# Proximal Point Imitation Learning

**Anonymous Author(s)**
Affiliation
Address
email

## 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 [1], robotics [2], and economics/finance [3] 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 [4–8], resource guarantees are still widely missing since the optimization literature focuses on the resource (R3) where IL literature mainly focuses on 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).

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

**Our contributions:** This work aims at designing an algorithm enjoying resource guarantees in infinite horizon Linear MDPs [9–15] without requiring coherence assumptions on the features and bounded concentrability coefficients. To our knowledge, such guarantees in this setting are provided for the first time.

We reach this goal by introducing a novel splitting approach on the classical LP formulation (see Section 3) for IL inspired from [16, 12] and noticing that the proximal-point method (PPM) [17] can be implemented solving a concave and smooth maximization in terms of both cost and $Q$-functions, instead of the conventional alternated cost and policy updates (see Section 4). Furthermore, we show in Section 4.1 that the maximization problem can be implemented efficiently in a model-free manner.

We account for potential errors in the convex maximization problem, presenting an error propagation analysis that leads to rigorous guarantees for both online and offline setting (see Section 5). Finally, we implement our method using both linear or neural networks function approximations, observing convincing results (see Section 6). As an independent interest, our techniques can be used to improve upon the best rate for REPS in the tabular setting [18] and to extend the guarantees to Linear MDPs.

**Related Literature.** The first algorithm addressing the imitation learning problem is behavioral cloning [19]. Due to the covariate shift problem [20, 21], it has low efficiency in terms of expert trajectories (R1). To address this issue, [22–31] 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 [16] highlights that repeated calls to a RL routine can be avoided. This work inspired generative adversarial imitation learning (GAIL) [4] and other follow-up works [5–8] 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 [32, 33] in IL bypasses the need of optimizing over cost functions by directly recovery a policy parametrized by state action value functions. These works achieved strong empirical performance, but guarantees are still lacking in the literature.

Existing IL theoretical papers assume either finite horizon setting [34], tabular MDPs [35], or the infinite horizon case but with restrictive assumptions, such as linear quadratic regulator setting [36], continuous kernelized nonlinear regulator [37, 38], generative model, coherence assumption [39, 12], bounded concentrability coefficients [40, 41].

Our work applies the technique known as regularization in the online learning literature [42, 43] and Bregman proximal-point or smoothing in optimization literature [17, 44] to the LP formulation for MDP [45–63, 10, 64]. From this perspective, we can see Deep Inverse Q-learning [33] and IQ-Learn [32] 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 [65, 12, 18] and O-REPS [66] 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 [67, 68] are equivalent, recognizing that they are *not equivalent* in IL is critical for stronger empirical performance.

# 2 Backgroud

## 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, $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ 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* $\pi\colon \mathcal{S} \to \Delta_{\mathcal{A}}$ 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 $\pi$.

The *state value function* $\mathbf{V}^\pi_{\mathbf{c}} \in \mathbb{R}^{|\mathcal{S}|}$ of $\pi$, 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 $\pi$ 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 $\pi$, we define the *normalized state-action occupancy measure* $\boldsymbol{\mu}_\pi \in \Delta_{\mathcal{S}\times\mathcal{A}}$, by $\mu_\pi(s,a) \triangleq (1-\gamma)\sum_{t=0}^\infty \gamma^t\, \mathbb{P}^\pi_{\boldsymbol{\nu}_0}[s_t = s, a_t = a]$, where $\mathbb{P}^\pi_{\boldsymbol{\nu}_0}[\cdot]$ denotes the probability of an event when following $\pi$ 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 $\pi_{\text{E}}$ in the environment. The goal is to learn a policy that performs better than $\pi_{\text{E}}$ with respect to the unknown $\mathbf{c}_{\text{true}}$. To this end, we adopt the AL formalism [24, 16, 69, 4, 35], which carries the assumption that $\mathbf{c}_{\text{true}}$ belongs to a class of cost functions $\mathcal{C}$. We then seek an *apprentice policy* $\pi_{\text{A}}$ 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 $d_{\mathcal{C}}(\pi,\pi_{\text{E}}) \triangleq \max_{\mathbf{c}\in\mathcal{C}}\big(\rho_{\mathbf{c}}(\pi) - \rho_{\mathbf{c}}(\pi_{\text{E}})\big)$ defines the $\mathcal{C}$-distance between $\pi$ and $\pi_{\text{E}}$ [4, 70, 40, 34]. Then, $\pi_{\text{A}}$ satisfies the goal of IL, since it holds that $\rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{A}}) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\text{E}}) \le \zeta^\star \le 0$. 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 $\pi_{\text{E}}$ [4], 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 $\bar{d}_{\mathcal{C}}(\boldsymbol{\mu}_\pi, \boldsymbol{\mu}_{\pi_{\text{E}}}) \triangleq \max_{\mathbf{c}\in\mathcal{C}}\big(\langle\boldsymbol{\mu}_\pi,\mathbf{c}\rangle - \langle\boldsymbol{\mu}_{\pi_{\text{E}}},\mathbf{c}\rangle\big)$, it can be interpreted as an *integral probability metric* [71, 72] 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 [73, 74] for $\mathcal{C} = \text{Lip}_1(\mathcal{S}\times\mathcal{A})$, the total variation for $\mathcal{C} = \{\mathbf{c} \mid \|\mathbf{c}\|_\infty \le 1\}$, or the maximum mean discrepancy for $\mathcal{C} = \{\mathbf{c} \mid \|\mathbf{c}\|_{\mathcal{H}} \le 1\}$, 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 $\mathcal{H}$ [75].

