Provable Benefits of Actor-Critic Methods for Offline Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Actor-critic methods are widely used in offline reinforcement learning practice but are understudied theoretically. In this work we show that the pessimism principle can be naturally incorporated into actor-critic formulations. We create an offline actor-critic algorithm for a linear MDP model more general than the low-rank model. The procedure is both minimax optimal and computationally tractable.

1 Introduction

Learning a near-optimal policy is a core reinforcement learning (RL) task. Oftentimes, we need to find a good policy using the available data and without the possibility of further interaction with the environment; this is called the *policy learning* problem in *offline* RL. Offline RL has unique challenges due to the incomplete information about the Markov decision process (MDP) encoded in the available dataset. For example, due to the maximization bias, a naive offline algorithm can settle for a policy with a dangerously high estimated value even if such value is highly uncertain. To avoid this phenomenon, researchers have introduced the idea of *pessimism* in offline RL [Liu et al., 2020, Jin et al., 2020b, Buckman et al., 2020, Kumar et al., 2019, Kidambi et al., 2020, Yu et al., 2020]. Additional literature is presented in Appendix A.

Pessimism prevents algorithms from settling down on uncertain policies whose value might be misleadingly high under the current dataset due to statistical errors. By using pessimism, uncertain policies are penalized and only those robust to statistical errors are returned. The principle can be implemented in two different ways: 1) by penalizing policies that are far from the one that generated the dataset or 2) by penalizing the value functions of policies not well covered by the dataset. We adopt the second view in this work.

Challenges Implementing pessimism with function approximation brings several challenges. First, *the uncertainty must be carefully estimated*: underestimating it may not lead to an effective algorithm and overestimating it leads to policies that are too conservative and thus underperform. Second, pessimism may *introduce complex, higher order perturbations* into the value function class handled by the algorithm (similar to adding optimistic bonuses in the exploration setting). Increasing the complexity of the function class often requires additional assumptions on the model, because the new class needs to interact 'nicely' with the Bellman operator. Prior art on pessimism with function approximation bypassed this problem by making strong model assumptions, like low rank transitions [Jin et al., 2020b] or algorithm-specific assumptions [Liu et al., 2020].

Actor-critic methods Moreover, most of these theoretically-justified algorithms are either model or value based [Liu et al., 2020, Jin et al., 2020b, Buckman et al., 2020, Kidambi et al., 2020, Yu et al., 2020], but in practice, actor-critic methods are widely used [Levine et al., 2020, Wu et al., 2019, Wu et al., 2021, Kumar et al., 2019, Kumar et al., 2020]. An

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

actor-critic method generally consists of an *actor* that changes the policy in order to maximize its value as estimated by the *critic*. In relation to value and model-based methods we ask:

Do actor-critic methods provably offer any advantage in offline RL?

We give a positive answer: by separating the policy optimization from the policy evaluation, both tasks become simpler to design and the pessimism principle can be incorporated more naturally.

Contributions We consider the linear function approximation setting and assume that a batch dataset $\mathcal{D}$ of states, actions, rewards and successor states is available. Using $\mathcal{D}$ we can construct the set $\mathcal{M}$ of statistically plausible MDPs subject to the linearity assumption.

Our objective is then to find the policy that performs the best in the face of uncertainty, namely the policy π with the highest minimum value function V_M^π across all plausible MDPs M in the set $\mathcal{M}$ of statistically plausible MDPs (e.g., [Mannor et al., 2012]):

$$\sup_\pi \inf_{M \in \mathcal{M}} V_M^\pi. \tag{1}$$

Actor-critic methods fit naturally in this framework: the *actor* solves the outer maximization problem over policies which are evaluated in the inner minimization problem by a pessimistic *critic*. This way, each algorithm solves a simple task: 1) the critic provides a pessimistic value function estimate for a fixed policy (the one currently examined by the actor) while 2) the actor ensures online learning-style guarantees with respect to a sequence of pessimistic MDPs implicitly identified by the critic. This is the first algorithmic idea and leads to a computationally tractable implementation.

The second algorithmic idea is to introduce pessimism without altering the prescribed function class. This is achieved by perturbing the value function (in the critic) within its prescribed functional space without adding pessimistic bonuses or absorbing states. This has two core advantages:

- There are no additional model assumptions compared to the vanilla (i.e., without pessimism) version of our actor-critic method; this is because the original value function class is not modified by the injection of pessimism.
- The algorithm operates on value functions with the original statistical complexity, enabling the construction of tight confidence intervals and ultimately minimax statistical rates.

2 Preliminaries and Assumptions

We consider an undiscounted finite-horizon MDP [Puterman, 1994] $M = (\mathcal{S}, \mathcal{A}, p, r, H)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, and horizon length $H \in \mathbb{N}^+$. For every $h \in [H] = \{1, \dots, H\}$, every state-action pair is characterized by an expected reward $r_h(s,a)$ with an associated reward distribution $R_h(s,a)$ and a transition kernel $p_h(\cdot \mid s,a)$ over next state. For any $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$, the state-action value function of a non-stationary policy $\pi = (\pi_1, \dots, \pi_H)$ is defined as $Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}_{s_l \sim \pi|(s,a)} \sum_{l=h+1}^H r_l(s_l, \pi_l(s_l))$, where the expectation is over the trajectories induced by π upon starting from (s,a). When we omit the starting state-action (s,a), the expectation is intended to start from a fixed state denoted by s_1. The value function associated to π is $V_h^\pi(s) = Q_h^\pi(s, \pi_h(s))$. Under some regularity conditions, e.g., [Shreve and Bertsekas, 1978], there always exists an optimal policy $\pi^\star$ whose value and action-value functions are defined as $V_h^\star(s) = V_h^{\pi^\star}(s) = \sup_\pi V_h^\pi(s)$ and $Q_h^\star(s,a) = Q_h^{\pi^\star}(s,a) = \sup_\pi Q_h^\pi(s,a)$. We define the Bellman evaluation operator

$$\mathcal{T}_h^\pi(Q_{h+1})(s,a) = r_h(s,a) + \mathbb{E}_{s' \sim p_h(s,a)} \mathbb{E}_{a' \sim \pi} Q_{h+1}(s', a').$$

We let $\mathcal{B}_d(r) = \{x \in \mathbb{R}^d \mid \|x\|_2 \le r\}$ denote the Euclidean ball of radius $r \in \mathbb{R}$ in dimension d; sometime we simply write $\mathcal{B}$ when there is no possibility of confusion. We use the $\widetilde{O}$ notation to suppress log factors in the input parameters $(\frac{1}{\delta}, d, H, \lambda)$, and the O and Ω notation to ignore constants in the upper and lower bound. The notation $\lesssim$ means $\le$ up to a constant while $\lessapprox, \approx, \gtrapprox, \asymp$ are used to highlight dominant terms in the proof sketch without rigorous mathematical definitions. For a vector $x \in \mathbb{R}^d$ we let $[x]_i$ denote its i component.

