Instabilities of Offline RL with Pre-Trained Neural Representation

Ruosong Wang [1] **Yifan Wu** [1] **Ruslan Salakhutdinov** [1] **Sham M. Kakade** [2 3]

Abstract

In offline reinforcement learning (RL), we seek to utilize offline data to evaluate (or learn) policies in scenarios where the data are collected from a distribution that substantially differs from that of the target policy to be evaluated. Recent theoretical advances have shown that such sample-efficient offline RL is indeed possible provided certain strong representational conditions hold, else there are lower bounds exhibiting exponential error amplification (in the problem horizon) unless the data collection distribution has only a mild distribution shift relative to the target policy. This work studies these issues from an empirical perspective to gauge how stable offline RL methods are. In particular, our methodology explores these ideas when using features from pre-trained neural networks, in the hope that these representations are powerful enough to permit sample efficient offline RL. Through extensive experiments on a range of tasks, we see that substantial error amplification does occur even when using such pre-trained representations (trained on the same task itself); we find offline RL is stable only under extremely mild distribution shift. The implications of these results, both from a theoretical and an empirical perspective, are that successful offline RL (where we seek to go beyond the low distribution shift regime) requires substantially stronger conditions beyond those which suffice for successful supervised learning.

1. Introduction

Offline reinforcement learning (RL) seeks to utilize offline data to alleviate the sample complexity burden in challenging sequential decision making settings where sample-efficiency is crucial (Mandel et al., 2014; Gottesman et al., 2018; Wang et al., 2018; Yu et al., 2019); it is seeing much recent interest due to the large amounts of offline data already available in numerous scientific and engineering domains. The goal is to efficiently evaluate (or learn) policies, in scenarios where the data are collected from a distribution that (potentially) substantially differs from that of the target policy to be evaluated. Broadly, an important question here is to better understand the practical challenges we face in offline RL problems and how to address them.

Let us start by considering when we expect offline RL to be successful from a theoretical perspective (Munos, 2003; Szepesvári and Munos, 2005; Antos et al., 2008; Munos and Szepesvári, 2008; Tosatto et al., 2017; Chen and Jiang, 2019; Duan et al., 2020). For the purpose of evaluating a given target policy, Duan et al. (2020) showed that under a (somewhat stringent) policy completeness assumption with regards to a linear feature mapping [1] along with data *coverage* assumption, then Fitted-Q iteration (FQI) (Gordon, 1999) — a classical offline Bellman backup based method — can provably evaluate a policy with low sample complexity (in the dimension of the feature mapping). While the coverage assumptions here are mild, the representational conditions for such settings to be successful are more concerning; they go well beyond simple *realizability* assumptions, which only requires the representation to be able to approximate the state-value function of the given target policy.

Recent theoretical advances (Wang et al., 2021) show that without such a strong representation condition, there are lower bounds exhibiting exponential error amplification (in the problem horizon) unless the data collection distribution has only a mild distribution shift relative to the target policy. [2] It is worthwhile to emphasize that this "low distribution condition" is a problematic restriction, since in offline RL, we seek to utilize diverse data collection distributions. As an intuitive example to contrast the issue of distribution shift vs. coverage, consider offline RL for spatial navigation tasks (e.g. (Chang et al., 2020)): coverage in our offline dataset would seek that our dataset has example transitions from a

[1]Carnegie Mellon University, Pittsburgh, PA, USA [2]Microsoft Research, New York, NY, USA [3]University of Washington, Seattle, WA, USA. Correspondence to: Ruosong Wang <ruosongw@andrew.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]A linear feature mapping is said to be complete if Bellman backup of a linear function remains in the span of the given features. See Assumption 2 for a formal definition.

[2]We discuss these issues in more depth in Section 4, where we give a characterization of FQI in the discounted setting

diverse set of spatial locations, while a low distribution shift condition would seek that our dataset closely resembles that of the target policy itself for which we desire to evaluate.

From a practical point of view, it is natural to ask to what extent these worst-case characterizations are reflective of the scenarios that arise in practical applications because, in fact, modern deep learning methods often produce representations that are extremely effective, say for transfer learning (computer vision (Yosinski et al., 2014) and NLP (Peters et al., 2018; Devlin et al., 2018; Radford et al., 2018) have both witnessed remarkable successes using pre-trained features on downstreams tasks of interest). Furthermore, there are number of offline RL methods with promising performance on certain benchmark tasks (Laroche et al., 2019; Fujimoto et al., 2019; Jaques et al., 2020; Kumar et al., 2019; Agarwal et al., 2020; Wu et al., 2020; Kidambi et al., 2020; Ross and Bagnell, 2012). There are (at least) two reasons which support further empirical investigations over these current works: (i) the extent to which these data collection distributions are diverse has not been carefully controlled [3] and (ii) the hyperparameter tuning in these approaches are done in an interactive manner tuned on how the policy actually behaves in the world as opposed to being tuned on the offline data itself (thus limiting the scope of these methods).

In this work we provide a careful empirical investigation to further understand how sensitive offline RL methods are to distribution shift. Along this line of inquiry, One specific question to answer is to what extent we should be concerned about the error amplification effects as suggested by worst-case theoretical considerations.

Our Contributions. We study these questions on a range of standard tasks (6 tasks from the OpenAI gym benchmark suite), using offline datasets with features from pre-trained neural networks trained on the task itself. Our offline datasets are a mixture of trajectories from the target policy itself, along the data from other policies (random or lower performance policies). Note that this is favorable setting in that we would not expect realistic offline datasets to have a large number of trajectories from the target policy itself.

