Submitted to Statistical Science

Offline Reinforcement Learning in Large State Spaces: Algorithms and Guarantees

Nan Jiang and Tengyang Xie **(draft version; under review)**

Abstract. This article introduces the theory of offline reinforcement learning in large state spaces, where good policies are learned from historical data without online interactions with the environment. Key concepts introduced include expressivity assumptions on function approximation (e.g., Bellman-completeness vs. realizability) and data coverage (e.g., all-policy vs. single-policy coverage), and a rich landscape of algorithms and results is described, depending on the assumptions one is willing to make and the sample and computational complexity guarantees one wishes to achieve. We also describe open questions and connections to adjacent areas.

Key words and phrases: offline reinforcement learning.

1. INTRODUCTION

Reinforcement learning (RL), despite the word "learning" in its name, has been historically about using sampling-based methods for *computational tasks*. As Sutton once put it:[1]

"Much of the [RL] field does not concern learning at all, but just planning from . . . a model of the environment."

This is indeed largely the case in empirical (deep) RL research, where algorithms find near-optimal policies by interacting with simulation environments to sample data trajectories. The goal here is very clear: finding a good policy using a given amount of computation, which includes both the cost of the algorithm and that of sampling data from the simulator.

While this paradigm has led to impressive successes in difficult simulation tasks [75, 87], it becomes increasingly clear that the above paradigm is insufficient for many potential applications we hope to apply RL to, including adaptive clinical trials [112, 113, 79, 74], recommendation systems and customer relationship management [114, 1], online education [12, 64], and more. A commonality of the above scenarios is that human patients/users/students are part of the "environment", and it can be difficult to come up with accurate simulators for the psychological/biological aspects of humans. Therefore, the only environment we have access to is the *real* one, i.e., the real patients, the real users, etc.

This leads to a further problem. Most simulation-based RL algorithms are *online*: while interacting with the environments to collect data, the algorithms will experiment with whatever decisions they deem suited and observe their effects. These decisions can lead to undesirable outcomes, especially at the early stages of learning when the algorithm has little knowledge of the environment. This is not a problem when the environment is a simulator, but can lead to serious consequences when people are part of the environment. Offline RL, which learns from pre-collected data without online interactions, is an answer to this challenge. For real-world environments, such data can come from logging the normal operations of the system without changing how decisions are made.

This article aims to provide a brief introduction to the core concepts and ideas in offline RL theory, focusing on the following two aspects:

- ***Data:*** As we will see, the restriction of no online interactions brings serious algorithmic challenges, and learning guarantees are largely subject to the quality and the quantity of the dataset, requiring us to elevate data as the first and foremost consideration. Hence, our discussion will be focused on the *statistical* aspects of offline RL, mostly the sample complexities, with occasional remarks about computational efficiency.
- ***Function approximation:*** Historically, RL theory with complexity guarantees started in settings where the number of states is finite and small, known as "tabular" RL [48]. In this article we will skip tabular RL and directly address large state spaces where tractable learning often requires function approximation. Readers may find it surprising that, unlike standard settings

Nan Jiang is Assistant Professor of Computer Science, University of Illinois at Urbana-Champaign (e-mail: nanjiang@illinois.edu). Tengyang Xie is Assistant Professor of Computer Science, University of Wisconsin-Madison (e-mail: tx@cs.wisc.edu).

[1] http://incompleteideas.net/RL-FAQ.html.

in other areas of machine learning and statistics, *a hypothesis class that perfectly captures a target function (a.k.a.* realizability*) is often* ***insufficient*** *for learning the said function in RL.* This fact has deep implications in not only learning but also evaluation and testing, and the literature considers a rich variety of function-approximation assumptions that lead to different algorithms and guarantees.