2.1 Assumptions on Data Generation

The algorithm operates on a dataset $\mathcal{D} = \{(s_{hk}, a_{hk}, r_{hk}, s^+_{hk})\}_{h=1,2,\ldots,H}^{k=1,2,\ldots,K}$ of state-action-reward-next states generated by the underlying MDP, possibly in an adaptive fashion.

Assumption 1 (Data Generation)**.** *Assume that for every (s, a) the reward random variable with distribution $R_h(s, a)$ is 1-subgaussian. The dataset $\mathcal{D}$ is such that*

$$r_{hk} \sim R(s_{hk}, a_{hk}), \qquad s^+_{hk} \sim p_h(s_{hk}, a_{hk}) \tag{2}$$

where each s_{hk}, a_{hk} is allowed to depend on the previously sampled $(s_{ij}, a_{ij}, r_{ij}, s^+_{ij})$.

This allows considerable freedom: 1) the dataset may be generated from (mixture) policies or by another mechanism that collects information at different state-actions; 2) the dataset may be generated by an adversarial procedure that changes the data acquisition strategy as feedback is received.

2.2 Policy and Value Function Class

Next, we define the policy space Π and the action value function space $\mathcal{Q}$ where we seek solutions. For a fixed timestep h (which we omit here for brevity), consider a fixed feature extractor $\phi : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$, $\|\phi(\cdot, \cdot)\|_2 \leq 1$ and two radii, $r_w \in (0, 1],\ r_\theta > 0$ for the value function parameter w and for the policy parameter θ.

Definition 1 (Functional Spaces)**.**

$$\mathcal{Q}(r_w) = \{(s, a) \mapsto \phi(s, a)^\top w \mid \|w\|_2 \leq r_w\}, \qquad \Pi(r_\theta) \stackrel{def}{=} \left\{ \frac{\exp[\phi(s, a)^\top \theta]}{\sum_{a'} \exp[\phi(s, a')^\top \theta]} \mid \|\theta\|_2 \leq r_\theta \right\}.$$

The policy radius can be large $r_\theta \gg 1$ but we constrain $r_w \leq 1$ so that $\sup_{(s,a,w)} |Q_w(s, a)| \leq 1$. In finite horizon problems one can select different feature extractors ϕ_h in every step h; this generates H functional spaces $\mathcal{Q}_1, \ldots, \mathcal{Q}_H$ and $\Pi_1, \ldots, \Pi_H$. We drop the dependence on the radii when referring to the functional spaces and implicitly assume that the terminal value function is zero.

2.3 Assumptions on Function Class

If we seek to find the policy $\pi \in \Pi$ with the highest value function, it seems reasonable to require that the following representation condition (approximately) holds. We assume a common feature extractor $\phi : \mathcal{S} \times \mathcal{A},\ \|\phi(\cdot, \cdot)\|_2 \leq 1$ throughout this section.

Assumption 2 (Linear Q^π)**.** *We say the MDP admits a linear action-value function representation for all policies in Π if Q^π is linear, i.e.,*

$$\forall \pi \in \Pi, h \in [H], \exists w^\pi_h \text{ such that } Q^\pi_h(s, a) = \phi_h(s, a)^\top w^\pi_h. \tag{3}$$

Unfortunately, Corollary 1 in [Zanette, 2020] or Theorem 4.1 in [Wang et al., 2020a] establish that even under such assumption, we might need exponentially many samples to do better than a random policy. This suggests we need even stronger conditions. One such condition is the assumption we make in this work, which allows a classical temporal-difference critic to evaluate the policies in Π.

Assumption 3 (Restricted Closedness)**.** *The policy and value function spaces $(\Pi, \mathcal{Q})$ are closed up to $\epsilon^{miss} \in \mathbb{R}$ error in ∞ norm with respect to a finite horizon MDP if*

$$\forall h \in [H] : \qquad \sup_{\substack{Q_{h+1} \in \mathcal{Q}_{h+1} \\ \pi_{h+1} \in \Pi_{h+1}}} \inf_{Q_h \in \mathcal{Q}_h} \|Q_h - \mathcal{T}_h^{\pi_{h+1}} Q_{h+1}\|_\infty \leq \epsilon^{miss}_h \in \mathbb{R}. \tag{4}$$

The restricted closedness assumption measures how well we can fit the action-value function resulting from the application of the Bellman evaluation operator to an action value function in $\mathcal{Q}$ and for a policy in Π. It enables the analysis of the classical *Least Square Policy Evaluation* (LSPE) [Nedić and Bertsekas, 2003], which will be our starting point when constructing the critic.

A related model assumption is the *low-rank* or *linear* MDP model [Jin et al., 2020a, Yang and Wang, 2020] used by the state of the art for offline RL with pessimismistic guarantees [Jin et al., 2020b] and much of the online RL literature [Agarwal et al., 2020a, Modi et al., 2021, Zanette et al., 2020a].

Assumption 4 (Low-Rank MDP)**.** *We say that an MDP is low-rank if* $\forall h \in [H]$ *there exist a reward parameter* $w_h^R \in \mathbb{R}^{d_h}$ *and a component-wise positive vector function* $\psi_h(\cdot) : \mathcal{S} \mapsto \mathbb{R}^d$ *such that* $\|[\psi_h(\cdot)]_i\|_1 = 1,\ \forall i \in [d]$ *and*

$$r_h(s,a) = \phi_h(s,a)^\top w_h^R, \qquad p_h(s' \mid s,a) = \phi_h(s,a)^\top \psi_h(s'), \qquad \forall (s,a,h,s'). \tag{5}$$

We clarify the relation between these assumptions in the following proposition where we assume that the value function parameter $w \in \mathbb{R}^d$ for simplicity; the proof is deferred to Appendix B *(Assumptions)*.

Proposition 1 (Low Rank $\subset$ Restricted Closedness $\subset$ Linear Q^π)**.** *Fix a state-action space, an horizon, and a feature extractor.*

1. *The class of low-rank MDPs is a strict subset of the class of MDPs such that restricted closedness holds*
2. *The class of MDPs such that restricted closedness holds is a subset of the MDP class where the linear* Q^π *assumption holds.*

Furthermore, all inclusions are strict.

Operating under assumptions stronger than linear Q^π enables polynomial sample complexity. Our algorithm can successfully operate in the low-rank framework as a special case.

