free actor-critic IRL algorithm described in Algorithm 1. The key feature of P^{2}IL is that the policy evaluation step involves optimization of a single smooth and concave objective over both cost and state-action value function parameters. In this way, we avoid instability or poor convergence in optimization due to nested policy evaluation and cost updates, as well as the undesirable properties of the widely used squared Bellman error. In particular, the kth iteration of P^{2}IL consists of the following two steps : (i) (**Critic Step**) Computation of an approximate maximizer $(\mathbf{w}_k,\boldsymbol{\theta}_k)\approx\arg\max_{\mathbf{w},\boldsymbol{\theta}}\mathcal{G}_k(\mathbf{w},\boldsymbol{\theta})$ of the concave logistic policy evaluation objective, by using a biased stochastic gradient ascent subroutine; (ii) (**Actor Step**) Soft-min policy update $\pi_k(a|s)\propto\pi_{k-1}(a|s)\,e^{-\alpha Q_{\boldsymbol{\theta}_k}(s,a)}$ expressed in terms of the logistic Q-function.

The domain Θ in Algorithm 1 is the ℓ_∞-ball with appropriately chosen radius D to be specified later (see Proposition 3). Moreover, $\Pi_\Theta(\mathbf{x})\triangleq\arg\min_{\mathbf{y}\in\Theta}\|\mathbf{x}-\mathbf{y}\|_2$ (resp. $\Pi_{\mathcal{W}}(\mathbf{w})$) denotes the Euclidean projection of $\mathbf{x}$ (resp. $\mathbf{w}$) onto Θ (resp. $\mathcal{W}$).

In order to estimate the gradients $\nabla_{\boldsymbol{\theta}}\,\mathcal{G}_k(\mathbf{w},\boldsymbol{\theta})$ and $\nabla_{\mathbf{w}}\,\mathcal{G}_k(\mathbf{w},\boldsymbol{\theta})$ we invoke the Biased Stochastic Gradient Estimator subroutine (BSGE) (Algorithm 2) given in Appendix H. By using the linear MDP Assumption 1 and leveraging ridge regression and plug-in estimators, the proposed stochastic gradients can be computed via simple linear algebra with computational complexity poly$(m,n(t))$, independent of the size of the state space.

Algorithm 1 Proximal Point Imitation Learning: $\texttt{P}^2\texttt{IL}(\mathbf{\Phi}, \mathcal{D}_{\mathrm{E}}, K, \eta, \alpha)$

```
Input: Feature matrix $\mathbf{\Phi}$, expert demonstrations $\mathcal{D}_{\mathrm{E}}$, number of iterations $K$, step sizes $\eta$ and $\alpha$, number of SGD iterations T, SGD learning rates $\boldsymbol{\beta} = \{\beta_t\}_{t=0}^{T-1}$, number-of-samples function $n : \mathbb{N} \to \mathbb{N}$
Initialize $\pi_0$ as uniform distribution over $\mathcal{A}$
Compute the empirical FEV $\boldsymbol{\rho}_{\mathbf{\Phi}}(\widehat{\pi_{\mathrm{E}}})$ using expert demonstrations $\mathcal{D}_{\mathrm{E}}$
for $k = 1, \ldots K$ do
  // Critic-step (policy evaluation)
  Initialize $\boldsymbol{\theta}_{k,0} = \mathbf{0}$ and $\mathbf{w}_{k,0} = \mathbf{0}$
  Run $\pi_{k-1}$ and collect i.i.d. samples $\mathcal{B}_k = \{(s_{k-1}^{(n)}, a_{k-1}^{(n)}, s_{k-1}^{\prime(n)})\}_{n=1}^{n(T)}$ such that $(s_{k-1}^{(n)}, a_{k-1}^{(n)}) \sim \boldsymbol{\mu}_{\pi_{k-1}}$ and $s_{k-1}^{\prime(n)} \sim \mathsf{P}(\cdot|s_{k-1}^{(n)}, a_{k-1}^{(n)})$
  for $t = 0, \ldots T-1$ do
    Compute biased stochastic gradient estimators
        $\big(\widehat{\nabla}_{\mathbf{w}}\mathcal{G}_k(\mathbf{w}_{k,t}, \boldsymbol{\theta}_{k,t}), \widehat{\nabla}_{\boldsymbol{\theta}}\mathcal{G}_k(\mathbf{w}_{k,t}, \boldsymbol{\theta}_{k,t})\big) = \mathrm{BSGE}\big(k, \mathbf{w}_{k,t}, \boldsymbol{\theta}_{k,t}, n(t)\big)$
    $\mathbf{w}_{k,t+1} = \Pi_{\mathcal{W}}\big(\mathbf{w}_{k,t} + \beta_t \widehat{\nabla}_{\mathbf{w}}\mathcal{G}_k(\mathbf{w}_{k,t}, \boldsymbol{\theta}_{k,t})\big)$
    $\boldsymbol{\theta}_{k,t+1} = \Pi_{\Theta}\big(\boldsymbol{\theta}_{k,t} + \beta_t \widehat{\nabla}_{\boldsymbol{\theta}}\mathcal{G}_k(\mathbf{w}_{k,t}, \boldsymbol{\theta}_{k,t})\big)$
  end for
  $(\mathbf{w}_k, \boldsymbol{\theta}_k) = (\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_{k,t}, \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\theta}_{k,t})$
  // Actor-step (policy update)
  Policy update: $\pi_k(a|s) \propto \pi_{k-1}(a|s)\, e^{-\alpha Q_{\boldsymbol{\theta}_k}(s,a)}$
end for
Output: Mixed policy $\widehat{\pi}_K$ of $\{\pi_k\}_{k\in[K]}$
```