2. A GENTLE START: THE CURSE OF HORIZON

Offline RL promotes a *data*-driven paradigm similar to supervised learning (SL). In this section, we quickly review a minimal theoretical setup of SL and establish its counterpart in offline RL using importance sampling. This analogy will help us appreciate the unique challenges RL faces and prepare for the subsequent discussion of alternative methods.

To start, consider a standard supervised classification setup (the notation will only be used for making the comparison between SL and RL and will not be carried to subsequent sections): we have dataset $\{(X_i, Y_i)\}_{i=1}^n$ drawn i.i.d. from some distribution, where $X_i \in \mathcal{X}$ is the input features (say images) and $Y_i \in \{-1, 1\}$ is the binary label (say whether the image contains a cat), and the goal is to learn $h : \mathcal{X} \to \{-1, 1\}$ that makes accurate predictions of Y from X, measured by the error rate: $\mathrm{err}(h) := \mathbb{E}[\mathbb{I}[Y_i \neq h(X_i)]]$.

A key but perhaps under-appreciated fact is that $\mathrm{err}(h)$ *can be efficiently estimated from data for a fixed* h by $\widehat{\mathrm{err}}(h) := \frac{1}{n}\sum_{i=1}^n \mathbb{I}[Y_i \neq h(X_i)]$. This is because $\widehat{\mathrm{err}}(h)$ is the average of i.i.d. r.v.s $\{\mathbb{I}[Y_i \neq h(X_i)]\}_{i=1}^n$, and Hoeffding's inequality tells us that the deviation from true mean is bounded as

$$|\widehat{\mathrm{err}}(h) - \mathrm{err}(h)| \leq \sqrt{\tfrac{1}{2n}\log\tfrac{2}{\delta}} \tag{1}$$

with probability at least $1 - \delta$.

Eq.(1) is the foundation of everything in SL: we train by Empirical Risk Minimization (ERM) $\arg\max_{h\in\mathcal{H}} \widehat{\mathrm{err}}(h)$, and a basic generalization error bound that depends on $\log|\mathcal{H}|$ can be obtained from Eq.(1) and union bounding over $\mathcal{H}$. In practice, we may try different training algorithms, and rely on $\widehat{\mathrm{err}}(h)$ estimated from holdout datasets for model selection and evaluation (test).

2.1 Off-Policy Evaluation by IS

To establish the counterpart of ERM-then-evaluate scheme for offline RL, the central question is *how to estimate the performance of a candidate solution and establish a guarantee similar to Eq.*(1). In RL, the learning algorithms output decision-making strategies, or *policies*, which are generally different from the policies used to collect the dataset in the first place. Evaluating a policy given data collected by a different policy is known as the problem of *off-policy evaluation* (OPE).

To introduce OPE methods, we consider a standard setup for RL in infinite-horizon discounted Markov Decision Processes (MDPs): An MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$ is specified by its state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $P : \mathcal{S}\times\mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S}\times\mathcal{A} \to [0, R_{\max}]$, discount factor $\gamma \in [0, 1)$, and initial distribution d_0. Here rewards are deterministic and non-negative, which are inconsequential simplifications.

We also assume $\mathcal{S}$ and $\mathcal{A}$ are discrete and finite for convenience; the results we introduce scale to arbitrarily large $\mathcal{S}$ (and large $\mathcal{A}$ sometimes), and can be adapted to continuous $\mathcal{S}$ under appropriate measure-theoretic notation.[2] A (stationary and stochastic) policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a decision-making strategy, and induces a distribution over *trajectories*:

$$\tau := (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_{H-1}, a_{H-1}, r_{H-1}, \ldots),$$

described by the following generative process: $s_0 \sim d_0$, $a_t \sim \pi(\cdot \mid s_t)$, $r_t = R(s_t, a_t)$, $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, $\forall t \geq 0$. The sampling process can go on forever ($t \to \infty$), but for simplicity we assume that after at most H steps the process always goes into a self-loop state with 0 reward (a.k.a. an *absorbing state*) which marks the termination of the system (patient completing a multi-stage treatment program, user concluding a multi-round conversation with a chatbot, etc.).[3]