3 Intuition and Algorithmic Choices

We next describe in detail the actor in Algorithm 1 and the critic in Algorithm 2 .

Algorithm 1 ACTOR (MIRROR DESCENT)

Input: Dataset $\mathcal{D}$, starting state s_1
Set $\theta_1 = (\vec{0}, \dots, \vec{0})$
for $k = 1, 2, \dots, K$ **do**
  $\underline{w}_k \leftarrow$ CRITIC$(\mathcal{D}, \pi_{\theta_k}, s_1)$
  $\theta_{k+1} = \theta_k + \eta \underline{w}_k$
end for
Mixture policy $\pi_{\theta_1}, \dots, \pi_{\theta_K}$

Algorithm 2 CRITIC (PLSPE)

Input: Dataset $\mathcal{D}$, target policy π, starting state s_1
Solve the optimization program (6)
Return $\underline{w}$

The Critic: Pessimistic Least Square Policy Evaluation The purpose of the critic is to provide pessimistic value function estimates corresponding to the policy π under consideration by the actor. Monte Carlo with importance sampling is not appropriate here, as the policy or distribution that generated the dataset might be unknown. This suggests we use a temporal difference method like LSPE, perturbed to return *pessimistic* value function estimates (Algorithm 2); we name this PLSPE.

The key algorithmic innovation is that the critic does not use additive bonuses to the value function [Jin et al., 2020b] or absorbing states [Liu et al., 2020] to achieve pessimism; instead, it introduces pessimistic perturbations $\xi = (\xi_1, \dots, \xi_H)$ directly to the regressed parameter w, see the program in Eq. (6). This way, *the action value function remains linear*. These perturbations try to compensate the statistical error of the parameter computed by linear regression. The critic is free to adjust the variable ξ so as to minimizes the value function estimate $\sum_a \pi(a \mid s_1)\phi_1(s_1,a)^\top w_1 = \mathbb{E}_{a' \sim \pi}\phi(s_1,a')^\top w_1$ for the policy π at the initial state s_1; precisely it minimizes

$$(\xi^\pi, \underline{w}^\pi) \stackrel{def}{=} \arg\min_{\xi, w} \underbrace{\sum_a \pi_1(a \mid s_1)\phi_1(s_1,a)^\top w_1}_{\textit{Value function estimate for } V_1^\pi(s_1)}$$

$$\text{s.t.} \qquad \forall h \in [H]: \qquad w_h = \xi_h + \Sigma_h^{-1} \sum_{k=1}^K \phi_{hk}\Big[r_{hk} + \sum_a \pi_{h+1}(a \mid s_{h+1,k})\phi_{h+1}(s_{h+1,k},a)^\top w_{h+1}\Big]$$

$$\|\xi_h\|_{\Sigma_h}^2 \le \alpha_h, \qquad \|w_h\|_2^2 \le r_h, \qquad w_{H+1} = 0. \tag{6}$$

For brevity, we used $\phi_{hk} = \phi_h(s_{hk}, a_{hk})$ and have defined the cumulative covariance matrix

$$\Sigma_h \stackrel{def}{=} \sum_{h=1}^{H} \phi_{hk}\phi_{hk}^\top + \lambda I. \tag{7}$$

The regularization $\lambda > 0$, the radii $r_h \in (0, 1]$ and the perturbation sizes α_h are parameters chosen by the user. The recommended value for α is defined in the main result and its role is clarified in the analysis. The evaluation policy π is chosen by the actor.

The optimization program (6) consists of a linear objective subject to quadratic constraints; it belongs to the class of convex quadratic programs subject to quadratic constraints, a special case of second order cone programs, and can be efficiently solved with mainstream numerical libraries.

The Actor: Mirror Descent As actor, we use Mirror Descent with Kullback Leibler (KL) divergence as regularizer which leads to an exponentiated gradient update rule [Bubeck, 2014]:

$$\pi_{k+1}(a \mid s) \propto \pi_k(a \mid s)e^{\eta' Q(s,a)} \qquad \text{for each } (s, a) \in \mathcal{S} \times \mathcal{A};\ \eta' \in \mathbb{R}. \tag{8}$$

If the Q-value above from the critic lives in $\mathcal{Q}$, then it is possible to show that $\pi_{k+1} \in \Pi$ and the update rule takes a much simpler and computationally more efficient form (cf. Line 5 of Algorithm 1). In this case, the spaces $(\mathcal{Q}, \Pi)$ are said to be *compatible* [Sutton et al., 1999, Kakade, 2001, Agarwal et al., 2020b, Raskutti and Mukherjee, 2015] and the resulting algorithm is often called the *Natural Policy Gradient* (NPG) (see also [Geist et al., 2019, Shani et al., 2020]).

Since by construction the critic maintains a linear action value function even after pessimistic perturbations, the actor policy space is the simple softmax policy class Π and the easier update rule can be used. As we explain in the analysis, this has important statistical benefits.

The NPG algorithm that we use here readily achieves online regret rates (in the optimization setting with exact feedback) $\sim 1/k$ or $\sim 1/\sqrt{k}$ depending on the analysis [Agarwal et al., 2020b] and the learning rate, although we mention that these rates could potentially be improved [Khodadadian et al., 2021, Lan, 2021, Bhandari and Russo, 2020].

4 Main Result

Let us introduce the optimization error $\mathcal{R}(K)$, function of the number of actor's iterations K, and the uncertainty function $U(\pi)$ for a policy π where $\sqrt{\alpha_h} = \widetilde{O}(\sqrt{d_h + d_{h+1}}) + \epsilon_h^{miss}\sqrt{K} + \sqrt{\lambda}$ is fully defined in Lemma 7 and Definition 6 in appendix:

$$U(\pi) \stackrel{def}{=} 2\sum_{h=1}^{H}\left[\epsilon_h^{miss} + \sqrt{\alpha_h}\|\mathbb{E}_{(s_h,a_h)\sim\pi}\phi(s_h, a_h)\|_{\Sigma_h^{-1}}\right], \qquad \mathcal{R}(K) \stackrel{def}{=} 4H\sqrt{\frac{\ln|\mathcal{A}|}{K}}. \tag{9}$$

The amount of information from the dataset $\mathcal{D}$ is fully encoded in the uncertainty function U through the cumulative covariance matrix Σ_h. The more data are available, the more positive definite Σ_h is and the smaller the uncertainty function $U(\pi)$ becomes for a fixed policy π. If the sampling distribution is fixed, then $U(\pi) \lessapprox C/\sqrt{n}$ where C can be interpreted as the condition number of Σ_h^{-1} and n is the number of samples.