4.2 Theoretical Analysis

The first step in our theoretical analysis is to study the propagation of optimization errors made by the algorithm on the true policy evaluation objective. In particular at each iteration step k, the ideal policy evaluation update $(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star)$ and the ideal policy update $\pi_k^\star$ are given by $(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) = \arg\max_{\mathbf{w},\boldsymbol{\theta}} \mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta})$, and $\pi_k^\star(a|s) = \pi_{k-1}(a|s)e^{-\alpha(Q_{\boldsymbol{\theta}_k^\star}(s,a) - V_{\boldsymbol{\theta}_k^\star}^k(s))}$. On the other hand, consider the realised policy evaluation update $(\mathbf{w}_k, \boldsymbol{\theta}_k)$ such that $\mathcal{G}_k(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) - \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k) = \epsilon_k$, the corresponding policy π_k given by $\pi_k = \pi_{k-1}(a|s)e^{-\alpha(Q_{\boldsymbol{\theta}_k}(s,a) - V_{\boldsymbol{\theta}_k}^k(s))}$, and let $\mathbf{d}_k \triangleq \boldsymbol{\mu}_{\pi_k}$. We denote by $\widehat{\pi}_K$ the extracted mixed policy of $\{\pi_k\}_{k=1}^K$. We are interested in upper-bounding the suboptimality gap $d_{\mathcal{C}}(\widehat{\pi}_K, \pi_{\mathrm{E}})$ of Algorithm 1 as a function of ε_k. To this end, we need the following assumption.

Assumption 2. *It holds that $\lambda_{\min}(\mathbb{E}_{(s,a)\sim\mathbf{d}_k} \boldsymbol{\phi}(s,a)\boldsymbol{\phi}(s,a)^\mathsf{T}) \geq \beta$, for all $k \in [K]$.*

Assumption 2 states that every occupancy measure $\mathbf{d}_k$ induces a positive definite feature covariance matrix, and so every policy π_k explores uniformly well in the feature space. This assumption is common in the RL theory literature [2, 46, 37, 66, 3, 7]. It is also related to the condition of persistent excitation from the control literature [81].

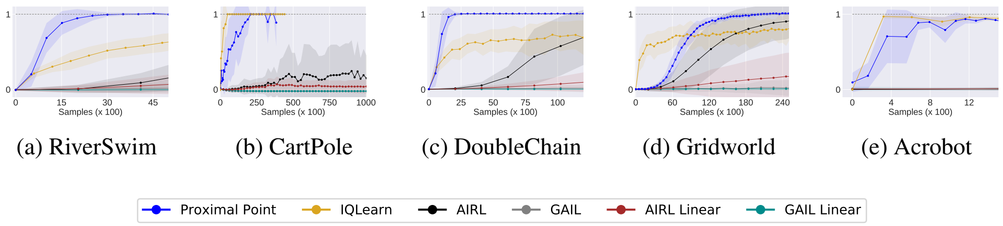
The following proposition ensures that $\max_{\mathbf{w},\boldsymbol{\theta}\in\mathcal{W}\times\mathbb{R}^m} \mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta}) = \max_{\mathbf{w},\boldsymbol{\theta}\in\mathcal{W}\times\Theta} \mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta})$. Therefore, this constraint does not change the problem optimality, but will considerably accelerate the convergence of the algorithm by considering smaller domains.

Proposition 3. *There exists a maximizer $\boldsymbol{\theta}_k^\star$ such that $\|\boldsymbol{\theta}_k^\star\|_\infty \leq \frac{1+|\log\beta|}{1-\gamma} \triangleq D$.*

We can now state our error propagation theorem.