Given the MDP model, it suffices to specify two things to define an estimation problem:

Estimand $J(\pi)$: Given a policy π we want to evaluate (often called a target/evaluation policy), the estimand is the expected discounted return, defined as:

$$J(\pi) := \mathbb{E}_\pi[\textstyle\sum_{t=0}^\infty \gamma^t r_t],$$

where $\mathbb{E}_\pi[\cdot]$ refers to distribution of trajectories under policy π. We will also use $\Pr_\pi[\cdot]$ to refer to probabilities under the same distribution. This is a standard objective that measures how much total reward a policy is able to collect in expectation. When we assume that the process terminates in H steps, $\sum_{t=0}^\infty$ can be replaced by $\sum_{t=0}^{H-1}$ since all rewards after $t = H$ are 0.

Dataset: In OPE, we have data trajectories (or episodes) collected from a different policy, π_D, often referred to as the behavior/logging policy. More concretely, the dataset is $\{\tau^{(i)} := (s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, \ldots, s_{H-1}^{(i)}, a_{H-1}^{(i)}, r_{H-1}^{(i)})\}_{i=1}^n$, where all actions $a_t^{(i)} \sim \pi_D(\cdot \mid s_t^{(i)})$.

[2] In fact, some early work did consider continuous state spaces [6], but later works often choose finite spaces for readability [15].

[3] This is only needed by importance sampling. When it does not hold, one can truncate an infinitely long trajectory at an *effective horizon* $H = O(1/(1-\gamma))$ due to discounting, as rewards after H steps are discounted so heavily that omitting them only incurs a small error.

IS Estimator: The importance sampling (IS) estimator [69], also known under the names of importance weighting and inverse propensity score (IPS), forms an unbiased estimate of $J(\pi)$ using a single trajectory (the (i) subscript is omitted):

$$\text{IS}(\tau) := \left(\prod_{t=0}^{H-1} \frac{\pi(a_t \mid s_t)}{\pi_D(a_t \mid s_t)}\right)\left(\sum_{t=0}^{H-1} \gamma^t r_t\right). \tag{2}$$

One can show that, as long as $\pi_D(a \mid s) > 0$ for all (s, a) where $\pi(a \mid s) > 0$ (or informally, $\pi/\pi_D < \infty$),

$$\mathbb{E}_{\pi_D}[\text{IS}(\tau)] = \mathbb{E}_{\pi}[\textstyle\sum_{t=0}^{\infty} \gamma^t r_t] \equiv J(\pi),$$

because the importance weight $\prod_{t=0}^{H-1} \frac{\pi(a_t|s_t)}{\pi_D(a_t|s_t)} = \frac{\Pr_\pi[\tau]}{\Pr_{\pi_D}[\tau]}$ converts the distribution from being sampled with π_D to π in expectation. $\pi/\pi_D < \infty$ is a *coverage* condition. In problems with small action spaces, this can be satisfied for any target policy π when π_D is properly randomized and puts nontrivial probabilities on all actions.

Then, given n trajectories, it is straightforward to form an estimation of $J(\pi)$ by $\widehat{J}_{\text{IS}}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \text{IS}(\tau^{(i)})$. $\text{IS}(\tau)$ in Eq.(2) is the most basic version of IS and can be improved in many ways, such as using a data-dependent normalization factor ("weighted IS") and control variates ("doubly robust") [42, 81]. Nevertheless, these estimators share similar high-level sample-complexity characteristics as we discuss below.