Our main result holds under Assumption 1 *(Data Generation)*, when the learning rate is $\eta = \sqrt{\ln|\mathcal{A}|/K}$, the radii for the action value function[1] parameters are in $(0, 1]$, the regularization is $\lambda \geq 1$ and the number of iterations is $K \geq \ln|\mathcal{A}|$; Π_{all} is the class of all stochastic policies.

Theorem 1 (Main Result)**.** *Algorithms 1 and 2 return a policy π_{ALG} such that*

$$\mathbb{P}\Big(V_1^{\pi_{\text{ALG}}}(s_1) \geq \sup_{\pi\in\Pi_{all}}\Big[V_1^{\pi}(s_1) - U(\pi)\Big] - \mathcal{R}(K)\Big) \geq 1 - \delta.$$

The result provides a lower bound on the quality of the returned policy and highlights a tradeoff between the suboptimality of the comparator π and its uncertainty $U(\pi)$. Note that the optimization

[1] This represents a setting where both the reward and the value function can be as large as 1 in absolute value. One easily recovers the setting with value functions in $[0, H]$ using a rescaling argument.

error $\mathcal{R}(K)$ goes to zero as $K \to \infty$; different choices of the learning rate are possible and they only affect the optimization error $\mathcal{R}(K)$ (i.e, the computational cost). Thus, ignoring the optimization error, regularization and misspecification and assuming $d_h = d, \forall h \in [H]$ we obtain with high probability $V_1^{\pi_{\text{ALG}}}(s_1) \gtrapprox \sup_{\pi \in \Pi_{all}} V_1^{\pi}(s_1) - \sqrt{d} \sum_{h=1}^{H} \|\mathbb{E}_{(s_h,a_h)\sim\pi_h} \phi(s_h, a_h)\|_{\Sigma_h^{-1}}$.

The result is complemented by a matching worst-case upper bound on the quality of the returned policy, excluding constants and log factors. The upper bound already arises in a setting that is easier for the learner, as it holds (1) when the MDP is *low-rank* (thus it applies when Assumption 3 *(Restricted Closedness)* holds), and (2) when the mechanism that generates the dataset is *non-adaptive* (thus it applies when Assumption 1 *(Data Generation)* holds).

We assume $d_h = d, \forall h \in [H]$ and $\epsilon_h^{miss} = 0$ for simplicity, as well as $\lambda = 1$ when referring to the uncertainty function U; $\mathbb{E}_M$ indicates that the expectation is with respect to MDP M.

Theorem 2 (Information-Theoretic Upper Bound)**.** *Fix any choice of horizon H, of dimension d and of number of samples n collected at each timestep. There exists an MDP class $\mathcal{M}$ such that*

$$\sup_{\text{ALG}} \inf_{M \in \mathcal{M}} \mathbb{E}_M V_{1M}^{\pi_{\text{ALG}}} \leq \left[V_{1M}^{\star}(s_1) - \Omega(1) \times dH \sqrt{\frac{d}{n}} \right] \leq \sup_{\pi \in \Pi_{all}} \left[V_{1M}^{\pi}(s_1) - \frac{\Omega(1)}{\log\left(\frac{1}{\delta}, K\right)} \times U(\pi) \right].$$

Comparison with literature Theorem 1 automatically implies the typical bound $\mathbb{P}[V_1^{\pi_{\text{ALG}}}(s_1) \geq V_1^{\star}(s_1) - U(\pi^{\star})] \geq 1 - \delta$ when the comparator policy is the optimal policy $\pi^{\star}$, e.g., [Jin et al., 2020b, Rashidinejad et al., 2021, Kidambi et al., 2020, Kumar et al., 2019, Buckman et al., 2020]. The guarantee can be written as $V_1^{\pi_{\text{ALG}}}(s_1) \gtrapprox V_1^{\star}(s_1) - C/\sqrt{n}$ where n is the number of samples and C is the (scaled) condition number of Σ_h^{-1}. One could interpret C as a concentrability coefficient that expresses the coverage of dataset — through Σ_h — with respect to the average direction in feature space $\mathbb{E}_{(s_h,a_h)\sim\pi_h^{\star}} \phi(s_h, a_h)$ of the optimal policy $\pi^{\star}$. As in [Jin et al., 2020b], such factor C can be small even when traditional concentrability coefficients are large because they depend on state-action visit ratios (see the literature in Appendix A, e.g., [Chen and Jiang, 2019]).

Even ignoring the concentrability coefficient, the form of our result is significantly stronger as our algorithm competes with all comparator policies simultaneously; these policies are not necessarily in the prescribed policy class Π. To highlight the strength of our formulation (see also [Yu et al., 2020, Liu et al., 2020] for results in a similar form), suppose that the optimal policy is not well covered, i.e., $U(\pi^{\star})$ infinite, but there exists a near-optimal policy π^{+} i.e., such that $V_1^{\pi^{+}}(s_1) \geq V_1^{\star}(s_1) - \epsilon$ for some small ϵ, which is well covered by the dataset, i.e., $U(\pi^{+}) \approx 0$. In this case, Theorem 1 ensures with high probability $V_1^{\text{ALG}}(s_1) \gtrapprox V_1^{\star}(s_1) - \epsilon$. In contrast, traditional analyses that use only $\pi^{\star}$ as comparator cannot return meaningful guarantees.

The work closest to ours is [Jin et al., 2020b]; our work directly improves on theirs by closing the dH gap between their upper and lower bound while working under the more permissive Assumption 3 *(Restricted Closedness)* which includes low-rank MDPs. A $\sqrt{d}$-improvement is due to the algorithm we use and the remaining is due to a more refined analysis and construction to certify optimality in Theorem 2 (notice that our upper and lower bounds differ from theirs by a factor of H due to a different normalization in the value function). The result of [Liu et al., 2020] can be specialized to the low-rank MDP setting but would give a suboptimal bound while additionally requiring density estimates.

Deriving a computationally tractable model-free algorithm without low-rank dynamics but subject to value function perturbations (e.g., optimistic or pessimistic perturbations) is an open problem even in the more heavily studied exploration setting: there the current state of the art [Zanette et al., 2020b, Jin et al., 2021, Du et al., 2021, Jiang et al., 2017] only present computationally *intractable* algorithms with the exception of [Zanette et al., 2020c] for a PAC setting with low inherent Bellman error which however requires an additional 'explorability' condition.

5 Proof Sketch

Our analysis has three main ingredients: the online learning guarantees of the actor, the pessimistic estimates from the critic, and the concept of induced MDP that connects the actor to the critic. The proof sketch follows a bottom-up approach: (1) starting with the critic, we explain the benefits of

working within the prescribed value function space, (2) we introduce the concept of MDPs induced by the actor's policy to interpret the pessimistic value function given by the critic as an exact value function on an adversarially chosen MDP and (3) we conclude by giving online-style learning guarantees for the actor.