The motivation for using pre-trained features are both conceptual and technical. First, we may hope that such features are powerful enough to permit sample-efficient offline RL because they were learned in an online manner on the task itself. Also, practically, while we are not able to verify if certain theoretical assumptions hold, we may optimistically hope that such pre-trained features will perform well under distribution shift (indeed, as discussed earlier, using

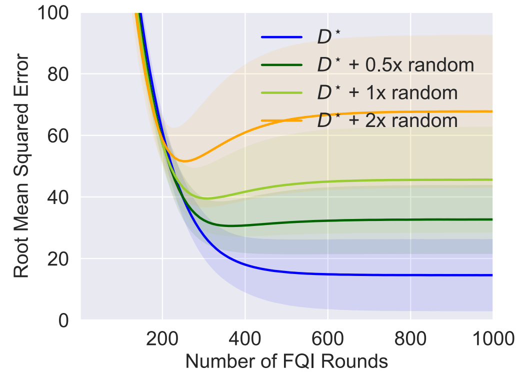


Figure 1. We show the performance of FQI on Walker-2d v2 when applied to policy evaluation. Here the x-axis is the number of rounds we run FQI, and the y-axis is the square root of the mean squared error of the predicted values (smaller is better). The blue line corresponds to performance when the dataset is generated by the target policy itself with 1 million samples, and other lines correspond to the performance when adding more data induced by random trajectories. E.g., the orange line corresponds to the case where we add 2x more data (i.e., 2 million extra samples) induced by random trajectories. As shown here, with more data induced by random trajectories added, the performance of FQI degrades.

pre-trained features has had remarkable successes in other domains). Second, using pre-trained features allows us to decouple practical representational learning questions from the offline RL question, where we can focus on offline RL with a given representation. We provide further discussion on our methodologies in the appendix. We also utilize random Fourier features (Rahimi et al., 2007) as a point of comparison.

The main conclusion of this work, through extensive experiments on a number of tasks, is that: *we do in fact observe substantial error amplification*, even when using pre-trained representations, even we tune hyper-parameters, regardless of what the distribution was shifted to; furthermore, this amplification even occurs under relatively mild distribution shift. As an example, Figure 1 shows the performance of FQI on Walker-2d v2 when our offline dataset has 1 millions samples generated by the target policy itself, with additional samples from random policies.

These experiments also complement the recent hardness results in Wang et al. (2021) showing the issue of error amplification is a real practical concern. From a practical point of view, our experiments demonstrate that the definition of a good representation is more subtle than in supervised learning. These results also raise a number of concerns about empirical practices employed in a number of benchmarks, and they also have a number of implications for moving forward (see Section 6 with regards to these two points).

Finally, it is worth emphasizing that our findings are not sug-

[3] The data collection in many benchmarks tasks are often taken from the data obtain when training an *online* policy, say with deep Q-learning or policy gradient methods.

gesting that offline RL is not possible. Nor does it suggest that there are no offline RL successes, as there have been some successes in realistic domains (e.g. (Mandel et al., 2014; Chang et al., 2020)). Instead, our emphasis is that the conditions for success in offline RL, both from a theoretical and an empirical perspective, are substantially stronger than those in supervised learning settings.

2. Related Work

Theoretical Understanding. Offline RL is closely related to the theory of Approximate Dynamic Programming (Bertsekas and Tsitsiklis, 1995). Existing theoretical work (Munos, 2003; Szepesvári and Munos, 2005; Antos et al., 2008; Munos and Szepesvári, 2008; Tosatto et al., 2017; Duan et al., 2020) usually makes strong representation conditions. In offline RL, the most natural assumption would be realizability, which only assumes the value function of the policy to be evaluated lies in the function class, and existing theoretical work usually make assumptions stronger than realizability. For example, Szepesvári and Munos (2005); Duan et al. (2020); Wang et al. (2021) assume (approximate) closedness under Bellman updates, which is much stronger than realizability. Polynomial sample complexity results are also obtained under the realizability assumption, albeit under coverage conditions (Xie and Jiang, 2020) or stringent distribution shift conditions (Wang et al., 2021). Technically, our characterization of FQI (in Section 4) is similar to the characterization of LSPE by Wang et al. (2021), although we work in the more practical discounted case while Wang et al. (2021) work in the finite-horizon setting.

Error amplification induced by distribution shift is a known issue in the theoretical analysis of RL algorithms. See (Gordon, 1995; 1996; Munos and Moore, 1999; Ormoneit and Sen, 2002; Kakade, 2003; Zanette et al., 2019) for discussion on this topic. Recently, Wang et al. (2021) show that in the finite-horizon setting, without a strong representation condition, there are lower bounds exhibiting exponential error amplification unless the data collection distribution has only a mild distribution shift relative to the target policy. Such lower bound was later generalized to the discounted setting by Amortila et al. (2020). Similar hardness results are also obtained by Zanette (2020), showing that offline RL could be exponentially harder than online RL.

Empirical Work. Error amplification in offline RL has been observed in empirical work (Fujimoto et al., 2019; Kumar et al., 2019) and was called "extrapolation error" in these work. For example, it has been observed in (Fujimoto et al., 2019) that DDPG (Lillicrap et al., 2015) trained on the replay buffer of online RL methods performs significantly worse than the behavioral agent. Compared to previous empirical study on the error amplification issue, in this work, we use pre-trained features which allow us to decouple practical representational learning questions from the offline RL question, where we can focus on offline RL with a given representation. We also carefully control the data collection distributions, with different styles of shifted distributions (those induced by random trajectories or induced by lower performance policies) and different levels of noise.

To mitigate the issue of error amplification, prior empirical work usually constrains the learned policy to be closer to the behavioral policy (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2020; Jaques et al., 2020; Nachum et al., 2019b; Peng et al., 2019; Siegel et al., 2020; Kumar et al., 2020; Yu et al., 2021) and utilizes uncertainty quantification (Agarwal et al., 2019; Yu et al., 2020; Kidambi et al., 2020; Rafailov et al., 2020). We refer interested readers to the survey by Levine et al. (2020) for recent developments on this topic.

3. Background

Discounted Markov Decision Process. Let $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu_{\text{init}})$ be a *discounted Markov Decision Process* (DMDP, or MDP for short) 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 operator, $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ is the reward distribution, $\gamma < 1$ is the discount factor and $\mu_{\text{init}} \in \Delta(\mathcal{S})$ is the initial state distribution. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ chooses an action a based on the state s. The policy π induces a trajectory $s_0, a_0, r_0, s_1, a_1, r_1, \ldots$, where $s_0 \sim \mu_{\text{init}}$, $a_0 \sim \pi(s_0)$, $r_0 \sim R(s_0, a_0)$, $s_1 \sim P(s_0, a_0)$, $a_1 \sim \pi(s_1)$, etc. For the theoretical analysis, we assume $r_h \in [0, 1]$.

Value Function. Given a policy π and $(s, a) \in \mathcal{S} \times \mathcal{A}$, define

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r_{h'} \mid s_0 = s, a_0 = a, \pi\right]$$

and

$$V^\pi(s) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r_h \mid s_0 = s, \pi\right].$$

For a policy π, we define $V^\pi = \mathbb{E}_{s_0 \sim \mu_{\text{init}}}[V^\pi(s_0)]$ to be the expected value of π from the initial state distribution μ_{init}.