Theorem 1. *Let $\widehat{\pi}_K$ be the output of running Algorithm 1 for K iterations, with $n_{\mathrm{E}} \geq \frac{2\log(\frac{2m}{\delta})}{\varepsilon^2}$ expert trajectories of length $H \geq \frac{1}{1-\gamma}\log(\frac{1}{\varepsilon})$. Let $C \triangleq \frac{1}{\beta\eta}\Big(\sqrt{\frac{2\alpha}{1-\gamma}} + \sqrt{8\eta}\Big) + \sqrt{\frac{18\alpha}{1-\gamma}}$. Then, with probability at least $1-\delta$, it holds that $d_{\mathcal{C}}(\widehat{\pi}_K, \pi_{\mathrm{E}}) \leq \frac{1}{K}\Big(\frac{D(\boldsymbol{\lambda}^*||\mathbf{\Phi}^\mathsf{T}\mathbf{d}_0)}{\eta} + \frac{H(\mathbf{d}^*||\mathbf{d}_0)}{\alpha} + C\sum_k\sqrt{\epsilon_k} + \sum_k \epsilon_k\Big) + \varepsilon$.*

By Theorem 1, whenever the policy evaluation errors ε_k, as well as the estimation error ε can be kept small, Algorithm 1 ouputs a policy $\widehat{\pi}_K$ with small suboptimality gap $\rho_{\mathbf{c}_{\text{true}}}(\widehat{\pi}_K) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\mathrm{E}})$.



(a) RiverSwim (b) CartPole (c) DoubleChain (d) Gridworld (e) Acrobot

Figure 1: **Online IL Experiments**. We show the total returns vs the number of env steps.

Notably, there is no direct dependence on the size of the state space or the dimension of the feature space. In the ideal case, where $\varepsilon_k = 0$ for all k, the convergence rate is $\mathcal{O}(1/K)$. The provided error propagation analysis still holds with general function approximation, i.e., in the context of deep RL. Indeed, by choosing $\mathbf{\Phi} = \mathbf{I}$, Assumption 1 is trivially satisfied and the $\boldsymbol{\theta}$ variable in the objective $\mathcal{G}_k$ is replaced by a Q-function. In practice, the estimation error ε can be made arbitrary small, by increasing the number of expert demonstrations n_{E}. Moreover, the next theorem ensures that under Assumptions 1 and 2 the biased stochastic gradient ascent (BSGA) subroutine has sublinear convergence rate.

Theorem 2. *Let $(\mathbf{w}_k, \boldsymbol{\theta}_k)$ be the output of the BSGA subroutine in Algorithm 1 for T iterations, with $n(t) \geq \max\left(\mathcal{O}\left(\frac{\gamma^2 mDt}{(\eta+\alpha)^2\beta}\log\frac{Tm}{\delta}\right), \mathcal{O}\left(\frac{mt}{(\eta+\alpha)^2\beta}\log\frac{Tm}{\delta}\right)\right)$ sample transitions, and learning rates $\beta_t = \mathcal{O}(\frac{1}{\sqrt{t}})$. Then, $\epsilon_k = \mathcal{G}_k(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) - \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k) \leq \mathcal{O}(\frac{\max\{\eta,1\}mD}{\beta\sqrt{T}})$, with probability $1-\delta$.*

Corollary 1 (Resource guarantees)**.** *Choose $\eta = \alpha = 1$ and let $K = \Omega\left(\epsilon^{-1}\right)$, $T = \Omega\left(\epsilon^{-4}\right)$. Then for $\Omega\left(KT\right) = \Omega\left(\epsilon^{-5}\right)$ sample transitions, $\Omega\left(\varepsilon^{-2}\right)$ expert trajectories and approximately solving $\Omega\left(\epsilon^{-1}\right)$ concave maximization problems, we can ensure $d_{\mathcal{C}}(\widehat{\pi}, \pi_{\mathrm{E}}) \leq \mathcal{O}(\epsilon + \varepsilon)$, with high probability.*

Offline Setting. Finally, we notice that using $\mathbf{\Phi}^\intercal \boldsymbol{\mu}_{\pi_{\mathrm{E}}}$ as the reference distribution for the relative entropy we can obtain an offline algorithm that does not require environment interactions. By reinterpreting smoothing [82] as one step of proximal point, and using similar arguments as in the proof of Theorem 1, we can provide similar theoretical guarantees for the offline setting. The formal statement of the theoretical result as well as the optimization of the empirical policy evaluation objective are presented in Appendix J (see Theorems 4 and 6).

5 Experiments

In this section, we demonstrate that our approach achieves convincing empirical performance in both online and offline IL settings on several environments.[1] The precise setting is detailed in Appendix L.

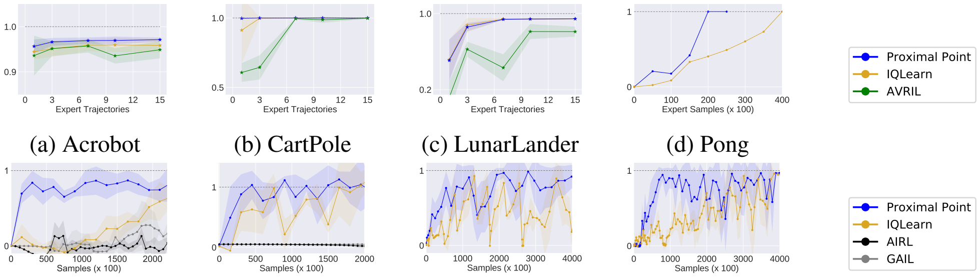
Online Setting. We first present results in various tabular environments where we can implement our algorithm without any practical relaxation outperforming GAIL [51], AIRL [38] and IQ-Learn [40]. Results are given in Figure 1. Good performance but inferior to IQ-Learn is observed also for continuous states environments (CartPole and Acrobot) where we used neural networks function approximation.