The proof sketch is specialized to the well specified case ($\epsilon^{miss} = 0$), it ignores the bias due to regularization ($\lambda = 0$) and it assumes we are working in a high probability 'good event' to simplify the analysis.

Additional notation We denote with $\pi_k = \pi_{\theta_k}$ the policy of the actor in iteration k; we use M_k to denote the corresponding induced MDP (the definition will be given later). When the critic is invoked with an actor's policy $\pi \in \{\pi_1, \dots, \pi_K\}$, it solves the numerical optimization program (6) yielding the solution $(\underline{\xi}^\pi, \underline{w}^\pi)$ that determines the (action) value functions

$$\underline{Q}_h^\pi(s,a) : (s,a) \mapsto \phi(s,a)^\top \underline{w}_h^\pi, \qquad \underline{V}_h^\pi(s) : (s) \mapsto \mathbb{E}_{a' \sim \pi_h(\cdot|s)} \underline{Q}_h^\pi(s,a'). \tag{10}$$

We let e_h^π denote the statistical error in parameter space

$$e_h^\pi \stackrel{def}{=} \Sigma_h^{-1} \sum_{k=1}^{K} \phi_{hk} \Big[r_{hk} + \underline{V}_{h+1}^\pi(s_{h+1,k}, a') - (\mathcal{T}_h^\pi \underline{Q}_{h+1}^\pi)(s_{hk}, a_{hk}) \Big], \qquad \forall h \in [H]. \tag{11}$$

Using the policy class Π_h and action value function $\mathcal{Q}_h$ from Definition 1 *(Functional Spaces)*, we define the induced value function class

$$\mathcal{V}_h \stackrel{def}{=} \big\{ s \mapsto \mathbb{E}_{a' \sim \pi_h(\cdot|s)} Q(s,a') \mid Q \in \mathcal{Q}, \pi \in \Pi \big\}. \tag{12}$$

5.1 Critic's Pessimistic Guarantees

We start by analyzing the value function error.

Lemma 1 (PLSPE Errors)**.** *For any input policy $\pi \in \Pi$, Algorithm 2 ensures that*

$$\underbrace{(\underline{V}_1^\pi - V_1^\pi)(s_1)}_{\textit{Error at the initial state}} = \sum_{h=1}^{H} \mathbb{E}_{(s_h,a_h)\sim\pi} \overbrace{\phi(s_h,a_h)^\top \Big(\underbrace{\underline{\xi}_h^\pi}_{\textit{Pess.Perturb.}} + \underbrace{e_h^\pi}_{\textit{Stat.Err.}} \Big)}^{\stackrel{def}{=} b_h^\pi(s,a) \quad \textit{(Total Perturb.)}}. \tag{13}$$

The complete statement and its proof can be found in Lemma 5 in appendix. The proof of this lemma exploits the least-squares constraint (6) and Assumption 3 *(Restricted Closedness)*. We highlight that such assumption already arises in the analysis of standard LSPE when $\underline{\xi}_h^\pi = 0$. Allowing $\underline{\xi}_h^\pi \neq 0$ is equivalent to experiencing a different noise perturbation, and therefore no further model assumption is needed. This is a benefit of working in parameter space in terms of modeling assumptions.

Ideally we would set $\underline{\xi}_h^\pi = -e_h^\pi$ in the program (6) leading to no value function error $\underline{V}_1^\pi = V_1^\pi$. Since e_h^π is unknown, the program finds a pessimistic value by allowing $\|\underline{\xi}_h^\pi\|_{\Sigma_h}$ to be of similar size as the noise error $\|e_h^\pi\|_{\Sigma_h}$, which is bounded below. Let $\log \mathcal{N}(\mathcal{V}_{\text{ALG},h+1})$ be the log covering number of the value function class for $\underline{V}_{h+1}^\pi$.

Lemma 2 (Statistical Error)**.** *Under Assumption 1, with probability $1-\delta$*

$$\|e_h^\pi\|_{\Sigma_h} \leq \sqrt{\alpha_h} \stackrel{def}{=} \widetilde{O}\left(\sqrt{d_h + \ln \mathcal{N}(\mathcal{V}_{\text{ALG},h+1}) + \ln \frac{1}{\delta}} \right) \qquad \forall \pi_h \in \Pi_h, \forall h \in [H].$$

As the feasible set for $\underline{\xi}_h^\pi$ in the program in Eq. (6) now contains $\xi_t^\pi = -e_t^\pi$ as feasible value, and this produces no value function error, it must follow that $\underline{V}_1^\pi(s_1) \leq V_1^\pi(s_1)$. Thus, we have informally shown:

Proposition 2 (Critic's Pessimistic Guarantees)**.** *With high probability we have the upper bound $\underline{V}_1^\pi(s_1) \leq V_1^\pi(s_1)$.*

It remains to compute the value of $\sqrt{\alpha}$ by computing the log covering number for the action value function class $\mathcal{V}_{\text{ALG},h+1}$ used by the algorithm. Thankfully, our choice of making perturbations

within the function class ensures that $\mathcal{V}_{\text{ALG},h+1} = \mathcal{V}_{h+1}$ (defined in Eq. (12)); this function class is relatively simple as reflected by its small covering number. This is where we strongly benefit from working in parameter space in terms of statistical efficiency. In order to cover the value function (10), it sufficies to cover the action value function class $\mathcal{Q}$ and the policy class Π.

First, for a fixed policy π, since the agent's action value function is enforced to be linear $\underline{Q}^\pi \in \mathcal{Q}$ even after perturbations, the union bound only needs to be done over the linear class $\mathcal{Q}$; we thus avoid a potentially more costly union bound over a much larger function class $\mathcal{Q}' =$ [linear functions] + [complex bonus], e.g. [Jin et al., 2020b].

Second, the union bound needs to be extended to all policies that the actor can use to invoke the critic. Since the action value function $\underline{Q}$ that the critic identifies is linear, the update takes a *simple closed form expression* because the softmax policy class Π and the linear action value function class $\mathcal{Q}$ are *compatible* [Kakade, 2001, Agarwal et al., 2020b]. Precisely, this gives the simple update rule in Line 5 of Algorithm 1. If the action value function $\underline{Q}$ was perturbed by bonuses, linearity of the critic's value function would be lost and its functional space $\neq \mathcal{Q}$ would not be compatible with the policy space Π. The more general form of the exponential gradient update in Eq. (8) would need to be used. However, the resulting policy would no longer live in Π and instead would generate a more complex (i.e., with a larger covering number) policy class for the actor. The space complexity would increase too: our update (cf. Line 5 in Algorithm 1) requires a simple vector addition while the exponential update rule in Eq. (8) should be performed in all (s, a) where the policy is needed.