Offline Reinforcement Learning. This paper is concerned with the offline RL setting. In this setting, the agent does not have direct access to the MDP and instead is given access to a data distribution $\mu \in \Delta(\mathcal{S} \times \mathcal{A})$. The inputs of the agent is a datasets D, consisting of i.i.d. samples of the form $(s, a, r, s') \in \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$, where $(s, a) \sim \mu$, $r \sim r(s, a)$, $s' \sim P(s, a)$. We primarily focus on the *offline policy evaluation* problem: given a target policy

$\pi : \mathcal{S} \to \Delta(\mathcal{A})$, the goal is to output an accurate estimate of the value of π (i.e., V^π) approximately, using the collected dataset D, with as few samples as possible.

Linear Function Approximation. In this paper, we focus on offline RL with linear function approximation. When applying linear function approximation schemes, the agent is given a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ which can either be hand-crafted or a pre-trained neural network, which transforms a state-action pair to a d-dimensional embedding, and it is commonly assumed that Q-functions can be predicted by linear functions of the features. For our theoretical analysis, we assume $\|\phi(s,a)\|_2 \le 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. For the theoretical analysis, we are interested in the offline policy evaluation problem, under the following assumption.

Assumption 1 (Realizability). *For the policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ to evaluated, there exists $\theta^* \in \mathbb{R}^d$ with $\|\theta^*\|_2 \le \sqrt{d}/(1-\gamma)$ [4], such that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q^\pi(s,a) = (\theta^*)^\top \phi(s,a)$. Here π is the target policy to be evaluated.*

Notation. For a vector $x \in \mathbb{R}^d$ and a positive semidefinite matrix $A \in \mathbb{R}^{d\times d}$, we use $\|x\|_2$ to denote its ℓ_2 norm, $\|x\|_A$ to denote $\sqrt{x^\top A x}$, $\|A\|_2$ to denote its operator norm, and $\sigma_{\min}(A)$ to denote its smallest eigenvalue. For two positive semidefinite matrices A and B, we write $A \succeq B$ if and only if $A - B$ is positive semidefinite.

4. An Analysis of Fitted Q-Iteration in the Discounted Setting

In order to illustrate the error amplification issue and discuss conditions that permit sample-efficient offline RL, in this section, we analyze Fitted Q-Iteration (FQI) (Gordon, 1999) when applied to the offline policy evaluation problem under the realizability assumption. Here we focus on FQI since it is the prototype of many practical algorithms. For example, when DQN (Mnih et al., 2015) is run on off-policy data, and the target network is updated slowly, it can be viewed as an analog of FQI, with neural networks being the function approximator. We give a description of FQI in Algorithm 1. We also perform experiments on temporal difference methods in our experiments (Section 5). For simplicity, we assume a deterministic target policy π.

We remark that the issue of error amplification discussed here is similar to that in Wang et al. (2021), which shows that if one just assumes realizability, geometric error amplification is inherent in offline RL in the finite-horizon setting. Here we focus on the discounted case which exhibit some subtle differences (see, e.g., Amortila et al. (2020)).

[4] Without loss of generality, we assume that we work in a coordinate system such that $\|\theta^*\|_2 \le \sqrt{d}/(1-\gamma)$ and $\|\phi(s,a)\|_2 \le 1$.

Algorithm 1 Fitted Q-Iteration (FQI)

1: **Input:** policy π to be evaluated, number of samples N, regularization parameter $\lambda > 0$, number of rounds T
2: Take samples $(s_i, a_i) \sim \mu$, $r_i \sim r(s_i, a_i)$ and $\overline{s}_i \sim P(s_i, a_i)$ for each $i \in [N]$
3: $\hat{\Lambda} = \frac{1}{N}\sum_{i\in[N]} \phi(s_i,a_i)\phi(s_i,a_i)^\top + \lambda I$
4: $Q_0(\cdot,\cdot) = 0$ and $V_0(\cdot) = 0$
5: **for** $t = 1, 2, \ldots, T$ **do**
6: $\quad \hat{\theta}_t = \hat{\Lambda}^{-1}(\frac{1}{N}\sum_{i=1}^N \phi(s_i,a_i) \cdot (r_i + \gamma \hat{V}_{t-1}(\overline{s}_i)))$
7: $\quad \hat{Q}_t(\cdot,\cdot) = \phi(\cdot,\cdot)^\top \hat{\theta}_t$ and $\hat{V}_t(\cdot) = \hat{Q}_t(\cdot, \pi(\cdot))$
8: **end for**
9: **return** $\hat{Q}_T(\cdot,\cdot)$

Notation. We define Φ to be a $N \times d$ matrix, whose i-th row is $\phi(s_i, a_i)$, and define $\overline{\Phi}$ to be another $N \times d$ matrix whose i-th row is $\phi(\overline{s}_i, \pi(\overline{s}_i))$ (see Algorithm 1 for the definition of $\overline{s}_i$). For each $i \in [N]$, define $\zeta_i = r_i + V(\overline{s}_i) - Q(s_i, a_i)$. Clearly, $\mathbb{E}[\zeta_i] = 0$. We use $\zeta \in \mathbb{R}^N$ to denote a vector whose i-th entry is ζ_i. We use $\Lambda = \mathbb{E}_{(s,a)\sim\mu}\mathbb{E}[\phi(s,a)\phi(s,a)^\top]$ to denote the feature covariance matrix of the data distribution, and use

$$\overline{\Lambda} = \mathbb{E}_{(s,a)\sim\mu, s'\sim P(s,a)}\mathbb{E}[\phi(s',\pi(s'))\phi(s',\pi(s'))^\top]$$

to denote the feature covariance matrix of the one-step lookahead distribution induced by μ and π. We also use $\Lambda_{\text{init}} = \mathbb{E}_{s\sim\mu_{\text{init}}}[\phi(s,\pi(s))\phi(s,\pi(s))^\top]$ to denote the feature covariance matrix induced by the initial state distribution.

Now we present a general lemma that characterizes the estimation error of Algorithm 1 by an equality. Later, we apply this general lemma to special cases.