Offline Setting. Figures 2a to 2c shows that our method is competitive with the state-of-the-art offline IL methods IQLearn [40] and AVRIL [25] that recently showed performances superior to other methods like [54][64]. We also tried our algorithm in the complex image-based `Pong` task from the Atari suite. Figure 2d shows that the algorithm reaches the expert level after observing $2e5$ expert samples. We did not find AVRIL competitive in this setting, and skip it for brevity. In these settings, we verified that the algorithmic performance is convincing even for costs parameterized by neural networks.

Continuous control experiments. We attain the expert performance also in 2 MuJoCo environments: `Ant`, `HalfCheetah`, `Hopper`, and `Walker` (see Figures 2e to 2h). The additional difficulty in implementing the algorithm in continuous control experiments is that the analytical form of the policy

[1] The code is available at the following link `https://github.com/lviano/P2IL`.

improvement step is no longer computationally tractable because this would require to compute an integral over the continuous action space. Therefore, we approximated this update using the Soft Actor Critic (SAC) [44] algorithm. SAC requires environment samples making the algorithm online. The good empirical result opens the question of analyzing policy improvement errors as in [41].

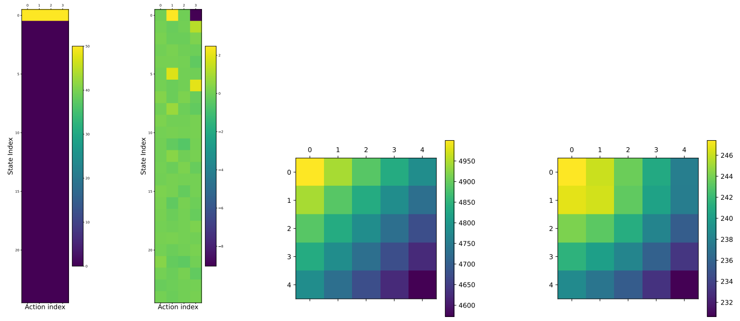


(a) Acrobot (b) CartPole (c) LunarLander (d) Pong

(e) HalfCheetah (f) Ant (g) Hopper (h) Walker2d

Figure 2: **Neural function approximation experiments.** Figures 2a to 2c show the total returns vs the number of expert trajectories. Figures 2e to 2h show the total returns vs the number of env steps. Figure 2d shows the total return vs the number of expert state-action pairs.

Recovered Costs. A unique algorithmic feature of the proposed methodology is that we can explicitly recover a cost along with the Q-function without requiring adversarial training. In Figure 3, we visualize our recovered costs in a simple 5x5 `Gridworld`. Most importantly, we verify that the recovered costs induce nearly optimal policies w.r.t. the unknown true cost function. Compared to IQ-Learn [40], we do not require knowledge or further interaction with the environment. Therefore, the recovered cost functions show promising transfer capability to new dynamics.

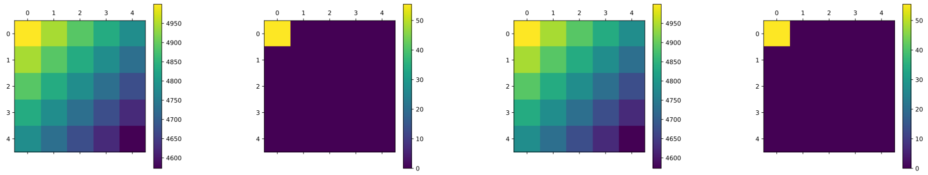


(a) $-\mathbf{c}_{\text{true}}$ (b) $-\mathbf{c}_K$ (c) -$V^{\star}_{\mathbf{c}_{\text{true}}}$ (d) -$V^{\star}_{\mathbf{c}_K}$

Figure 3: **Recovered Costs in `Gridworld`.** Comparison between the true cost $\mathbf{c}_{\text{true}}$ and the cost $\mathbf{c}_K$ recovered by $\mathtt{P^2IL}$. We notice that the optimal value functions $V^{\star}_{\mathbf{c}_{\text{true}}}$ and $V^{\star}_{\mathbf{c}_K}$ present the same pattern. Hence, the optimal policy with respect to $\mathbf{c}_K$ is nearly optimal with respect to $\mathbf{c}_{\text{true}}$.