A simple computation now gives $\log \mathcal{N}(\mathcal{V}_{h+1}) \asymp d$ and thus $\sqrt{\alpha} = \widetilde{O}(\sqrt{d})$: this means that our union bounds are small enough that the resulting confidence intervals for the error e_h^π are about the same size as those arising from linear bandit regression where no union bound is needed. Ultimately, this is where we can save a $\sqrt{d}$ factor compared to [Jin et al., 2020b].

5.2 Induced MDP

The critic identifies *pessimistically biased* $\underline{Q}^{\pi_k}$ values for each actor's policy π_k. Such Q values could be interpreted as being the *exact* Q values on a perturbed MDP M_k. This connection is useful in relating the bias from the critic to the online learning guarantees from the actor.

For every policy π that the critic receives, define the MDP M^π induced by π as the original MDP M with an additional reward function b^π

$$M^\pi = \langle \mathcal{S}, \mathcal{A}, p, r + b^\pi, H \rangle \quad \text{where} \quad b_h^\pi(s,a) \stackrel{def}{=} \phi(s,a)^\top \left(e_h^\pi + \underline{\xi}_h^\pi \right), \qquad \forall (s,a,h). \tag{14}$$

Every actor's policy π_k identifies an induced MDP M_k whose additional (compared to M) reward function b^π is the pessimistic perturbation induced by the critic + statistical errors. These MDPs are *adversarial* as they minimize the value of the actor's policies. We introduce them here because they allow us to *evaluate* arbitrary policies in the critic's pessimistic world.

Proposition 3 (Value of Policies on Induced MDP)**.** *For any actor's policy π_k, we have*

$$Q_{M_k}^{\pi_k} = \underline{Q}^{\pi_k} \quad \textit{and} \quad V_{M_k}^{\pi_k} = \underline{V}^{\pi_k}. \tag{15}$$

I.e., the critic's pessimistic $\underline{Q}^{\pi_k}$ function equals the action value function $Q_{M_k}^{\pi_k}$ in the induced MDP M_k. Furthermore, we have the following guarantees on the values of policies on M_k

$$V_{1,M_k}^{\pi_k}(s_1) \le V_1^{\pi_k}(s_1) \qquad \textit{for the actor's policy } \pi_k\textit{, and} \tag{16a}$$

$$V_{1,M_k}^{\pi}(s_1) \le V_1^{\pi}(s_1) + U(\pi) \qquad \textit{for any } \pi \tag{16b}$$

The statements above follow directly from the definition of induced MDP through the additional reward function b_h and can be found in Appendix F. The proposition states three important facts: 1) the policy π_k that induces the MDP M_k is 'special', as its Q function on M_k equals the critic's $\underline{Q}^{\pi_k}$ function (Eq. (15)); 2) as a result, the value of π_k on M_k is pessimistic compared to its value on the original MDP (first equation in Eq. (16)); 3) the value of any other policy on M_k and M differs by at most the uncertainty function $U(\cdot)$ (second equation in Eq. (16)).

We highlight that the induced MDP M_k is unknown to the learner, because constructing M_k requires knowledge of the original reward and transition function in addition to the error function $e_h^{\pi_k}$; the induced MDP is used only in the analysis.

5.3 Actor's Analysis

In this section, we present the actor's guarantees on the sequence of adversarial MDPs M_k identified by the critic. In order to do so, we modify an analysis of the natural policy gradient algorithm [Agarwal et al., 2020b] to derive online learning-style guarantees. The proposition below holds under additional preconditions that are satisfied during the execution of the actor; see Proposition 7 in the appendix for details.

Proposition 4 (Actor's Analysis)**.** *Suppose that the actor receives a parameter $\underline{w}_k$ corresponding to the action value function of π_k on M_k, i.e.,*

$$Q^{\pi_k}_{h,M_k}(s,a) \stackrel{def}{=} \phi_h(s,a)^\top \underline{w}_{hk}, \qquad \forall (s,a,h,k). \tag{17}$$

Then with respect to any fixed comparator policy π (not necessarily in Π), the sequence of actor policies $\{\pi_k\}$ satisfies the bound

$$\frac{1}{K}\sum_{k=1}^{K}(V^{\pi}_{1,M_k} - V^{\pi_k}_{1,M_k})(s_1) \le \mathcal{R}(K) \lesssim H\sqrt{\frac{\ln|\mathcal{A}|}{K}}. \tag{18}$$

The result ensures convergence to the best comparator policy for any sequence of potentially adversarially chosen MDPs $\{M_k\}$. The lemma assumes through Eq. (17) that the actor receives from the critic the exact action value function $Q^{\pi_k}_{M_k}$ (compactly represented by the parameter $\underline{w}_k$) for each policy π_k on M_k. This is indeed the case: the critic returns the parameter $\underline{w}_k$ such that

$$Q^{\pi_k}_{h,M_k}(s,a) \stackrel{(1)}{=} \underline{Q}^{\pi_k}_h(s,a) \stackrel{(2)}{=} \phi_h(s,a)^\top \underline{w}_{hk} \tag{19}$$

where (2) holds by definition, Eq. (10) and (1) holds by the property of the induced MDP which preserves the $\underline{Q}$ values from the critic for the policy it is induced by, see Eq. (15).

5.4 Combining the Ingredients

Now we have all the ingredients to prove Theorem 1. The following reasoning holds under a high probability event and for any comparator π, not necessarily in Π.

Recall the actor's final policy π_{ALG} is a mixture policy among $\pi_1, \ldots, \pi_k$, which justifies step (1) below. Using both equations in Eq. (16), we can evaluate the average suboptimality gap in the original MDP M as suboptimality gap in the induced MDP sequence $\{M_k\}$ plus the uncertainty of the comparator, which is step (2) below. Finally, we can use the actor's online learning guarantees from Proposition 4 *(Actor's Analysis)* for step (3).

$$(V^{\pi}_1 - V^{\pi_{\text{ALG}}}_1)(s_1) \stackrel{(1)}{=} \frac{1}{K}\sum_{k=1}^{K}\left(V^{\pi}_1 - V^{\pi_k}_1\right)(s_1) \stackrel{(2)}{\le} \frac{1}{K}\sum_{k=1}^{K}\left(V^{\pi}_{1,M_k} - V^{\pi_k}_{1,M_k}\right)(s_1) + U(\pi) \stackrel{(3)}{\le} \mathcal{R}(K) + U(\pi).$$

Rearranging terms and taking $\sup$ over all comparators concludes the proof.