Guarantee: Since $\widehat{J}_{\text{IS}}(\pi)$ is the average of i.i.d. r.v.s $\text{IS}(\tau^{(i)})$, we immediately have an estimation guarantee similar to Eq.(1) for SL, except for one thing: Hoeffding's inequality depends on the range of the r.v.s., which is $[0, 1]$ for Eq.(1). For IS, $\sum_{t=0}^{H-1} \gamma^t r_t \in [0, V_{\max}]$ where $V_{\max} := R_{\max}/(1-\gamma)$ is a standard range parameter for the total reward in MDPs, but we also need a bound on the importance weight $\prod_{t=0}^{H-1} \frac{\pi(a_t|s_t)}{\pi_D(a_t|s_t)}$. Assuming

$$\max_{s,a} \frac{\pi(a \mid s)}{\pi_D(a \mid s)} \le C_{\mathcal{A}},$$

it immediately follows that $\prod_{t=0}^{H-1} \frac{\pi(a_t|s_t)}{\pi_D(a_t|s_t)} \le (C_{\mathcal{A}})^H$. Then, using a standard concentration argument,[4] we have

$$|\widehat{J}_{\text{IS}}(\pi) - J(\pi)| \lesssim V_{\max}\sqrt{\frac{(C_{\mathcal{A}})^H}{n}\log\tfrac{1}{\delta}}, \tag{3}$$

where "$\lesssim$" means LHS$= O($RHS$)$ and suppresses multiplicative absolute constants.

With the IS estimator, we can establish an exact parallel of SL's framework for offline RL: given a class of policies Π we wish to optimize, we can simply perform

$$\arg\max_{\pi\in\Pi} \widehat{J}_{\text{IS}}(\pi)$$

to approximately find the best policy in Π. The learned policy can then be evaluated, again using IS but on a hold-out dataset, for model selection and testing.

2.2 The Curse of Horizon

The exact parallel between the IS-based framework and SL brings many desirable properties (which we will certainly miss in subsequent sections!). In fact, the framework is used in industrial applications of contextual bandits, which can be viewed as 1-step RL ($H = 1$) [54]. There are also interesting and practically relevant technical challenges in optimization and generalization when importance weights are part of the loss functions for policy optimization [78, 30].

Unfortunately, the framework has a crucial caveat for multi-step RL: the exponential term $(C_{\mathcal{A}})^H$, which enters the sample complexity of *even evaluating a single policy* (Eq.(3))! This term can be small if $C_{\mathcal{A}}$ is close to 1. In fact, if $C_{\mathcal{A}} = 1 + O(1/H)$, $(C_{\mathcal{A}})^H$ will be a constant. However, that restricts us to only target policies that are very close to the behavior policy.

To see why this can be unsatisfactory, consider an MDP with only one state and two actions. The state always transitions back to itself. Behavior policy collects data using a uniformly random policy. Intuitively, even with a moderately large dataset we will have the needed information to evaluate any policy accurately, but IS still suffers exponential variance when evaluating a deterministic policy. This clearly contradicts the intuition that this is a simple problem and should not require an exponential-in-horizon dataset for learning. The rest of this article will be largely focused on how to address such a "curse of horizon".

Some final remarks about IS before we move on:

On $C_{\mathcal{A}}$ and $|\mathcal{A}|$ If we have full control over the choice π_b but do not have a priori information about π, the best-case scenario for $C_{\mathcal{A}}$ is $|\mathcal{A}|$ when π_b chooses actions uniformly randomly (a.k.a. uniform exploration). In contextual bandit applications [54], a small amount of uniform exploration is often injected into the system to ensure that the collected data can be useful for IS. On the other hand, $C_{\mathcal{A}}$ can be large or even infinite if the behavior policy lacks randomization.

Moreover, IS can also work for large action spaces. If all policies in Π are sufficiently stochastic (e.g., Gaussian policies in robotics [57]), it is possible to have a π_D that provides bounded π/π_D for a rich family of policies, even if $|\mathcal{A}|$ is large or even infinite. In fact, popular deep RL algorithms such as PPO [72] use one-step importance sampling and are frequently applied to problems with large action spaces.