Cost Transfer Setting. We experimented with a transfer cost setting on a `Gridworld` (Figure 4). We consider two different Gridworld MDP environments, say M and $\widetilde{M}$, with opposite action effects. This means that action `Down` in $\widetilde{M}$ corresponds to action `Left` in M and vice versa. Similarly, the effects of `Up` and `Right` are swapped between $\widetilde{M}$ and M. We denote by $\mathbf{V}^{\pi}_{\widetilde{M},\mathbf{c}_{\text{true}}}$ (resp. $\mathbf{V}^{\star}_{\widetilde{M},\mathbf{c}_{\text{true}}}$) the value function of policy π (resp. optimal value function) in the MDP environment $\widetilde{M}$ with cost function $\mathbf{c}_{\text{true}}$. Moreover, we denote by $\pi^{\star}_{M,\mathbf{c}}$ the optimal policy in the MDP environment M under cost function $\mathbf{c}$. Figure (a) gives the corresponding optimal value function. Figure (b) presents the value function of the expert policy $\pi_{\text{E}} = \pi^{\star}_{M,\mathbf{c}_{\text{true}}}$ used as target by $\mathtt{P^2IL}$. Figure (d) shows the value function of the learned imitating policy π_K from $\mathtt{P^2IL}$. Finally, Figure (b) depicts the value function of the optimal policy $\pi^{\star}_{\widetilde{M},\mathbf{c}_K}$ for the environment $\widetilde{M}$ endowed with the recovered cost function $\mathbf{c}_K$ by

P^{2}IL (with access to samples from M). We conclude that the policy $\pi^\star_{\widetilde{M},\mathbf{c}_K}$ is optimal in $\widetilde{M}$ with cost $\mathbf{c}_{\text{true}}$. By contrast, the expert policy $\pi_{\text{E}} = \pi^\star_{M,\mathbf{c}_{\text{true}}}$ used as target by P^{2}IL performs poorly and as a consequence also the imitating policy π_K does so. All in all, we notice that the recovered cost induces an optimal policy for the new dynamics while the imitating policy fails. Albeit, cost transfer is successful in this experiment we do not expect this fact to be true in general because we do not tackle the issue of cost shaping [87].



(a) $-\mathbf{V}^\star_{\widetilde{M},\mathbf{c}_{\text{true}}}$ (b) $-\mathbf{V}^{\pi^\star_{M,\mathbf{c}_{\text{true}}}}_{\widetilde{M},\mathbf{c}_{\text{true}}}$ (c) $-\mathbf{V}^{\pi^\star_{\widetilde{M},\mathbf{c}_K}}_{\widetilde{M},\mathbf{c}_{\text{true}}}$ (d) $-\mathbf{V}^{\pi_K}_{\widetilde{M},\mathbf{c}_{\text{true}}}$

Figure 4: **Cost Transfer Experiment in Gridworld.** We compare the performance of several policies in the new MDP environment $\widetilde{M}$ with cost function $\mathbf{c}_{\text{true}}$. We notice that the recovered cost induces an optimal policy for the new dynamics while the imitating policy fails.

6 Discussion and Outlook

In this work, we studied a Proximal Point Imitation Learning (P^{2}IL) algorithm with both theoretical guarantees and convincing empirical performance. Our methodology is rooted in classical optimization tools and the LP approach to MDPs. The most significant merits of P^{2}IL are the following: (i) It optimizes a convex and smooth logistic Bellman evaluation objective over both cost and Q-functions. In particular, it avoids instability due to adversarial training and can also recover an explicit cost along with Q function; (ii) In the context of linear MDPs, it comes with efficient resource guarantees and error bounds for the suboptimality of the learned policy (Theorem 2 and Corollary 1). In particular, given $\text{poly}(1/\varepsilon, \log(1/\delta), m)$ many samples , it recovers an ε-optimal policy, with probability $1 - \delta$. Notably, the bound is independent of the size of the state-action space; (iii) Beyond the linear MDP setting, it can be implemented in a model-free manner, for both online and offline setups, with general function approximation without losing its theoretical specifications. This is justified by providing an error propagation analysis (Theorems 1 and 4), guaranteeing that small optimization errors lead to high-quality output policy; (iv) It enjoys not only strong theoretical guarantees but also favorable empirical performance. At the same time, our newly introduced methods bring challenges and open questions. One interesting question is whether one can accelerate the PPM updates and improve the convergence rate. Another direction for future work is to provide rigorous arguments for the near-optimality of the recovered cost function. On the practical side, we plan to conduct experiments in more challenging environments than MuJoCo and Atari. We hope our new techniques will be useful to future algorithm designers and lay the foundations for overcoming current limitations and challenges. In Appendix B, we point out in detail a few interesting future directions.

Acknowledgements

The authors would like to thank the anonymous reviewer for their suggestions to improve the presentation and for motivating us to inspect the recovered cost function. This work has received funding from the Enterprise for Society Center (E4S), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement OCAL, No. 787845, the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 725594 - time-data), the Swiss National Science Foundation (SNSF) under grant number 200021_205011. Gergely Neu was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 950180). Luca Viano acknowledges travel support from ELISE (GA no 951847).