In our theoretical analysis, we focus on linearly parameterized cost classes [26, 16, 4, 34, 35] 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 $\|\boldsymbol{\phi}_i\|_1 \le 1$ 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 [24]. Moreover, the cost features can be complex nonlinear functions that can be obtained via unsupervised learning from raw state observations [76, 77]. The matrix $\boldsymbol{\Phi} \triangleq [\boldsymbol{\phi}_1 \quad \dots \quad \boldsymbol{\phi}_m]$ gives rise a *feature expectation vector* (FEV) $\boldsymbol{\rho}_{\boldsymbol{\Phi}}(\pi) \triangleq (\rho_{\boldsymbol{\phi}_1}(\pi_{\text{E}}), \dots, \rho_{\boldsymbol{\phi}_m}(\pi_{\text{E}}))^\intercal \in \mathbb{R}^m$ of a policy $\pi$. Then, by choosing $\mathcal{W}$ to be the $\ell_2$ unit ball $B_1^m \triangleq \{\mathbf{w}\in\mathbb{R}^m \mid \|\mathbf{w}\|_2 \le 1\}$ [24], we get a *feature expectation matching* objective $d_{\mathcal{C}}(\pi,\pi_{\pi_{\text{E}}}) = \|\boldsymbol{\rho}_{\boldsymbol{\Phi}}(\pi) - \boldsymbol{\rho}_{\boldsymbol{\Phi}}(\pi_{\text{E}})\|_2$, while for $\mathcal{W}$ being the probability simplex $\Delta_{[m]}$ [26, 16] we have a worst-case excess cost objective $d_{\mathcal{C}}(\pi,\pi_{\pi_{\text{E}}}) = \max_{i\in[m]}\big(\rho_{\boldsymbol{\phi}_i}(\pi) - \rho_{\boldsymbol{\phi}_i}(\pi_{\text{E}})\big)$. 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 [16], 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** (78)**.** *We have that $\boldsymbol{\mu} \in \mathfrak{F}$ if and only if there exists a unique stationary Markov policy $\pi$ 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 $\bar{d}_{\mathcal{C}}(\boldsymbol{\mu}, \boldsymbol{\mu}_{\pi_{\mathrm{E}}}) = \max_{\mathbf{w} \in \mathcal{W}} \left\langle \boldsymbol{\mu} - \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_{\mathbf{w}} \right\rangle$, 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 [16, 39], while for $\mathcal{W} = B_1^m$ we get a quadratic objective with linear constraints [24].

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 [79] and has been recently revisited by [12, 15, 61, 80–82]. 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, we shed light to its dual and provide an interpretation of the dual optimizers. For brevity, we focus on the case $\mathcal{W} = B_1^m$. The proof can be found in Appendix 1 and is based on strong duality between the two convex programs.