"Constants" in Finite-sample Guarantees Eq.(3) translates to a sample complexity guarantee of

$$n = O((C_{\mathcal{A}})^H V_{\max}^2 \log\tfrac{1}{\delta}/\epsilon^2),$$

[4] Here Bernstein's inequality is used to leverage the "low variance" (compared to its range) of importance weights: given any non-negative function ρ such that $\mathbb{E}[\rho] = 1$ (which is satisfied by importance weights in general), we always have $\mathbb{E}[\rho^2] \le \|\rho\|_\infty$, i.e., variance scales with range linearly instead of quadratically.

for estimating $J(\pi)$ to ϵ accuracy with high probability (i.e., at least $1-\delta$). This is also a good example showing the importance of *finite-sample results* in RL, as asymptotic identification would "hide away" the exponential term $(C_{\mathcal{A}})^H$. For the same reason, the convention of only highlighting the dependence on ϵ (or the number of interaction rounds T in online RL setups) and treating all other quantities as "constants", which is common in adjacent fields such as online learning, can also be inappropriate.

As a bonus, discrete vs. continuous $\mathcal{S}$ and $\mathcal{A}$ are qualitatively different for asymptotic identification, but their boundary is blurred for finite-sample results. If a finite-sample guarantee has no dependence on $|\mathcal{S}|$, extending it to continuous $\mathcal{S}$ is mostly a matter of formality. This allows us to adopt a minimal setup of finite spaces, while the algorithms and insights are directly applicable to continuous spaces.

3. VALUE FUNCTION ESTIMATION

Back to the 1-state-2-action MDP example: what makes it feel tractable? The answer is Markovianity: despite there exist exponentially many action sequences, they all pass through the same *state*. As we will see, methods that properly leverage Markovianity enjoy guarantees when states and actions visited by the target policy are *covered*—in a precise technical sense explained later—in the data. In contrast, IS completely ignores the notion of state and essentially treats all problems as an exponential tree with ever-branching histories. (This makes IS directly applicable to partially observed domains, which we will discuss at the end.)

For problems with finite and small state spaces, one can certainly estimate the transition and reward for each state-action pair separately and compute the policy's return in the estimated MDP, known as (tabular) "certainty equivalence" [51].[5] The question is how to leverage Markovianity in a way that scales to large state spaces. For that we must turn to a familiar object: *value functions*.

Value Functions Given $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, its (Q-)value function is

$$Q^\pi(s,a) := \mathbb{E}_\pi[\textstyle\sum_{t=0}^{\infty} \gamma^t r_t \,|\, s_0 = s, a_0 = a],$$

which is the expected return when a trajectory starts with (s,a) and actions follow π from $t=1$ onwards.[6] Once Q^π is known, $J(\pi)$ can be extracted as

$$J(\pi) = \mathbb{E}_{s\sim d_0}[Q^\pi(s,\pi)],$$

where $f(s,\pi)$ is the shorthand for $\mathbb{E}_{a\sim\pi(\cdot|s)}[f(s,a)]$. $\mathbb{E}_{d_0}[\cdot]$ can be easily estimated from an i.i.d. bag of states sampled from d_0 (e.g., $\{s_0^{(i)}\}_{i=1}^n$ from Section 2.1), and we assume d_0 is known to simplify presentation.

We will show below that learning Q-functions is an effective approach to overcoming the curse of horizon. There are two key questions:

Q1: How to estimate Q^π from data?
Q2: What guarantees can we say about estimated $J(\pi)$?

We start with **Q1** and introduce two important ideas for estimating Q^π in Sections 3.1 and 3.2, and provide analyses based on a notion of state-action coverage in Section 3.3.