References

[1] Y. Abbasi-Yadkori, P. L. Bartlett, and A. Malek. Linear programming for large-scale Markov decision problems. In *International Conference on Machine Learning (ICML)*, 2014.

[2] Y. Abbasi-Yadkori, P. Bartlett, K. Bhatia, N. Lazic, C. Szepesvari, and G. Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning (ICML)*, 2019.

[3] Y. Abbasi-Yadkori, N. Lazic, C. Szepesvari, and G. Weisz. Exploration-enhanced politex. *arXiv:1908.10479*, 2019.

[4] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2004.

[5] P. Abbeel, D. Dolgov, A. Y. Ng, and S. Thrun. Apprenticeship learning for motion planning with application to parking lot navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[6] J. D. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Annual Conference on Learning Theory (COLT)*, 2008.

[7] A. Agarwal, S. Kakade, A. Krishnamurthy, and W. Sun. Flambe: Structural complexity and representation learning of low rank MDPs. *Advances in neural information processing systems (NeurIPS)*, 2020.

[8] A. Ayoub, Z. Jia, C. Szepesvari, M. Wang, and L. Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning (ICML)*, 2020.

[9] J. A. Bagnell and J. G. Schneider. Covariant policy search. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[10] G. Banjac and J. Lygeros. A data-driven policy iteration scheme based on linear programming. In *IEEE Conference on Decision and Control (CDC)*, 2019.

[11] P. Barde, J. Roy, W. Jeon, J. Pineau, C. Pal, and D. Nowrouzezahrai. Adversarial soft advantage fitting: Imitation learning without policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[12] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, pages 834–846, 1983.

[13] J. Bas-Serrano and G. Neu. Faster saddle-point optimization for solving large-scale Markov decision processes. In *Conference on Learning for Dynamics and Control (L4DC)*, 2020.

[14] J. Bas-Serrano, S. Curi, A. Krause, and G. Neu. Logistic Q-learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[15] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[16] P. N. Beuchat, A. Georghiou, and J. Lygeros. Performance guarantees for model-based approximate dynamic programming in continuous spaces. *IEEE Transactions on Automatic Control*, 65(1):143–158, 2020.

[17] V. S. Borkar. A convex analytic approach to Markov decision processes. *Probability Theory and Related Fields*, 78(4):583–602, 1988.

[18] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv:1606.01540*, 2016.

[20] D. S. Brown, R. Coleman, R. Srinivasan, and S. Niekum. Safe imitation learning via fast Bayesian reward inference from preferences. In *International Conference on Machine Learning (ICML)*, 2020.

[21] Q. Cai, M. Hong, Y. Chen, and Z. Wang. On the global convergence of imitation learning: a case for linear quadratic regulator. *arXiv:1901.03674*, 2019.

[22] Q. Cai, Z. Yang, C. Jin, and Z. Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning (ICML)*, 2020.

[23] Y. Carmon, Y. Jin, A. Sidford, and K. Tian. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[24] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[25] A. J. Chan and M. van der Schaar. Scalable Bayesian inverse reinforcement learning. *arXiv:2102.06483*, 2021.

[26] J. Chang, M. Uehara, D. Sreenivas, R. Kidambi, and W. Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems (NeuriPS)*, 2021.

[27] A. Charpentier, R. Elie, and C. Remlinger. Reinforcement learning in economics and finance. *arXiv:20031004*, 2020.

[28] M. Chen, Y. Wang, T. Liu, Z. Yang, X. Li, Z. Wang, and T. Zhao. On computation and generalization of generative adversarial imitation learning. *International Conference on Learning Representations (ICLR)*, 2020.

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[30] Y. Chen, L. Li, and M. Wang. Scalable bilinear π learning using state and action features. In *International Conference on Machine Learning (ICML)*, 2018.

[31] C.-A. Cheng, R. T. des Combes, B. Boots, and G. Gordon. A reduction from reinforcement learning to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[32] R. Dadashi, L. Hussenot, M. Geist, and O. Pietquin. Primal Wasserstein imitation learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[33] D. P. De Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–865, 2003.

[34] D. P. De Farias and B. Van Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[35] G. T. De Ghellinck and G. D. Eppen. Linear programming solutions for separable Markovian decision problems. *Management Science*, 13(5):371–394, 1967.

[36] E. V. Denardo. On linear programming in a Markov decision problem. *Management Science*, 16(5):281–288, 1970.

[37] Y. Duan, Z. Jia, and M. Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning (ICML)*, 2020.

[38] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

[39] T. Furmston and D. Barber. Variational methods for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[40] D. Garg, S. Chakraborty, C. Cundy, J. Song, and S. Ermon. IQ-learn: Inverse soft-Q learning for imitation. In *Advances in Neural Information Processing Systems (NeuRIPS)*, 2021.