**Proposition 2.** *The dual convex program is given by*

$$\zeta^\star = \max_{(\mathbf{w}, \mathbf{V}, \mathbf{Q})} \Big\{ (1-\gamma) \left\langle \boldsymbol{\nu}_0, \mathbf{V} \right\rangle - \left\langle \boldsymbol{\mu}_{\pi_{\mathrm{E}}}, \mathbf{c}_{\mathbf{w}} \right\rangle \mid \mathbf{Q} \geq \mathbf{B}\mathbf{V},\ \mathbf{Q} = \mathbf{c}_{\mathbf{w}} + \gamma \mathbf{P}\mathbf{V}, \\ \mathbf{V} \in \mathbb{R}^{|\mathcal{S}|},\ \mathbf{Q} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|},\ \mathbf{w} \in \mathcal{W} \Big\}. \tag{Dual}$$

*Moreover, for $\mathcal{W} = B_1^m$, a triple $(\mathbf{V}_{\mathrm{A}}, \mathbf{Q}_{\mathrm{A}}, \mathbf{w}_{\mathrm{A}})$ is dual optimal if and only if (i) $\pi_{\mathrm{E}}$ is optimal for the RL problem with cost $\mathbf{c} = \mathbf{c}_{\mathbf{w}_{\mathrm{A}}}$, (ii) $\mathbf{V}_{\mathrm{A}} = \mathbf{V}^\star_{\mathbf{w}_{\mathrm{A}}}$, (iii) $\mathbf{Q}_{\mathrm{A}} = \mathbf{Q}^\star_{\mathbf{w}_{\mathrm{A}}}$, and (iv) $\mathbf{w}_{\mathrm{A}} \in \mathcal{W}$. In particular, $(\mathbf{V}^\star_{\mathbf{w}_{\text{true}}}, \mathbf{Q}^\star_{\mathbf{w}_{\text{true}}}, \mathbf{w}_{\text{true}})$ is a dual optimizer.*

Proposition 2 states that the set of dual optimal costs $\mathbf{c}_{\mathbf{w}_{\mathrm{A}}}$ is the set of costs in $\mathcal{C}$ for which the expert is optimal. In this case, the optimal $\mathbf{V}_{\mathrm{A}}$ coincides with the corresponding optimal value function[1], while the optimal $\mathbf{Q}_{\mathrm{A}}$ coincides with the corresponding optimal state-action value function. In particular, the true weights $\mathbf{w}_{\text{true}}$, the true optimal value function $\mathbf{V}^\star_{\mathbf{w}_{\text{true}}}$ and the true optimal state-action value function $\mathbf{Q}^\star_{\mathbf{w}_{\text{true}}}$ are dual optimizers. Therefore, the presented $Q$-convex approach allows to recover an optimal solution to the original problem (1) from both the (Primal) and (Dual) formulations: it can be obtained either as the induced policy of a primal optimal occupancy measure or as a greedy policy associated to a dual optimal $Q$-function. In the next section, we will generalize the later observation to implement PPM using softmin updates in terms of $Q$-functions.

For our theoretical analysis we focus on the linear MDP setting [10], 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 $\boldsymbol{\Phi}$.

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

[1] To be precise, this is the case if $\boldsymbol{\nu}_0 \in \mathbb{R}^{|\mathcal{S}|}_{++}$, otherwise they coincide $\boldsymbol{\nu}_0$-almost surely.

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 [40, 34, 35] 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

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

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 [15, 12, 80], for linear MDPs, the set of occupancy measures $\mathfrak{F}$ can be completely characterized by the set $\mathfrak{M}_{\mathbf{\Phi}}$ (c.f., Proposition 3). 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 Inverse Q-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 A), $\mathbf{x} \triangleq [\boldsymbol{\lambda}^\intercal,\mathbf{d}^\intercal]^\intercal$, $\mathbf{y} \triangleq [\mathbf{w}^\intercal,\mathbf{V}^\intercal,\boldsymbol{\theta}^\intercal]^\intercal$, $\mathcal{X} \triangleq \Delta_{[m]}\times\Delta_{\mathcal{S}\times\mathcal{A}}$, 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 $\pi_{\mathrm{E}}$, but instead can observe a finite set of i.i.d. sample trajectories $\mathcal{D}_{\mathrm{E}} \triangleq \{(x_0^{(l)},a_0^{(l)},x_1^{(l)},a_1^{(l)},\dots,x_H^{(l)},a_H^{(l)})\}_{l=1}^s \sim \pi_{\mathrm{E}}$, 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

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

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 3.** *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*

$$\lambda_k^\star(s,a) \propto (\mathbf{\Phi}^\intercal\mathbf{d}_{k-1})(i)\,e^{-\eta\delta_{\mathbf{w}_k^\star,\theta_k^\star}^k(i)}, \tag{3}$$