Lemma 4.1. *Under Assumption 1,*

$$\theta_T - \theta^* = \sum_{t=1}^{T} (\gamma L)^{t-1} \left(\frac{\gamma}{N}\hat{\Lambda}^{-1}\Phi^\top\zeta - \lambda\hat{\Lambda}^{-1}\theta^*\right) + (\gamma L)^T \theta^*$$

where $L = \hat{\Lambda}^{-1}\Phi^\top\overline{\Phi}/N$.

By Lemma 4.1, to achieve a bounded error, the matrix $L = \hat{\Lambda}^{-1}\Phi^\top\overline{\Phi}/N$ should satisfy certain non-expansive properties. Otherwise, the estimation error grows exponentially as t increases, and geometric error amplification occurs. Now we discuss two cases when geometric error amplification does not occur, in which case the estimation error can be bounded with a polynomial number of samples.

Policy Completeness. The policy completeness assumption (Szepesvári and Munos, 2005; Duan et al., 2020) assumes the feature mapping is complete under bellman updates.

Assumption 2 (Policy Completeness)**.** *For any $\theta \in \mathbb{R}^d$, there exists $\theta' \in \mathbb{R}^d$, such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\phi(s,a)^\top \theta' = \mathbb{E}_{r \sim R(s,a), s' \sim P(s,a)}[r + \gamma \phi(s', \pi(s'))^\top \theta].$$

Now we show that under Assumption 2, FQI achieves bounded error with polynomial number of samples.

Lemma 4.2. *Suppose $N \geq \mathrm{poly}(d, 1/\varepsilon, 1/(1-\gamma), 1/\sigma_{\min}(\Lambda))$, by taking $T \geq C \log\left(d/(\varepsilon(1-\gamma))\right)/(1-\gamma)$ for some constant $C > 0$, we have*

$$|\hat{Q}_T(s,a) - Q^\pi(s,a)| \leq \varepsilon$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Proof Sketch. By Lemma 4.1, it suffices to show the non-expansiveness of $L = \hat{\Lambda}^{-1}\Phi^\top\overline{\Phi}/N$. For intuition, let us consider the case where $N \to \infty$ and $\lambda \to 0$. Let $\Phi_{\text{all}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times d}$ denote a matrix whose row indexed by $(s,a) \in \mathcal{S} \times \mathcal{A}$ is $\phi(s,a) \in \mathbb{R}^d$. Let $D^\mu \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ denote a diagonal matrix whose diagonal entry indexed by (s,a) is $\mu(s,a)$. We use $P^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}||\mathcal{A}|}$ to denote a matrix where

$$P^\pi((s,a),(s',a')) = \begin{cases} P(s' \mid s,a) & a' = \pi(s) \\ 0 & \text{otherwise} \end{cases}.$$

When $N \to \infty$ and $\lambda \to 0$, we have $\hat{\Lambda} = \Phi_{\text{all}}^\top D^\mu \Phi_{\text{all}}$ and $\frac{\Phi^\top\overline{\Phi}}{N} = \Phi_{\text{all}}^\top D^\mu P^\pi \Phi_{\text{all}}$. By Assumption 2, for any $x \in \mathbb{R}^d$, there exists x' such that $P^\pi \Phi_{\text{all}} x = \Phi_{\text{all}} x'$. Thus,

$$\begin{aligned}\Phi_{\text{all}}\hat{\Lambda}^{-1}\frac{\Phi^\top\overline{\Phi}}{N}x &= \Phi_{\text{all}}(\Phi_{\text{all}}^\top D^\mu \Phi_{\text{all}})^{-1}\Phi_{\text{all}}^\top D^\mu P^\pi \Phi_{\text{all}} x \\ =&\Phi_{\text{all}} x' = P^\pi \Phi_{\text{all}} x.\end{aligned}$$

Therefore, the magnitude of entries in Φx will not be amplified after applying $\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi}/N$ onto x since $\|P^\pi \Phi_{\text{all}} x\|_\infty \leq \|\Phi_{\text{all}} x\|_\infty$.

In the formal proof, we combine the above with standard concentration arguments to obtain a finite-sample rate. □

We remark that variants of Lemma 4.2 have been known in the literature (see, e.g., (Duan et al., 2020) for a similar analysis in the finite-horizon setting). Here we present Lemma 4.2 mainly to illustrate the versatility of Lemma 4.1.

Low Distribution Shift. Now we focus on the case where the distribution shift between the data distributions and the distribution induced by the target policy is low. Here, our low distribution shift condition is similar to that in Wang et al. (2021), though we focus on the discounted case while Wang et al. (2021) focus on the finite-horizon case.

To measure the distribution shift, our main assumption is as follows.

Assumption 3 (Low Distribution Shift)**.** *There exists $C_{\text{policy}} < 1/\gamma^2$ such that $\overline{\Lambda} \preceq C_{\text{policy}}\Lambda$ and $\Lambda_{\text{init}} \preceq C_{\text{init}}\Lambda$*

Assumption 3 assumes that the data distribution dominates both the one-step lookahead distribution and the initial state distribution, in a spectral sense. For example, when the data distribution μ is induced by the target policy π itself, and π induces the same data distribution for all timesteps $t \geq 0$, Assumption 3 holds with $C_{\text{policy}} = C_{\text{eval}} = 1$. In general, C_{policy} characterizes the difference between the data distribution and the one-step lookahead distribution induced by the data distribution and the target policy.

Now we show that under Assumption 3, FQI achieves bounded error with polynomial number of samples. The proof can be found in the appendix.

Lemma 4.3. *Suppose $N \geq \mathrm{poly}(d, 1/\varepsilon, 1/(1 - C_{\text{policy}}^{1/2}\gamma), 1/C_{\text{init}})$. By taking $T \geq C\log(d \cdot C_{\text{init}}/(\varepsilon(1 - C_{\text{policy}}^{1/2}\gamma)))/(1 - C_{\text{policy}}^{1/2}\gamma)$ for some $C > 0$, under Assumption 1 and 3, we have $\mathbb{E}_{s \sim \mu_{\text{init}}}[(V^\pi(s) - \hat{V}_T(s))^2] \leq \varepsilon$.*

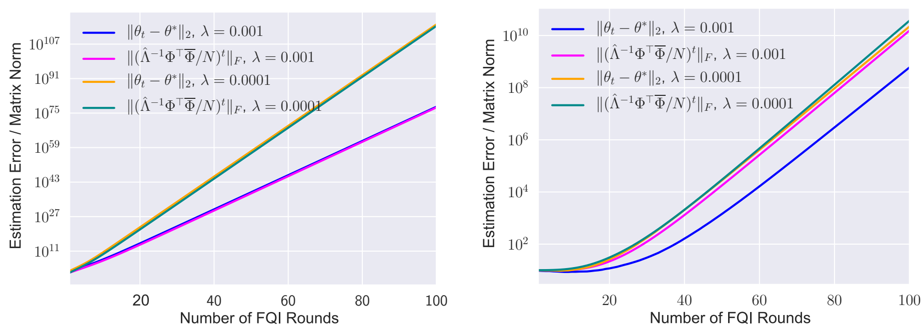
4.1. Simulation Results

We now provide simulation results on a synethic environment to better illustrate the issue of error amplification and the tightness of our characterization of FQI in Lemma 4.1.