3.1 Fitted-Q and Bellman Completeness

Estimation of Q^π is typically based on the fact that it is the unique fixed point of the (policy-specific) Bellman operator, i.e.,

$$Q^\pi = \mathcal{T}^\pi Q^\pi, \tag{4}$$

where $\mathcal{T} : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is defined as: $\forall f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$,

$$(\mathcal{T}^\pi f)(s,a) := \mathbb{E}_{r=R(s,a),s'\sim P(\cdot|s,a)}[r + \gamma f(s',\pi)]. \tag{5}$$

Computational algorithms for solving Q^π are often based on dynamic programming (DP): for example, value iteration (VI) repeatedly applies $\mathcal{T}^\pi$ to an arbitrary initial function f_0, and the contraction property[7] implies that $(\mathcal{T}^\pi)^n f_0 \to Q^\pi$ with a geometric convergence speed in $\|\cdot\|_\infty$, that is,

$$\|(\mathcal{T}^\pi)^n f_0 - Q^\pi\|_\infty \le \gamma^n \|f_0 - Q^\pi\|_\infty. \tag{6}$$

This motivates one of the most popular family of algorithms for value-function estimation, which approximates the operator $\mathcal{T}^\pi$ from data. Note that $\mathcal{T}^\pi f$ in Eq.(5) takes the form of a conditional expectation, which can be written as a regression problem:

$$\mathcal{T}^\pi f \in \operatorname*{arg\,min}_{f'\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \mathcal{L}(f'; f, \pi),$$

where $\mathcal{L}(f'; f, \pi) := \mathbb{E}_D[(f'(s,a) - r - \gamma f(s',\pi))^2]$. This expression suggests that we may approximate $\mathcal{T}^\pi$ with a least-square regression, and for that we must first clarify the form of data we use in this section.

Data Protocol Here D is a shorthand for the distribution of (s,a,r,s') observed in the offline data. In the rest of this article, we no longer need trajectory data, and learn with these transition tuples which can be extracted as $\tau^{(i)} \to (s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}), (s_1^{(i)}, a_1^{(i)}, r_1^{(i)}, s_2^{(i)}), \ldots$. In such a case, the concentration argument for estimation needs to take into account the dependencies between tuples from the same trajectory [6]. We instead consider

[5] Tabular certainty equivalence is actually a special case of FQE/BRM when the function classes contain all possible functions over $\mathcal{S}\times\mathcal{A}$, and hence the analyses of FQE/BRM are directly applicable.

[6] "$s_0 = s, a_0 = a$" are better thought of as defining a new distribution of trajectories, but we will follow the convention and write them as conditions.

[7] $\forall f, f' \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, $\|\mathcal{T}^\pi f - \mathcal{T}^\pi f'\|_\infty \le \gamma \|f - f'\|_\infty$.

a standard simplification, that the dataset $\mathcal{D}$ consists of i.i.d. transition tuples: $(s,a,r,s') \sim D \Leftrightarrow$

$$(s,a) \sim d^D, r = R(s,a), s' \sim P(\cdot \mid s,a).$$

Fitted-Q Algorithm We are now ready to describe Fitted-Q (Evaluation), which can be viewed as the prototype or theoretical version of many empirically popular methods, including the TD family [58]. Fitted-Q assumes a function class $\mathcal{F}$ for modelling Q^π and chooses an arbitrary initialization $f_0 \in \mathcal{F}$. Then, it solves a sequence of least-regression algorithms:

$$f_k \leftarrow \arg\min_{f' \in \mathcal{F}} \widehat{\mathcal{L}}(f'; f_{k-1}, \pi), \tag{7}$$

where $\widehat{\mathcal{L}}$ is the empirical approximation of $\mathcal{L}$ based on dataset $\mathcal{D}$ (we will omit the formula for such straightforward estimations henceforward):

$$\widehat{\mathcal{L}}(f'; f, \pi) := \frac{1}{|\mathcal{D}|} \sum_{(s,a,r,s') \in \mathcal{D}} (f'(s,a) - r - \gamma f(s', \pi))^2.$$

Divergence of Fitted-Q Before we can analyze the algorithm, there is a very serious problem: Fitted-Q can diverge under arguably very strong assumptions.