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

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

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

*Moreover, it holds that* $\mathcal{G}_k(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) = \langle \boldsymbol{\lambda}_k^\star, \mathbf{w}_k^\star \rangle - \langle \boldsymbol{\rho}_{\boldsymbol{\Phi}}(\widehat{\pi_{\mathrm{E}}}), \mathbf{w}_k^\star \rangle + \frac{1}{\eta} D(\boldsymbol{\lambda}_k^\star || \boldsymbol{\Phi}^\intercal \boldsymbol{\lambda}_{k-1}) + \frac{1}{\alpha} H(\mathbf{d}_k^\star || \mathbf{d}_{k-1})$*. 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 3 is broken down into a sequence of lemmas and is presented in Appendix B.1. It employs an `analytical-oracle` $\mathbf{g} : \mathcal{Y} \to \mathcal{X}$ given by

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

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 $D_\Omega$ 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 [27], the classic REPS algorithm [65] 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 [12]. Such particular divergence leads to an actor-critic algorithm that comes with several merits. By Proposition 3, it is apparent that the softmin policy update expressed in terms of the state-action logistic value function $\mathbf{Q}_{\boldsymbol{\theta}}$ does not need knowledge of the dynamics. Consequently, we avoid the restrictive coherence assumption on the choice of features needed in [62, 39], as well as the biased policy updates appearing in REPS [65, 18]. In addition, note that the newly introduced policy evaluation objective $\mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta})$ is concave and smooth in $(\mathbf{w}, \boldsymbol{\theta})$. Moreover, it has bounded gradients and therefore avoids the need of heuristic gradient clipping techniques.

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 [40, 35, 34]. 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!

### 4.1 Practical Implementation

Exact optimization of the logistic Bellman 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 3 gives rise to Proximal Point Inverse Q-Learning (PPIQ-Learn), a model-free actor-critic IRL algorithm described in Algorithm 1. The key feature of PPIQ-Learn is that the policy evaluation step involves optimization of a single smooth and convex 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 $k$th iteration of PPIQ-Learn 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.

**Algorithm 1** Proximal Point Inverse Q-learning: PPIQ-Learn($\mathbf{\Phi}, \mathcal{D}_{\mathrm{E}}, K, \eta, \alpha$)

**Input:** Feature matrix $\mathbf{\Phi}$, expert demonstrations $\mathcal{D}_{\mathrm{E}}$, number of iterations $K$, step sizes $\eta$ and $\alpha$
**Input:** 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** $k = 1, \dots K$ **do**
  `// Critic-step (policy evaluation)`
  Initialize $\boldsymbol{\theta}_{k,0} = \mathbf{0}$ and $\mathbf{w}_{k,0} = \mathbf{0}$
  **for** $t = 0, \dots T-1$ **do**
    Run $\pi_{k-1}$ and collect i.i.d. samples $\{(s_{k-1}^{(n)}, a_{k-1}^{(n)}, s_{k-1}^{\prime(n)})\}_{n=1}^{n(t)+1}$
    Compute biased stochastic gradient estimators

$$(\widehat{\nabla}_{\mathbf{w}} \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k), \widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k)) = \mathrm{BSGE}(k, \mathbf{w}_t, \boldsymbol{\theta}_t, n(t))$$

    $\mathbf{w}_{k,t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_{k,t} + \beta_t \widehat{\nabla}_{\mathbf{w}} \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k))$
    $\boldsymbol{\theta}_{k,t+1} = \Pi_{\Theta}(\boldsymbol{\theta}_{k,t} + \beta_t \widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k))$
  **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 $\{\pi_k\}_{k \in [K]}$

The domain $\Theta$ in Algorithm 1 is the $\ell_\infty$-ball with appropriately chosen radius $D$. 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 4.** *It holds that* $\|\boldsymbol{\theta}_k^\star\|_\infty \leq M \frac{1+|\log \beta|}{1-\gamma} \triangleq D$*, for some constant* $M \geq 1$.