Simulation Settings. In our construction, the number of data points is $|D| = N$, where $N = 100$ or $N = 200$. The feature dimension is fixed to be $d = 100$ and the discount factor $\gamma = 0.99$. We draw θ^* from $\mathcal{N}(0, I_d)$. The data distribution, the transition operator and the rewards are all deterministic in this environment. For each $(s_i, a_i, r_i, s_i') \in D$, $\phi(s_i, a_i)$ and $\phi(s_i', \pi(s_i'))$ are independently drawn from $\mathcal{N}(0, I_d)$, and $r_i = \phi(s,a)^\top\theta^* - \gamma\phi(s', \pi(s'))^\top\theta^*$ so that Assumption 1 holds. We then run FQI in Algorithm 1, by setting $T = 100$ and $\lambda = 10^{-4}$ or 10^{-3}. In Figure 2, we plot the estimation error $\|\theta_t - \theta^*\|_2$ and the Frobenius norm of $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t$, for $t = 1, 2, \ldots, 100$. We repeat the experiment for 100 times and report the mean estimation error and the mean Frobenius norm of $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t$.

We remark that our dataset D has sufficient coverage over the feature space, both when $N = 100$ and $N = 200$. This is because the feature covariance matrix has lower bounded eigenvalue with high probability in both cases, according to standard random matrix theory (Chen and Dongarra, 2005).

Results. For deterministic environments, by Lemma 4.1, the estimation error is dominated by $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t\theta^*$. As shown in Figure 2, geometric error amplification does occur, and the norm of $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t$ grows exponentially as t increases. Moreover, the norm of $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t$ has almost the same growth trend as $\|\theta_t - \theta^*\|_2$. E.g., when $N = 200$, the estimation error $\|\theta_t - \theta^*\|_2$ grows exponentially, although



(a) $N = 100$

(b) $N = 200$

Figure 2. In this figure we report the results of the simulation when $N = 100$ and $N = 200$. The x-axis is the number of rounds t we run FQI, while the y-axis is the estimation error $\|\theta_t - \theta^*\|_2$ or the the Frobenius norm of $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t$, taking average over 100 repetitions.

much slower than the case when $N = 100$. In that case, the norm of $(\hat{\Lambda}^{-1}\Phi^\top\overline{\Phi})^t$ also increases much slower than the case when $N = 200$. Our simulation results show that the issue of error amplification could occur even in simple environments, and our theoretical result in Lemma 4.1 gives a tight characterization of the estimation error.

5. Experiments

The goal of our experimental evaluation is to understand whether offline RL methods are sensitive to distribution shift in practical tasks, given a good representation (features extracted from pre-trained neural networks or random features). Our experiments are performed on a range of challenging tasks from the OpenAI gym benchmark suite (Brockman et al., 2016), including two environments with discrete action space (MountainCar-v0, CartPole-v0) and four environments with continuous action space (Ant-v2, HalfCheetah-v2, Hopper-v2, Walker2d-v2). We also provide further discussion on our methodologies in the appendix.

5.1. Experimental Methodology

Our methodology proceeds according to the following steps:

1. We **decide on a (target) policy** to be evaluated, along with a good feature mapping for this policy.
2. **Collect offline data** using trajectories that are a mixture of the target policy along with another distribution.
3. **Run offline RL methods** to evaluate the target policy using the feature mapping found in Step 1 and the offline data obtained in Step 2.

We now give a detailed description for each step.

Step 1: Determine the Target Policy. To find a policy to be evaluated together with a good representation, we run classical online RL methods. For environments with discrete action space (MountainCar-v0, CartPole-v0), we run Deep Q-learning (DQN) (Mnih et al., 2015), while for environments with continuous action space (Ant-v2, HalfCheetah-v2, Hopper-v2, Walker2d-v2), we run Twin Delayed Deep Deterministic policy gradient (TD3) (Fujimoto et al., 2018). The hyperparameters used can be found in Section C. The target policy is set to be the final policy output by DQN or TD3. We also set the feature mapping to be the output of the last hidden layer of the learned value function networks, extracted in the final stage of the online RL methods. Since the target policy is set to be the final policy output by the online RL methods, such feature mapping contains sufficient information to represent the value functions of the target policy. We also perform experiments using random Fourier features (Rahimi et al., 2007).

Step 2: Collect Offline Data. We consider two styles of shifted distributions: distributions induced by random policies and by lower performance policies. When the data collection policy is the same as the target policy, we will see that offline methods achieve low estimation error, as expected. In our experiments, we use the target policy to generate a dataset $D^\star$ with 1 million samples. We then consider two types of datasets induced by shifted distributions: adding random trajectories into $D^\star$, and adding samples induced by lower performance policies into $D^\star$. In both cases, the amount of data from the target policy remains unaltered (fixed to be 1 million). For the first type of dataset, we add 0.5 million, 1 million, or 2 million samples from random trajectories into $D^\star$. For the second type of dataset, we manually pick four lower performance policies $\pi^1_{\mathrm{sub}}, \pi^2_{\mathrm{sub}}, \pi^3_{\mathrm{sub}}, \pi^4_{\mathrm{sub}}$ with $V^{\pi^1_{\mathrm{sub}}} > V^{\pi^2_{\mathrm{sub}}} > V^{\pi^3_{\mathrm{sub}}} > V^{\pi^4_{\mathrm{sub}}}$, and use each of them to collect 1 million samples. We call these four datasets (each with 1 million samples)

$D_{\text{sub}}^1, D_{\text{sub}}^2, D_{\text{sub}}^3, D_{\text{sub}}^4$, and we run offline RL methods on $D^\star \cup D_{\text{sub}}^i$ for each $i \in \{1, 2, 3, 4\}$.