6 Discussion

A key idea of this paper is to introduce pessimism while remaining in the prescribed function class. Doing so allows us to avoid making additional model assumptions, and achieves minimax optimality. Similar ideas have appeared before in the exploration setting (e.g., [Zanette et al., 2020b, Jin et al., 2021, Du et al., 2021]) with similar advantages (batch-style assumptions + minimax regret) *but at the expense of computational tractability*.

Fortunately, the offline RL setting differs from the online setting and we are able to maintain computational tractability by clearly separating the actor's update from the critic evaluation. In this way, *each algorithm solves a simpler task*, and computational tractability is retained.

The numerical evaluation of this procedure and the extension to more general function classes are important next steps, and it will be interesting to see if any of these ideas can be translated to the more challenging exploration setting.

References

[Abbasi-Yadkori et al., 2011] Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*.

[Agarwal et al., 2020a] Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. (2020a). Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*.

[Agarwal et al., 2020b] Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. (2020b). Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pages 64–66.

[Agarwal et al., 2020c] Agarwal, R., Schuurmans, D., and Norouzi, M. (2020c). An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR.

[Antos et al., 2007] Antos, A., Munos, R., and Szepesvári, C. (2007). Fitted q-iteration in continuous action-space mdps.

[Antos et al., 2008] Antos, A., Szepesvári, C., and Munos, R. (2008). Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129.

[Bhandari and Russo, 2020] Bhandari, J. and Russo, D. (2020). A note on the linear convergence of policy gradient methods. *arXiv preprint arXiv:2007.11120*.

[Bubeck, 2014] Bubeck, S. (2014). Convex optimization: Algorithms and complexity. *arXiv preprint arXiv:1405.4980*.

[Buckman et al., 2020] Buckman, J., Gelada, C., and Bellemare, M. G. (2020). The importance of pessimism in fixed-dataset policy optimization. *arXiv preprint arXiv:2009.06799*.

[Chen and Jiang, 2019] Chen, J. and Jiang, N. (2019). Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pages 1042–1051.

[Du et al., 2021] Du, S. S., Kakade, S. M., Lee, J. D., Lovett, S., Mahajan, G., Sun, W., and Wang, R. (2021). Bilinear classes: A structural framework for provable generalization in rl. *arXiv preprint arXiv:2103.10897*.

[Duan et al., 2021] Duan, Y., Jin, C., and Li, Z. (2021). Risk bounds and rademacher complexity in batch reinforcement learning. *arXiv preprint arXiv:2103.13883*.

[Duan and Wang, 2020] Duan, Y. and Wang, M. (2020). Minimax-optimal off-policy evaluation with linear function approximation. *arXiv preprint arXiv:2002.09516*.

[Fan et al., 2020] Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2020). A theoretical analysis of deep q-learning. In *Learning for Dynamics and Control*, pages 486–489. PMLR.

[Farahmand et al., 2016] Farahmand, A.-m., Ghavamzadeh, M., Szepesvári, C., and Mannor, S. (2016). Regularized policy iteration with nonparametric function spaces. *The Journal of Machine Learning Research*, 17(1):4809–4874.

[Farahmand et al., 2010] Farahmand, A.-m., Szepesvári, C., and Munos, R. (2010). Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems (NIPS)*.

[Farajtabar et al., 2018] Farajtabar, M., Chow, Y., and Ghavamzadeh, M. (2018). More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456. PMLR.

[Fu et al., 2020] Fu, Z., Yang, Z., and Wang, Z. (2020). Single-timescale actor-critic provably finds globally optimal policy. *arXiv preprint arXiv:2008.00483*.

[Geist et al., 2019] Geist, M., Scherrer, B., and Pietquin, O. (2019). A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR.

[Hao et al., 2021] Hao, B., Ji, X., Duan, Y., Lu, H., Szepesvári, C., and Wang, M. (2021). Bootstrapping statistical inference for off-policy evaluation. *arXiv preprint arXiv:2102.03607*.

[Jaques et al., 2019] Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. (2019). Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

[Jiang and Huang, 2020] Jiang, N. and Huang, J. (2020). Minimax value interval for off-policy evaluation and policy optimization. *arXiv preprint arXiv:2002.02081*.

[Jiang et al., 2017] Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. (2017). Contextual decision processes with low Bellman rank are PAC-learnable. In Precup, D. and Teh, Y. W., editors, *International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 1704–1713, International Convention Centre, Sydney, Australia. PMLR.

[Jiang and Li, 2016] Jiang, N. and Li, L. (2016). Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR.

[Jin et al., 2021] Jin, C., Liu, Q., and Miryoosefi, S. (2021). Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*.

[Jin et al., 2020a] Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020a). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*.

[Jin et al., 2020b] Jin, Y., Yang, Z., and Wang, Z. (2020b). Is pessimism provably efficient for offline rl? *arXiv preprint arXiv:2012.15085*.

[Kakade, 2001] Kakade, S. M. (2001). A natural policy gradient. *Advances in neural information processing systems*, 14.

[Kallus and Uehara, 2019] Kallus, N. and Uehara, M. (2019). Efficiently breaking the curse of horizon in off-policy evaluation with double reinforcement learning. *arXiv preprint arXiv:1909.05850*.

[Khodadadian et al., 2021] Khodadadian, S., Jhunjhunwala, P. R., Varma, S. M., and Maguluri, S. T. (2021). On the linear convergence of natural policy gradient algorithm. *arXiv preprint arXiv:2105.01424*.

[Kidambi et al., 2020] Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. (2020). Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*.

[Kumar et al., 2019] Kumar, A., Fu, J., Tucker, G., and Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*.

[Kumar et al., 2020] Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*.

[Lan, 2021] Lan, G. (2021). Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *arXiv preprint arXiv:2102.00135*.

[Laroche et al., 2019] Laroche, R., Trichelair, P., and Des Combes, R. T. (2019). Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pages 3652–3661. PMLR.

[Levine et al., 2020] Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

[Liao et al., 2020] Liao, P., Qi, Z., and Murphy, S. (2020). Batch policy learning in average reward markov decision processes. *arXiv preprint arXiv:2007.11771*.

[Liu et al., 2019] Liu, B., Cai, Q., Yang, Z., and Wang, Z. (2019). Neural proximal/trust region policy optimization attains globally optimal policy. *arXiv preprint arXiv:1906.10306*.

[Liu et al., 2018] Liu, Q., Li, L., Tang, Z., and Zhou, D. (2018). Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pages 5356–5366.

[Liu et al., 2020] Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2020). Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*.

[Mannor et al., 2012] Mannor, S., Mebel, O., and Xu, H. (2012). Lightning does not strike twice: Robust mdps with coupled uncertainty. *arXiv preprint arXiv:1206.4643*.