In order to estimate the gradient $\nabla_{\boldsymbol{\theta}} G_k(\mathbf{w}, \boldsymbol{\theta})$, we define the policy $\pi_{k,\boldsymbol{\theta}}(a|s) \propto \pi_k(a|s) e^{-\alpha Q_{\boldsymbol{\theta}}(s,a)}$, for all $k \in \mathbb{N}$, and for all $\boldsymbol{\theta} \in \mathbb{R}^m$. As justified in Appendix D, the following estimators of $\nabla_{\boldsymbol{\theta}} G_k(\mathbf{w}, \boldsymbol{\theta})$ and $\nabla_{\mathbf{w}} G_k(\mathbf{w}, \boldsymbol{\theta})$ are unbiased: Sample $(s', a') \sim \mathbf{d}_{k-1}$, $i' \sim \boldsymbol{\phi}(s', a')$, $s_0 \sim \boldsymbol{\nu}_0$, and $a_0 \sim \pi_{k-1,\boldsymbol{\theta}}(\cdot|s_0)$, then define

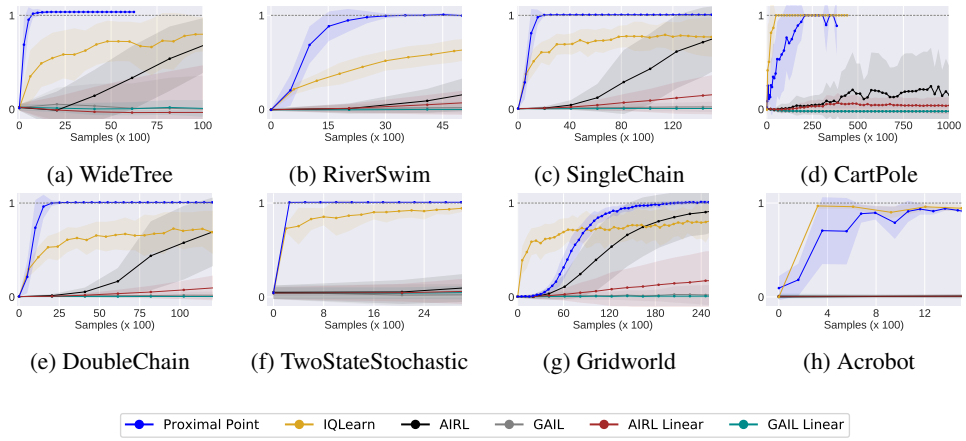
$$\widetilde{\nabla}_{\mathbf{w},j} \mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta}) = -\rho_{\boldsymbol{\phi}_j}(\widehat{\pi_{\mathrm{E}}}) + \mathbf{B}_{\mathbf{w},\boldsymbol{\theta}}^k(i') \mathbb{1}\{i' = j\} \tag{6}$$

$$\widetilde{\nabla}_{\boldsymbol{\theta},j} \mathcal{G}_k(\mathbf{w}, \boldsymbol{\theta}) = \mathbf{B}_{\mathbf{w},\boldsymbol{\theta}}^k(i') \Big[\gamma \mathbf{\Gamma}_k(i', j) - \mathbb{1}\{i' = j\}\Big] + (1-\gamma)\boldsymbol{\phi}_j(s_0, a_0). \tag{7}$$

where $\mathbf{B}_{\mathbf{w},\boldsymbol{\theta}}^k(i) \triangleq \frac{\exp\left(-\eta \boldsymbol{\delta}_{\mathbf{w},\boldsymbol{\theta}}^k(i)\right)}{Z_k}$, $Z_k \triangleq \sum_{i=1}^m \exp\big(-\eta \boldsymbol{\delta}_{\mathbf{w},\boldsymbol{\theta}}^k(i)\big)\rho_{\boldsymbol{\phi}_i}(\pi_{k-1})$, and $\mathbf{\Gamma}_k(i,j) \triangleq \sum_{s',a'} \mathbf{M}_{i,s'} \pi_{k-1,\boldsymbol{\theta}}(a'|s') \boldsymbol{\phi}_j(s', a')$. These expressions give rise to the Biased Stochastic Gradient Estimator subroutine (BSGE) given in Algorithm 4 in Appendix D, where we plug-in estimators $\widehat{\mathbf{B}}_{\mathbf{w},\boldsymbol{\theta}}^k \in \mathbb{R}^m$ and $\widehat{\mathbf{\Gamma}}_k \in \mathbb{R}^{m \times m}$ in Equations (6) and (7). In Appendix D, we discuss how to maintain good estimators $\widehat{\mathbf{B}}_{\mathbf{w},\boldsymbol{\theta}}^k$ and $\widehat{\mathbf{\Gamma}}_k \in \mathbb{R}^{m \times m}$ by using the linear MDP Assumption 1 and leveraging ridge regression. These estimators can be computed via simple linear algebra and their computational complexity is poly(m) and independent of the size of the state space.