Step 3: Run Offline RL Methods. With the collected offline data and the target policy (together with a good representation), we can now run offline RL methods to evaluate the (discounted) value of the target policy. In our experiments, we run FQI (described in Section 4) and Least-Squares Temporal Difference [5] (LSTD, a temporal difference offline RL method) (Bradtke and Barto, 1996). For both algorithms, the only hyperparameter is the regularization parameter λ (cf. Algorithm 1), which we choose from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-8}\}$. In our experiments, we report the performance of the best-performing λ (measured in terms of the square root of the mean squared estimation error in the final stage of the algorithm, taking average over all repetitions of the experiment); such favorable hyperparameter tuning is clearly not possible in practice (unless we have interactive access to the environment). See Section 5.2 for more discussion on hyperparameter tuning.

In our experiments, we repeat this whole process 5 times. For each FQI round, we report the square root of the mean squared evaluation error, taking average over 100 randomly chosen states. We also report the values ($V^\pi(s)$) of those randomly chosen states in Table 1. We note that in our experiments, the randomness combines both from the feature generation process (representation uncertainty, Step 1) and the dataset (Step 2). Even though we draw millions of samples in Step 2, the estimation of FQI could still have high variance. Note that this is consistent with our theory in Lemma 4.1, which shows that the variance can also be exponentially amplified without strong representation conditions and low distribution shift conditions. We provide more discussion regarding this point in the appendix.

Environment	Discounted Value
CartPole-v0	90.17 ± 20.61
Hopper-v2	321.42 ± 30.26
Walker2d-v2	336.64 ± 49.80

Table 1. Values of Randomly Chosen States. Mean value of the 100 randomly chosen states (used for evaluating the estimations), $\pm$ standard deviation.

5.2. Results and Analysis

Due to space limitations, we present experiment results on Walker2d-v2, Hopper-v2 and CartPole-v0. Other experimental results are provided in the appendix. We also defer results on policy comparison to the appendix.

[5] See the Section C for a description of LSTD.

Distributions Induced by Random Policies. We first present the performance of FQI with features from pre-trained neural networks and distributions induced by random policies. The results are reported in Figure 3. Perhaps surprisingly, compared to the result on $D^\star$, adding more data (from random trajectories) into the dataset generally hurts the performance. With more data added into the dataset, the performance generally becomes worse. Thus, even with features from pre-trained neural networks, the performance of offline RL methods is still sensitive to data distribution.

Distributions Induced by Lower Performance Polices. Now we present the performance of FQI with features from pre-trained neural networks and datasets with samples from lower performance policies. The results are reported in Figure 4. Similar to Figure 3, adding more data into the dataset could hurt performance, and the performance of FQI is sensitive to the quality of the policy used to generate samples. Moreover, the estimation error increases exponentially in some cases (see, e.g., the error curve of $D^\star \cup D_{\text{sub}}^2$ in Figure 4(a)), showing that geometric error amplification is not only a theoretical consideration, but could occur in practical tasks when given a good representation as well.

Random Fourier Features. Now we present the performance of FQI with random Fourier features and distributions induced by random policies. The results are reported in Figure 5. Here we tune the hyperparameters of the random Fourier features so that FQI achieves reasonable performance on $D^\star$. Again, with more data from random trajectories added into the dataset, the performance generally becomes worse. This implies our observations above hold not only for features from pre-trained neural networks, but also for random features. On the other hand, it is known random features achieve reasonable performance in policy gradient methods (Rajeswaran et al., 2017) in the online setting. This suggests that the representation condition required by offline policy evaluation could be stronger than that of policy gradient methods in online setting.

Sensitivity to Hyperparameters. In previous experiments, we tune the regularization parameter λ and report the performance of the best-performing λ. However, we remark that in practice, without access to online samples, hyperparameter tuning is hard in offline RL. Here we investigate how sensitive FQI is to different regularization parameters λ. The results are reported in Figure 5. Here we fix the environment to be Walker2d-v2 and vary the number of additional samples from random trajectories and the regularization parameter λ. As observed in experiments, the regularization parameter λ significantly affects the performance of FQI, as long as there are random trajectories added into the dataset.

Dataset	$D^\star$	$D^\star$ + 0.5x random	$D^\star$ + 1x random	$D^\star$+ 2x random
Hopper-v2	2.18 ± 1.14	9.38 ± 3.84	13.18 ± 2.77	16.86 ± 2.84
Walker2d-v2	13.88 ± 11.22	32.73 ± 11.05	45.61 ± 17.06	67.78 ± 24.77

Table 2. Performance of LSTD. Performance of LSTDwith features from pre-trained neural networks and distributions induced by random policies. Each number of is the square root of the mean squared error of the estimation, taking average over 5 repetitions, $\pm$ standard deviation.

Performance of LSTD. Finally, we present the performance of LSTD with features from pre-trained neural networks and distributions induced by random policies. The results are reported in Table 2. With more data from random trajectories added into the dataset, the performance of LSTD becomes worse. This means the sensitivity to distribution shift is not specific to FQI, but also holds for LSTD.

6. Discussion and Implications

The main conclusion of this work, through extensive experiments on a number of tasks, is that we observe substantial error amplification, even when using pre-trained representations, even we (unrealistically) tune hyper-parameters, regardless of what the distribution was shifted to. Furthermore, this amplification even occurs under relatively mild distribution shift. Our experiments complement the recent hardness results in Wang et al. (2021) showing the issue of error amplification is a real practical concern.

The implications of these results, both from a theoretical and an empirical perspective, are that successful offline RL (where we seek to go beyond the constraining, low distribution shift regime) requires substantially stronger conditions beyond those which suffice for successful supervised learning. These results also raise a number of concerns about empirical practices employed in a number of benchmarks. We now discuss these two points further.

Representation Learning in Offline RL. Our experiments demonstrate that the definition of a good representation in offline RL is more subtle than in supervised learning, since features extracted from pre-trained neural networks are usually extremely effective in supervised learning. Certainly, features extracted from pre-trained neural networks and random features satisfy the realizability assumption (Assumption 1) approximately. However, from our empirical findings, these features do not seem to satisfy strong representation conditions (e.g. Assumption 2) that permits sample-efficient offline RL. This suggests that better representation learning process (feature learning methods that differs from those used in supervised learning) could be a route for achieving better performance in offline RL.