Proposition 1 ([82]) *FQE can diverge even when all of the following hold:*

1. $|\mathcal{D}| = \infty$ *and minimization in Eq.(7) is exact.*
2. $\mathcal{F}$ *is a 1-dim linear function class that exactly captures* Q^π*, i.e.,* $Q^\pi \in \mathcal{F}$ *(a.k.a.* realizability*).*

The divergence happens under seemingly perfect assumptions: infinite data, exact optimization, simple function class, and exact realizability. Moreover, this so-called "deadly triad" phenomenon [86] is not merely a theoretical construction: deep RL algorithms are known for their instability and training divergence is commonly observed [90]. So what is going wrong?

The problem is that FQE solves a *sequence* of regression problems, $(s,a) \mapsto r + \gamma f_{k-1}$, where the regression target (also known as the TD target) depends on f_{k-1}, the function from the last iteration. Therefore, for FQE to closely mimic VI, we need *every regression in this sequence to be well-specified*, i.e., $\mathcal{F}$ must include (a close approximation of) the Bayes-optimal predictor, $\mathcal{T}^\pi f_{k-1}$, for every k. Given that f_{k-1} depends on data randomness, we can relax f_{k-1} to any function in $\mathcal{F}$ to obtain an assumption independent of the data randomness, leading to a very important assumption in modern offline RL theory and the major expressivity assumption of this section:

Assumption 1 (Bellman Completeness (for $\mathcal{T}^\pi$))

$$\mathcal{T}^\pi f \in \mathcal{F}, \ \forall f \in \mathcal{F}.$$

The assumption asserts that the function space is *closed* under the operator $\mathcal{T}^\pi$, and is thus also referred to as Bellman closure/closedness; see Figure 1 for an illustration. For finite $\mathcal{F}$, completeness implies realizability ($Q^\pi = (\mathcal{T}^\pi)^\infty f \in \mathcal{F}$) and is generally a stronger assumption. The very nature of the assumption is still a topic of debate; below we present several different angles.

Information-theoretic Angle If we know that the true MDP is in some MDP class $\mathcal{M}$ with bounded log-cardinality, then an (approximately) Bellman-complete $\mathcal{F}$ with a mild blow-up in size can be constructed by repeating $\mathcal{F} \leftarrow \mathcal{F} \bigcup \{\mathcal{T}^\pi_M f : f \in \mathcal{F}, M \in \mathcal{M}\}$ $O(H)$ times, where H is the effective horizon (Section 2.1) and the subscript M in $\mathcal{T}^\pi_M$ means the Bellman operator is defined w.r.t. the MDP M.[8]

Structured $\mathcal{F}$ Angle The above reasoning considers completely unstructured $\mathcal{F}$. When more structured function classes (e.g., linear) are desirable, Bellman completeness is often found satisfied in structured MDPs. (In comparison, the information-theoretic construction above does not place any restrictions on the MDP dynamics.) Example scenarios include the low-rank MDP (with a linear function class) and bisimulation abstractions (with a piecewise constant function class under the given state abstraction) [15]. We present the former here as it is a very representative structural model in recent RL theory.

Example 1 (Low-rank MDP [10, 11, 40]) *An MDP is a low-rank MDP with rank* d*, if there exists* $\phi^\star : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$*,* $\psi^\star : \mathcal{S} \to \mathbb{R}^{\mathsf{d}}$*, such that* $P(s' \mid s,a) = \langle \phi^\star(s,a), \psi^\star(s') \rangle$*. Also assume* $R(s,a) = \phi^\star(s,a)^\top \theta_R$ *for some* $\theta_R \in \mathbb{R}^{\mathsf{d}}$*. When* $\phi^\star$ *is known, the setting is called a linear MDP with* $\phi^\star$ *as its features [44]; for any* $f : \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ *and any* π*,* $\mathcal{T}^\pi f$ *is linear in* $\phi^\star$*, so* $\mathcal{F}_{\phi^\star} = \{\langle \phi^\star, \theta \rangle : \theta \in \mathbb{R}^{\mathsf{d}}\}$*, satisfies Bellman completeness.*[9] *When* $\phi^\star$ *is unknown but belongs to a feature class* Φ*,* $\bigcup_{\phi \in \Phi} \mathcal{F}_\phi$ *is still Bellman complete.*