## 5 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




(a) WideTree (b) RiverSwim (c) SingleChain (d) CartPole

(e) DoubleChain (f) TwoStateStochastic (g) Gridworld (h) Acrobot

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

ideal policy evaluation update $(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star)$ and the ideal policy update $\pi_k^\star$ are given by $(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) = \arg\max_{\mathbf{w},\boldsymbol{\theta}} \mathcal{G}_k^m(\mathbf{w}, \boldsymbol{\theta})$, 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 $\pi_k$ 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 $\varepsilon_k$. To this end, we need the following assumption.

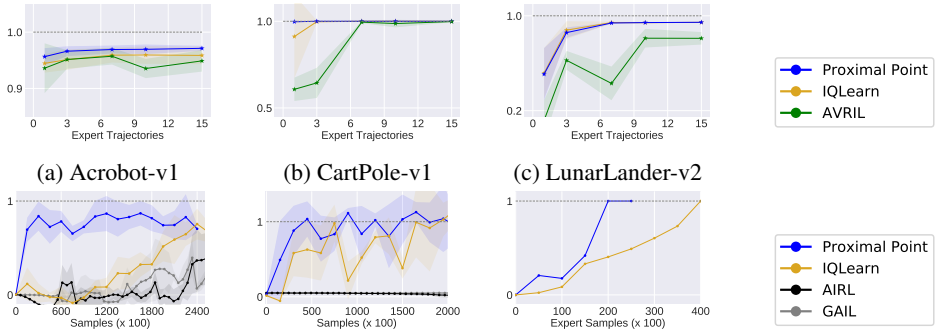
**Assumption 2.** *It holds that* $\lambda_{\min}(\mathbb{E}_{(s,a)\sim\mathbf{d}_k} \boldsymbol{\phi}(s,a)\boldsymbol{\phi}(s,a)^T) \geq \beta$, *for all* $k \in [K]$.

Assumption 2 states that every occupancy measure $\mathbf{d}_k$ induces a positive definite feature covariance matrix, and so every policy $\pi_k$ explores uniformly well in the feature space. This assumption is common in the RL theory literature [83–88]. It is also related to the condition of persistent excitation from the control literature [89]. In tabular MDPs, the assumption boils down to bounded state-action concentrability coefficients used in [12, 18], and so is restrictive, since, for example, is violated by deterministic policies. For the general non tabular case, the less restrictive Assumption 2 has been obtained by introducing the variable $\boldsymbol{\lambda} = \boldsymbol{\Phi}^\intercal \boldsymbol{\mu}$ instead of the variable $\boldsymbol{\mu}$ used in in [15, 12, 80]. 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* $C \triangleq \frac{1}{\beta\eta}\left(\sqrt{\frac{2\alpha}{1-\gamma}} + \sqrt{8\eta}\right) + \sqrt{\frac{18\alpha}{1-\gamma}}$. *Then, with probability at least* $1-\delta$, *it holds that* $d_{\mathcal{C}}(\widehat{\pi}, \pi_{\mathrm{E}}) \leq \frac{1}{K}\left(\frac{D(\boldsymbol{\lambda}^*||\boldsymbol{\Phi}^T\mathbf{d}_0)}{\eta} + \frac{H(\mathbf{d}^*||\mathbf{d}_0)}{\alpha} + C\sum_k \sqrt{\epsilon_k} + \sum_k \epsilon_k\right) + \varepsilon$.

By Theorem 1, whenever the policy evaluation errors $\varepsilon_k$, as well as the estimation error $\varepsilon$ can be kept small, Algorithm 1 ouputs a policy $\widehat{\pi}_K$ with small suboptimality gap $\rho_{\mathbf{c}_{\text{true}}}(\widehat{\pi}_K) - \rho_{\mathbf{c}_{\text{true}}}(\pi_{\mathrm{E}})$. 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 $\mathcal{O}(1/K)$. Moreover, it is worth noting that the provided error propagation analysis justifies using our derived actor-critic scheme with general function approximation. Indeed, by choosing $\boldsymbol{\Phi} = \mathbf{I}$, Assumption 1 is trivially satisfied and the $\boldsymbol{\theta}$ variable in the objective $\mathcal{G}_k$ is replaced by a $Q$-function. A scalable deep reinforcement learning implementation is then possible, without losing the theoretical guarantees.