Implications for Empirical Practices and Benchmarks. Our empirical findings suggests a closer inspection of certain empirical practices used in the evaluation of offline RL algorithms.

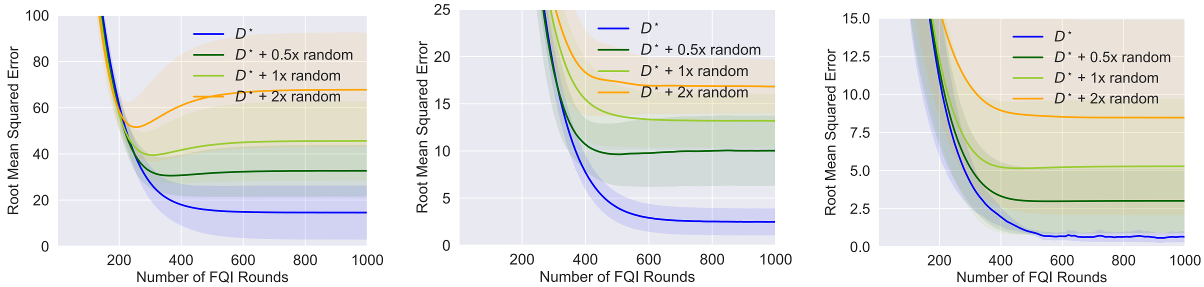
- **Offline Data Collection.** Many empirical settings create an offline dataset under a distribution which contains a large fraction from the target policy itself (e.g. creating the dataset using an online RL algorithm). This may substantially limit the methodology to only testing algorithms in a low distribution shift regime; our results suggests this may not be reflective of what would occur with more realistic and diverse datasets.

- **Hyperparameter Tuning in Offline RL.** A number of methodologies tune hyperparameters using interactive access to the environment, a practice that is clearly not possible with the given offline dataset (e.g. see (Paine et al., 2020) for further discussion). The instability of hyperparameter tuning, as observed in our experiments, suggests that hyperparameter tuning in offline RL may be a substantial hurdle.

Finally, we should remark that the broader motivation of our results (and this discussion) is to help with advancing the field of offline RL through better linking our theoretical understanding with the empirical practices. It is also worth noting that there are notable empirical successes in more realistic settings, e.g. (Mandel et al., 2014; Chang et al., 2020).

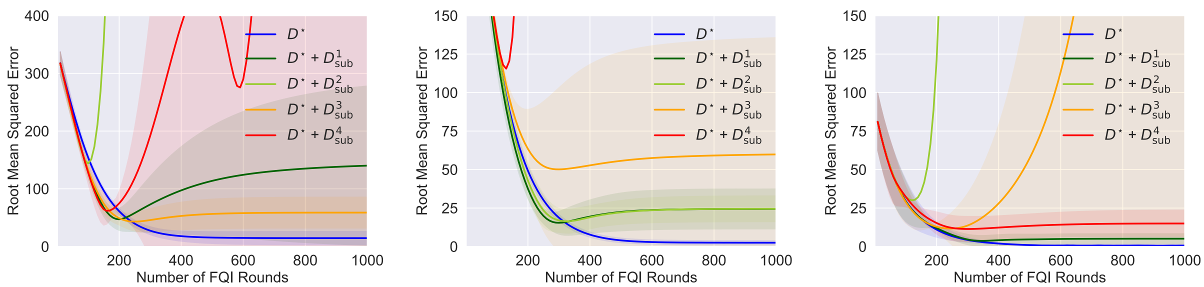
Acknowledgements

The authors would like to thank Alekh Agarwal, Akshay Krishnamurthy, Aditya Kusupati, and Nan Jiang for helpful discussions. Ruosong Wang and Ruslan Salakhutdinov are supported in part by NSF IIS1763562, US Army W911NF1920104, and ONR Grant N000141812861. Sham M. Kakade acknowledges funding from the ONR award N00014-18-1-2247. Part of this work was done while Ruosong Wang was visiting the Simons Institute for the Theory of Computing.



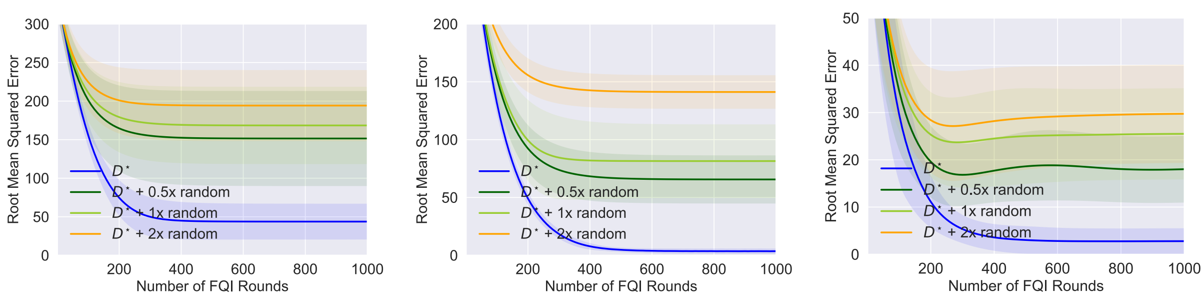
(a) Walker2d-v2 (b) Hopper-v2 (c) CartPole-v0

Figure 3. Performance of FQI with features from pre-trained neural networks and datasets induced by random policies. See Figure 9 in Section D for results on other environments.



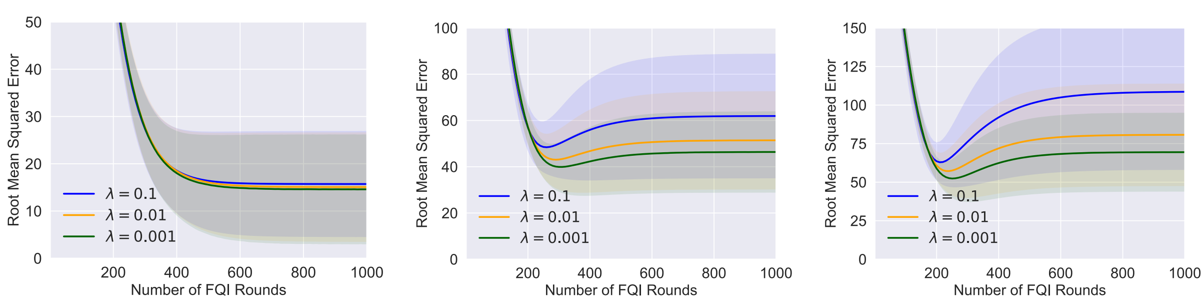
(a) Walker2d-v2 (b) Hopper-v2 (c) CartPole-v0

Figure 4. Performance of FQI with features from pre-trained neural networks and datasets induced by lower performance policies. See Figure 10 in Section D for results on other environments.