[41] M. Geist, B. Scherrer, and O. Pietquin. A Theory of Regularized Markov Decision Processes. In *International Conference on Machine Learning (ICML)*, 2019.

[42] A. Geramifard, C. Dann, R. H. Klein, W. Dabney, and J. P. How. RLPy: A value-function-based reinforcement learning framework for education and research. *Journal of Machine Learning Research*, 16(46):1573–1578, 2015.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[44] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018.

[45] F. Hanzely, P. Richtarik, and L. Xiao. Accelerated bregman proximal gradient methods for relatively smooth convex optimization. *Computational Optimization and Applications*, 79(2): 405–440, 2021.

[46] B. Hao, T. Lattimore, C. Szepesvári, and M. Wang. Online sparse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[47] E. Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

[48] O. Hernández-Lerma and J. B. Lasserre. *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Springer-Verlag New York, 1996.

[49] O. Hernández-Lerma and J. B. Lasserre. *Further Topics on Discrete-Time Markov Control Processes*. Springer-Verlag New York, 1999.

[50] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[51] J. Ho, J. K. Gupta, and S. Ermon. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning (ICML)*, 2016.

[52] S. R. Howard, A. Ramdas, J. McAuliffe, and J. Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2), 2021.

[53] D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. In *Conference on Learning Theory (COLT)*, 2012.

[54] D. Jarrett, I. Bica, and M. van der Schaar. Strictly batch imitation learning by energy-based distribution matching. *arXiv:2006.14154*, 2021.

[55] Y. Jin and A. Sidford. Efficiently solving MDPs with stochastic mirror descent. In *International Conference on Machine Learning (ICML)*, 2020.

[56] S. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[57] G. Kalweit, H. Maria, M. Werling, and J. Boedecker. Deep inverse Q-learning with constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[58] A. Kamoutsi, G. Banjac, and J. Lygeros. Efficient performance bounds for primal-dual reinforcement learning from demonstrations. In *International Conference on Machine Learning (ICML)*, 2021.

[59] L. Ke, S. Choudhury, M. Barnes, W. Sun, G. Lee, and S. Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2020.

[60] C. Kent, J. Li, J. Blanchet, and P. Glynn. Modified Frank Wolfe in probability space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[61] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[62] W. B. Knox, A. Allievi, H. Banzhaf, F. Schmitt, and P. Stone. Reward (mis)design for autonomous driving, 2021.

[63] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[64] I. Kostrikov, O. Nachum, and J. Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations (ICLR)*, 2020.

[65] C. Lakshminarayanan, S. Bhatnagar, and C. Szepesvári. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic Control*, 63(4): 1185–1191, 2018.

[66] N. Lazic, D. Yin, M. Farajtabar, N. Levine, D. Gorur, C. Harris, and D. Schuurmans. A maximum-entropy approach to off-policy evaluation in average-reward MDPs. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[67] D. Lee and N. He. Stochastic primal-dual Q-learning algorithm for discounted MDPs. In *American Control Conference (ACC)*, 2019.

[68] S. Levine, Z. Popović, and V. Koltun. Feature construction for inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[69] S. Levine, Z. Popović, and V. Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[70] Z. Liu, Y. Zhang, Z. Fu, Z. Yang, and Z. Wang. Learning from demonstration: Provably efficient adversarial policy imitation with linear function approximation. In *International Conference on Machine Learning (ICML)*, 2022.

[71] F. Lu, P. G. Mehta, S. P. Meyn, and G. Neu. Convex q-learning. In *2021 American Control Conference (ACC)*, pages 4749–4756, 2021. doi: 10.23919/ACC50511.2021.9483244.

[72] Y. Malitsky and M. K. Tam. A forward-backward splitting method for monotone inclusions without cocoercivity. *SIAM Journal on Optimization*, 30(2):1451–1472, 2020.

[73] A. Manne. Linear programming and sequential decisions. *Management Science*, 6(3):259–267, 1960.

[74] A. Martinelli, M. Gargiani, and J. Lygeros. Data-driven optimal control with a relaxed linear program. *arXiv:2003.08721*, 2020.

[75] C. McDiarmid. *Concentration*, pages 195–248. Springer Berlin Heidelberg, 1998.

[76] P. Mehta and S. Meyn. Q-learning and pontryagin's minimum principle. In *IEEE Conference on Decision and Control (CDC)*, 2009.

[77] P. G. Mehta and S. P. Meyn. Convex Q-learning, Part 1: Deterministic optimal control. *arXiv:2008.03559*, 2020.

[78] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[79] P. Mohajerin Esfahani, T. Sutter, D. Kuhn, and J. Lygeros. From infinite to finite programs: explicit error bounds with applications to approximate dynamic programming. *SIAM Journal on Optimization*, 28(3):1968–1998, 2018.

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

[81] K. S. Narendra and A. M. Annaswamy. Persistent excitation in adaptive systems. *International Journal of Control*, 45(1):127–160, 1987.