In practice, the estimation error $\varepsilon$ can be made arbitrary small, by increasing the number of expert demonstrations $n_{\mathrm{E}}$. Moreover, the next theorem ensures that under Assumptions 1 and 2 the biased stochastic gradient ascent (BSGA) subroutine has sublinear convergence rate. Indeed, by choosing appropriately the learning rates and the number of sample transitions, the bias in the updates can be controlled and the BSGA subroutine outputs nearly optimal policy evaluation updates $(\mathbf{w}_k, \boldsymbol{\theta}_k)$.




(a) Acrobot-v1 (b) CartPole-v1 (c) LunarLander-v2

(d) HalfCheetah-v2 (e) Ant-v2 (f) Pong

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

**Theorem 2.** *Let* $(\mathbf{w}_k, \boldsymbol{\theta}_k)$ *be the output of the BSGA subroutine in Algorithm 4 for* $T$ *iterations, with* $n(t) \geq \max\left(\mathcal{O}\left(\frac{\gamma^2 mDt}{\beta}\log\frac{Tm}{\delta}\right), \mathcal{O}\left(\frac{mt}{\beta}\log\frac{Tm}{\delta}\right)\right)$ *sample transitions, and learning rates* $\beta_t = \mathcal{O}(\frac{1}{\sqrt{t}})$*. Then, with probability* $1-\delta$*, it holds that* $\epsilon_k = \mathcal{G}_k(\mathbf{w}_k^\star, \boldsymbol{\theta}_k^\star) - \mathcal{G}_k(\mathbf{w}_k, \boldsymbol{\theta}_k) \leq \mathcal{O}(\frac{\eta m D}{\beta\sqrt{T}})$.

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

**Offline Setting.** Finally, we notice that using $\boldsymbol{\Phi}^\intercal \boldsymbol{\mu}_{\pi_{\mathrm{E}}}$ as the reference distribution for the relative entropy we can obtain an offline algorithm that does not require environment interactions. By reinterpreting smoothing [44] 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 details are presented in Appendix G (see Theorem 4).

## 6 Experiments

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

**Online Setting.** We first present results in various tabular environments where we can implement our algorithm without any practical relaxation outperforming GAIL [4], AIRL [5] and IQ-Learn [32]. 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 [32] and AVRIL [90] that recently showed performances superior to other methods like [91][8]. We also tried our algorithm in the complex image-based `Pong` task from the Atari suite. Figure 2f 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` and `HalfCheetah` (see Figures 2e and 2d). The additional difficulty in implementing the algorithm in continuous control experiments is that the analytical form of the policy 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) [92] algorithm. SAC requires environment samples making the algorithm online. The good empirical result opens the question of analyzing policy improvement errors as in [93].

# References


[1] W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis)design for autonomous driving, 2021.

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

[3] Arthur Charpentier, Romuald Elie, and Carl Remlinger. Reinforcement learning in economics and finance, 2020.

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

[5] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2018.

[6] Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 313–329. Springer, 2020.

[7] Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

[8] Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations (ICLR)*, 2020.

[9] Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019.

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

[11] Qi Cai, Zhuoran Yang, Chi Jin, and Zhaoran Wang. Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*, pages 1283–1294. PMLR, 2020.

[12] Joan Bas-Serrano, Sebastian Curi, Andreas Krause, and Gergely Neu. Logistic Q-learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[13] Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems*, 33:17816–17826, 2020.

[14] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.

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

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

[17] R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

[18] Aldo Pacchiano, Jonathan Lee, Peter Bartlett, and Ofir Nachum. Near optimal policy optimization via REPS. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

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

[21] 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.

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

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

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

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

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

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

[28] 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.

[29] 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.

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

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

[32] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-learn: Inverse soft-Q learning for imitation. In *Advances in Neural Information Processing Systems*, 2021.

[33] Gabriel Kalweit, Maria Hügle, Moritz Werling, and Joschka Boedecker. Deep inverse Q-learning with constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[34] Zhihan Liu, Yufeng Zhang, Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Provably efficient generative adversarial imitation learning for online and offline setting with linear function approximation. *arXiv:2108.08765*, 2021.

[35] Lior Shani, Tom Zahavy, and Shie Mannor. Online apprenticeship learning. *arXiv:2102.06924*, 2021.

[36] 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.

[37] Jonathan Chang, Masatoshi Uehara, Dhruv Sreenivas, Rahul Kidambi, and Wen Sun. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34, 2021.

[38] Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325, 2020.