(a) Walker2d-v2 (b) Hopper-v2 (c) CartPole-v0

Figure 5. Performance of FQI with random Fourier features and datasets induced by random policies. See Figure 11 in Section D for results on other environments.



(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6. Performance of FQI on Walker2d-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ. See Figure 12 to Figure 17 in Section D for results on other environments.

References

R. Agarwal, D. Schuurmans, and M. Norouzi. Striving for simplicity in off-policy deep reinforcement learning. 2019.

R. Agarwal, D. Schuurmans, and M. Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 2020.

P. Amortila, N. Jiang, and T. Xie. A variant of the wang-foster-kakade lower bound for the discounted setting. *arXiv preprint arXiv:2011.01075*, 2020.

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

D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE, 1995.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

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

M. Chang, A. Gupta, and S. Gupta. Semantic visual navigation by watching youtube videos. *arXiv preprint arXiv:2006.10034*, 2020.

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

Z. Chen and J. J. Dongarra. Condition numbers of gaussian random matrices. *SIAM Journal on Matrix Analysis and Applications*, 27(3):603–620, 2005.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Y. Duan, Z. Jia, and M. Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning*, pages 2701–2709. PMLR, 2020.

M. Dudík, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1097–1104, 2011.

M. Farajtabar, Y. Chow, and M. Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456, 2018.

Y. Feng, T. Ren, Z. Tang, and Q. Liu. Accountable off-policy evaluation with kernel bellman statistics. *arXiv preprint arXiv:2008.06668*, 2020.

S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

G. J. Gordon. Stable function approximation in dynamic programming. In *Machine Learning Proceedings 1995*, pages 261–268. Elsevier, 1995.

G. J. Gordon. Stable fitted reinforcement learning. In *Advances in neural information processing systems*, pages 1052–1058, 1996.

G. J. Gordon. Approximate solutions to markov decision processes. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 1999.

O. Gottesman, F. Johansson, J. Meier, J. Dent, D. Lee, S. Srinivasan, L. Zhang, Y. Ding, D. Wihl, X. Peng, J. Yao, I. Lage, C. Mosch, L. wei H. Lehman, M. Komorowski, M. Komorowski, A. Faisal, L. A. Celi, D. Sontag, and F. Doshi-Velez. Evaluating reinforcement learning algorithms in observational health settings, 2018.

Z. Guo, P. S. Thomas, and E. Brunskill. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pages 2492–2501, 2017.

R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. Vime: Variational information maximizing exploration. *arXiv preprint arXiv:1605.09674*, 2016.

D. Hsu, S. Kakade, T. Zhang, et al. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17, 2012.

N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard. Way off-policy batch deep reinforcement learning of human preferences in dialog, 2020. URL https://openreview.net/forum?id=rJl5rRVFvH.

N. Jiang and J. Huang. Minimax confidence interval for off-policy evaluation and policy optimization. *arXiv preprint arXiv:2002.02081*, 2020.

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

S. M. Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

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

N. Kallus and M. Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *Journal of Machine Learning Research*, 21(167): 1–63, 2020.

R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel : Model-based offline reinforcement learning, 2020.

A. Kumar, J. Fu, M. Soh, G. Tucker, and S. Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *neural information processing systems*, pages 11761–11771, 2019.

A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

R. Laroche, P. Trichelair, and R. Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

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

L. Li, R. Munos, and C. Szepesvari. Toward minimax off-policy value estimation. In *Artificial Intelligence and Statistics*, pages 608–616, 2015.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

Y. Liu, P.-L. Bacon, and E. Brunskill. Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. *arXiv preprint arXiv:1910.06508*, 2019.

T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, pages 1077–1084, 2014.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

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

R. Munos and A. W. Moore. Barycentric interpolators for continuous space and time reinforcement learning. In *Advances in neural information processing systems*, pages 1024–1030, 1999.

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

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

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

D. Ormoneit and Ś. Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

T. L. Paine, C. Paduraru, A. Michi, C. Gulcehre, K. Zolna, A. Novikov, Z. Wang, and N. de Freitas. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

X. B. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

R. Rafailov, T. Yu, A. Rajeswaran, and C. Finn. Offline reinforcement learning from images with latent space models. *arXiv preprint arXiv:2012.11547*, 2020.

A. Rahimi, B. Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.

A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade. Towards generalization and simplicity in continuous control. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6550–6561. Curran Associates, Inc., 2017.

S. Ross and D. Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

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

C. Szepesvári and R. Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887, 2005.

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

P. S. Thomas, G. Theocharous, and M. Ghavamzadeh. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

S. Tosatto, M. Pirotta, C. d'Eramo, and M. Restelli. Boosted fitted q-iteration. In *International Conference on Machine Learning*, pages 3434–3443. PMLR, 2017.

J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8 (1-2):1–230, 2015.

M. Uehara and N. Jiang. Minimax weight and q-function learning for off-policy evaluation. *arXiv preprint arXiv:1910.12809*, 2019.

L. Wang, W. Zhang, X. He, and H. Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

R. Wang, D. P. Foster, and S. M. Kakade. What are the statistical limits of offline rl with linear function approximation? In *International Conference on Learning Representations*, 2021.

Y.-X. Wang, A. Agarwal, and M. Dudık. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pages 3589–3597. PMLR, 2017.

Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning, 2020. URL https://openreview.net/forum?id=BJg9hTNKPH.

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

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

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

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27:3320–3328, 2014.

C. Yu, G. Ren, and J. Liu. Deep inverse reinforcement learning for sepsis treatment. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–3, 2019.

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

T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn. Combo: Conservative offline model-based policy optimization. *arXiv preprint arXiv:2102.08363*, 2021.

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

A. Zanette, A. Lazaric, M. J. Kochenderfer, and E. Brunskill. Limiting extrapolation in linear approximate value iteration. In *Advances in Neural Information Processing Systems*, pages 5616–5625, 2019.

R. Zhang, B. Dai, L. Li, and D. Schuurmans. Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*, 2020a.

S. Zhang, B. Liu, and S. Whiteson. Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, pages 11194–11203. PMLR, 2020b.