[82] Y. Nesterov. Smooth minimization of nonsmooth functions. *Math. Programming*, 103:127–152, 2005.

[83] G. Neu and J. Olkhovskaya. Online learning in mdps with linear function approximation and bandit feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[84] G. Neu and C. Pike-Burke. A unifying view of optimism in episodic reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[85] G. Neu and C. Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

[86] G. Neu, A. Jonsson, and V. Gómez. A unified view of entropy-regularized Markov decision processes. *arXiv:1705.07798*, 2017.

[87] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2000.

[88] T. Osa, J. Pajarinen, G. Neumann, J. Bagnell, P. Abbeel, and J. Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 2018.

[89] A. Pacchiano, J. Lee, P. Bartlett, and O. Nachum. Near optimal policy optimization via REPS. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[90] J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *National Conference on Artificial Intelligence (AAAI)*, 2010.

[91] M. Petrik and S. Zilberstein. Constraint relaxation in approximate linear programs. In *International Conference on Machine Learning (ICML)*, pages 809–816, 2009.

[92] M. Petrik, G. Taylor, R. Parr, and S. Zilberstein. Feature selection using regularization in approximate linear programs for Markov decision processes. In *International Conference on International Conference on Machine Learning (ICML)*, 2010.

[93] D. A. Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

[94] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994.

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

[96] S. Reddy, A. D. Dragan, and S. Levine. SQIL: imitation learning via regularized behavioral cloning. *arXiv:1905.11108*, 2019.

[97] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[98] S. Ross and D. Bagnell. Efficient reductions for imitation learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[99] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[100] S. Russell. Learning agents for uncertain environments (extended abstract). In *Annual Conference on Computational Learning Theory (COLT)*, 1998.

[101] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

[102] P. J. Schweitzer and A. Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110(2):568–582, 1985.

[103] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[104] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[105] L. Shani, T. Zahavy, and S. Mannor. Online apprenticeship learning. *arXiv:2102.06924*, 2021.

[106] R. Shariff and C. Szepesvári. Efficient planning in large MDPs with weak linear function approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[107] M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

[108] A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[109] T. Sutter, A. Kamoutsi, P. E. Esfahani, and J. Lygeros. Data-driven approximate dynamic programming: A linear programming approach. In *IEEE Conference on Decision and Control (CDC)*, 2017.

[110] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, second edition, 2018.

[111] U. Syed and R. E. Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2007.

[112] U. Syed, M. Bowling, and R. Schapire. Apprenticeship learning using linear programming. In *International Conference on Machine Learning (ICML)*, 2008.

[113] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5026–5033, 2012.

[114] N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, and M. Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

[115] M. Wang. Randomized linear programming solves the Markov decision problem in nearly linear (sometimes sublinear) time. *Mathematics of Operations Research*, 45(2):517–546, 2020.

[116] R. Wang, S. S. Du, L. Yang, and R. R. Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[117] H. Xiao, M. Herman, J. Wagner, S. Ziesche, J. Etesami, and T. H. Linh. Wasserstein adversarial imitation learning. *arXiv:1906.08113*, 2019.

[118] T. Xu, Z. Li, and Y. Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[119] S. Yan and N. He. Bregman augmented lagrangian and its acceleration. *arXiv preprint arXiv:2002.06315*, 2020.

[120] L. Yang and K.-C. Toh. Bregman proximal point algorithm revisited: a new inexact version and its variant. *arXiv preprint arXiv:2105.10370*, 2021.

[121] L. Yang and M. Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning (ICML)*, 2019.

[122] Y. Zhang, Q. Cai, Z. Yang, and Z. Wang. Generative adversarial imitation learning with neural network parameterization: global optimality and convergence rate. In *International Conference on Machine Learning (ICML)*, 2020.

[123] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *National Conference on Artificial Intelligence (AAAI)*, 2008.

[124] A. Zimin and G. Neu. Online learning in episodic Markovian decision processes by relative entropy policy search. *Advances in neural information processing systems (NeurIPS)*, 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We reflect the contribution described in the introduction with the theoretical results in Section 4, Section 4.2
   (b) Did you describe the limitations of your work? [Yes] We state the assumptions needed for the analysis in Sections 3 and 4.2
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] The work is mainly theoretical, we do not foresee potential negative impacts on the society.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We acknowledge habing read the review guidelines.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We state the assumptions needed for the analysis in Section 3 Section 4.2
   (b) Did you include complete proofs of all theoretical results? [Yes] The main results are stated in Section 4.2 and the proofs are included as supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code is included in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Training details are provided in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We averaged multiple seeds whenever possible computationally. We ran a single seed for the computational expensive Pong environment.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We specified the resource in the Supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite [40] for using their codebase and their expert data.
   (b) Did you mention the license of the assets? [Yes] We mention the license in the Supplementary.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We attach the code to the supplementary.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We discuss this in the supplemnetary material
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We do not use personal data.
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not involve human partecipants.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not involve human partecipants.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not involve human partecipants.