[39] Angeliki Kamoutsi, Goran Banjac, and John Lygeros. Efficient performance bounds for primal-dual reinforcement learning from demonstrations. In *International Conference on Machine Learning (ICML)*.

[40] 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.

[41] Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments, 2020.

[42] Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

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

[44] Yu. Nesterov. Smooth minimization of nonsmooth functions. *Math. Programming*, pages 127–152, 2005.

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

[46] Guy T de Ghellinck and Gary D Eppen. Linear programming solutions for separable markovian decision problems. *Management Science*, 13(5):371–394, 1967.

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

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

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

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

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

[52] 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.

[53] Paul J Schweitzer and Abraham Seidmann. Generalized polynomial approximations in markovian decision processes. *Journal of mathematical analysis and applications*, 110(2):568–582, 1985.

[54] Marek Petrik and Shlomo Zilberstein. Constraint relaxation in approximate linear programs. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 809–816, 2009.

[55] Marek Petrik, Gavin Taylor, Ron Parr, and Shlomo Zilberstein. Feature selection using regularization in approximate linear programs for markov decision processes. *arXiv preprint arXiv:1005.1860*, 2010.

[56] 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.

[57] 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.

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

[59] 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.

[60] Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations (ICLR)*, 2020.

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

[62] 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.

[63] 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.

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

[65] Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *National Conference on Artificial Intelligence (AAAI)*, 2010.

[66] Alexander Zimin and Gergely Neu. Online learning in episodic markovian decision processes by relative entropy policy search. *Advances in neural information processing systems*, 26, 2013.

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

[68] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

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

[70] 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.

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

[72] Carson Kent, Jiajin Li, Jose Blanchet, and Peter Glynn. Modified Frank Wolfe in probability space. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[73] Huang Xiao, Michael Herman, Joerg Wagner, Sebastian Ziesche, Jalal Etesami, and Thai Hong Linh. Wasserstein adversarial imitation learning. *arXiv:1906.08113*, 2019.

[74] Robert Dadashi, Leonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal Wasserstein imitation learning. In *International Conference on Learning Representations (ICLR)*, 2021.

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

[76] 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.

[77] 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.

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

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

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

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

[82] Fan Lu, Prashant G. Mehta, Sean P. Meyn, and Gergely Neu. Convex q-learning. In *2021 American Control Conference (ACC)*, pages 4749–4756, 2021.

[83] Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702. PMLR, 2019.

[84] Botao Hao, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Online sparse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 316–324. PMLR, 2021.

[85] Yaqi Duan, Zeyu Jia, and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning*, pages 2701–2709. PMLR, 2020.

[86] Nevena Lazic, Dong Yin, Mehrdad Farajtabar, Nir Levine, Dilan Gorur, Chris Harris, and Dale Schuurmans. A maximum-entropy approach to off-policy evaluation in average-reward mdps. *Advances in Neural Information Processing Systems*, 33:12461–12471, 2020.

[87] Yasin Abbasi-Yadkori, Nevena Lazic, Csaba Szepesvari, and Gellert Weisz. Exploration-enhanced politex. *arXiv preprint arXiv:1908.10479*, 2019.

[88] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in Markov decision processes. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 64–66. PMLR, 09–12 Jul 2020.

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

[90] Alex J. Chan and Mihaela van der Schaar. Scalable bayesian inverse reinforcement learning, 2021.

[91] Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Strictly batch imitation learning by energy-based distribution matching, 2021.

[92] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[93] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A Theory of Regularized Markov Decision Processes. In *International Conference on Machine Learning (ICML)*, 2019.

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

[95] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[96] Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 9.1–9.24, Edinburgh, Scotland, 25–27 Jun 2012. PMLR.

[97] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2), Apr 2021.

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

[99] Thomas Furmston and David Barber. Variational methods for reinforcement learning. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 241–248, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

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

[101] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

[102] J Andrew Bagnell and Jeff Schneider. Covariant policy search. 2003.

[103] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[104] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.

[105] Alborz Geramifard, Christoph Dann, Robert H Klein, William Dabney, and Jonathan P How. Rlpy: a value-function-based reinforcement learning framework for education and research. *J. Mach. Learn. Res.*, 16(1):1573–1578, 2015.

[106] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[107] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[108] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

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

# 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 5
   (b) Did you describe the limitations of your work? [Yes] We state the assumptions needed for the analysis in Sections 3 and 5
   (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 5
   (b) Did you include complete proofs of all theoretical results? [Yes] The main results are stated in Section 5 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 [32] 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.