[Modi et al., 2021] Modi, A., Chen, J., Krishnamurthy, A., Jiang, N., and Agarwal, A. (2021). Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*.

[Munos, 2003] Munos, R. (2003). Error bounds for approximate policy iteration. In *ICML*, volume 3, pages 560–567.

[Munos, 2005] Munos, R. (2005). Error bounds for approximate value iteration. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[Nachum et al., 2019a] Nachum, O., Chow, Y., Dai, B., and Li, L. (2019a). Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *arXiv preprint arXiv:1906.04733*.

[Nachum and Dai, 2020] Nachum, O. and Dai, B. (2020). Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*.

[Nachum et al., 2019b] Nachum, O., Dai, B., Kostrikov, I., Chow, Y., Li, L., and Schuurmans, D. (2019b). Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*.

[Nair et al., 2020] Nair, A., Dalal, M., Gupta, A., and Levine, S. (2020). Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*.

[Nedić and Bertsekas, 2003] Nedić, A. and Bertsekas, D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13(1):79–110.

[Puterman, 1994] Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA.

[Rashidinejad et al., 2021] Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. (2021). Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *arXiv preprint arXiv:2103.12021*.

[Raskutti and Mukherjee, 2015] Raskutti, G. and Mukherjee, S. (2015). The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457.

[Shani et al., 2020] Shani, L., Efroni, Y., and Mannor, S. (2020). Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675.

[Shreve and Bertsekas, 1978] Shreve, S. E. and Bertsekas, D. P. (1978). Alternative theoretical frameworks for finite horizon discrete-time stochastic optimal control. *SIAM Journal on control and optimization*, 16(6):953–978.

[Siegel et al., 2020] Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. (2020). Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*.

[Sutton et al., 1999] Sutton, R. S., McAllester, D. A., Singh, S. P., Mansour, Y., et al. (1999). Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer.

[Tang et al., 2019] Tang, Z., Feng, Y., Li, L., Zhou, D., and Liu, Q. (2019). Doubly robust bias reduction in infinite horizon off-policy estimation. *arXiv preprint arXiv:1910.07186*.

[Thomas and Brunskill, 2016] Thomas, P. and Brunskill, E. (2016). Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148.

[Tsybakov, 2009] Tsybakov, A. B. (2009). *Introduction to Nonparametric Estimation*. Springer, New York.

[Uehara et al., 2020] Uehara, M., Huang, J., and Jiang, N. (2020). Minimax weight and q-function learning for off-policy evaluation. In *International Conference on Machine Learning*, pages 9659–9668. PMLR.

[Vershynin, 2010] Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

[Voloshin et al., 2021] Voloshin, C., Jiang, N., and Yue, Y. (2021). Minimax model learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1612–1620. PMLR.

[Wang et al., 2019] Wang, L., Cai, Q., Yang, Z., and Wang, Z. (2019). Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*.

[Wang et al., 2020a] Wang, R., Foster, D. P., and Kakade, S. M. (2020a). What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*.

[Wang et al., 2020b] Wang, Z., Novikov, A., Żołna, K., Springenberg, J. T., Reed, S., Shahriari, B., Siegel, N., Merel, J., Gulcehre, C., Heess, N., et al. (2020b). Critic regularized regression. *arXiv preprint arXiv:2006.15134*.

[Wu et al., 2019] Wu, Y., Tucker, G., and Nachum, O. (2019). Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.

[Wu et al., 2021] Wu, Y., Zhai, S., Srivastava, N., Susskind, J., Zhang, J., Salakhutdinov, R., and Goh, H. (2021). Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*.

[Xie and Jiang, 2020a] Xie, T. and Jiang, N. (2020a). Batch value-function approximation with only realizability. *arXiv preprint arXiv:2008.04990*.

[Xie and Jiang, 2020b] Xie, T. and Jiang, N. (2020b). Q* approximation schemes for batch reinforcement learning: A theoretical comparison. volume 124 of *Proceedings of Machine Learning Research*, pages 550–559, Virtual. PMLR.

[Xie et al., 2019] Xie, T., Ma, Y., and Wang, Y.-X. (2019). Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, pages 9668–9678.

[Yang and Wang, 2020] Yang, L. F. and Wang, M. (2020). Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning (ICML)*.

[Yang et al., 2020] Yang, M., Nachum, O., Dai, B., Li, L., and Schuurmans, D. (2020). Off-policy evaluation via the regularized lagrangian. *arXiv preprint arXiv:2007.03438*.

[Yin et al., 2020] Yin, M., Bai, Y., and Wang, Y.-X. (2020). Near optimal provable uniform convergence in off-policy evaluation for reinforcement learning. *arXiv preprint arXiv:2007.03760*.

[Yin and Wang, 2020] Yin, M. and Wang, Y.-X. (2020). Asymptotically efficient off-policy evaluation for tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3948–3958. PMLR.

[Yu et al., 2020] Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., and Ma, T. (2020). Mopo: Model-based offline policy optimization. *arXiv preprint arXiv:2005.13239*.

[Zanette, 2020] Zanette, A. (2020). Exponential lower bounds for batch reinforcement learning: Batch rl can be exponentially harder than online rl. *arXiv preprint arXiv:2012.08005*.

[Zanette et al., 2020a] Zanette, A., Brandfonbrener, D., Pirotta, M., and Lazaric, A. (2020a). Frequentist regret bounds for randomized least-squares value iteration. In *AISTATS*.

[Zanette et al., 2021] Zanette, A., Cheng, C.-A., and Agarwal, A. (2021). Cautiously optimistic policy optimization and exploration with linear function approximation. *arXiv preprint arXiv:2103.12923*.

[Zanette et al., 2020b] Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. (2020b). Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning (ICML)*.

[Zanette et al., 2020c] Zanette, A., Lazaric, A., Kochenderfer, M. J., and Brunskill, E. (2020c). Provably efficient reward-agnostic navigation with linear value iteration. In *Advances in Neural Information Processing Systems*.

[Zhang et al., 2020a] Zhang, J., Koppel, A., Bedi, A. S., Szepesvari, C., and Wang, M. (2020a). Variational policy gradient method for reinforcement learning with general utilities. *arXiv preprint arXiv:2007.02151*.

[Zhang et al., 2020b] Zhang, R., Dai, B., Li, L., and Schuurmans, D. (2020b). Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*.

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] Yes, they synthetically represent the main contributions
   (b) Did you describe the limitations of your work? [Yes] This is implicit through the required assumptions
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not anticipate negative social consequences
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] The author(s) have read them and have ensured that the paper conforms to them
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Yes, assumptions are discussed in detail and emphasized
   (b) Did you include complete proofs of all theoretical results? [Yes] A proof sketch is included in the main text and all proofs are in the appendix
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)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (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)? [N/A]
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? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]