Practical Angle From a more practical viewpoint, the closure nature of the assumption makes it very different from realizability-type assumptions in SL: when we lack expressivity and underfit in SL, we can simply use a richer function class, hoping that the best approximation in class will be closer to the target (at least it does not hurt). However, Bellman completeness is a *non-monotone* assumption, that a richer class may violate the assumption more than its subset! This, among many other challenges,

[8] The construction is more straightforward for the $(\mathcal{F}, \mathcal{G})$ form in Section 3.2: $\mathcal{F} = \{Q^\pi_M : M \in \mathcal{M}\}$, $\mathcal{G} = \{\mathcal{T}_M f : f \in \mathcal{F}, M \in \mathcal{M}\}$.

[9] In more detailed analyses, one often has to add norm constraints in the definition of the linear class $\mathcal{F}_{\phi^\star}$ for concentration purposes. This adds some slight complication to Bellman completeness, since the norm of the linear coefficient may blow up after a Bellman update. This is often overcome by making appropriate norm assumptions on objects like $\phi^\star$, $\psi^\star$, and θ_R; see e.g., [44].

makes model selection highly challenging in offline RL, which we will discuss in Section 5.

3.2 Bellman Residual Minimization

Before we analyze FQE, we will discuss an alternative algorithm for estimating value functions. The divergence in Proposition 1 can be partly attributed to the iterative nature of Fitted-Q. In comparison, in SL we often just write down a loss function and minimize it, and a consistent and "global" loss function is what RL is missing. Can we frame value function as loss minimization?

An immediate idea is to observe that Q^π is the (unique) solution to $f = \mathcal{T}^\pi f$ (Eq.(4)), so we can find f that minimizes the inconsistency of this Bellman equation. The difference, $f - \mathcal{T}^\pi f$, known as the Bellman error (or residual), is a function of (s,a). To turn that into a scalar objective, it is natural to take the expected square error on the offline data distribution D:

$$\mathcal{E}(f;\pi) := \mathbb{E}_D[(f - \mathcal{T}^\pi f)^2]. \tag{8}$$

We may thus attempt to estimate $\mathcal{E}(f;\pi)$ from data and minimize it over $f \in \mathcal{F}$ to find Q^π, and algorithms of this kind are often referred to as Bellman residual (error) minimization (BRM) [6].

The Double-Sampling Problem Unfortunately, the Bellman error in Eq.(8) *cannot be estimated* without further assumptions [24, 77]. The problem is that $\mathcal{T}^\pi f$ is a conditional expectation, which is *inside* square:

$$\mathcal{E}(f;\pi) = \mathbb{E}_{(s,a)\sim d^D}[\big(f(s,a) - \mathbb{E}_{r,s'|s,a}[r + \gamma f(s',\pi)]\big)^2],$$

where $r, s' \mid s, a$ is a shorthand for $r = R(s,a), s' \sim P(\cdot \mid s,a)$. The naïve estimator is to simply ignore the conditional expectation $\mathbb{E}_{r,s'|s,a}$, which becomes $\widehat{\mathcal{L}}(f;f,\pi)$ as defined below Eq.(7). However, this is an incorrect estimate even with infinite data (when $\widehat{\mathcal{L}}(\cdot) \to \mathcal{L}(\cdot)$), as $\mathcal{L}(f;f,\pi) \neq \mathcal{E}(f;\pi)$. More precisely,

$$\mathcal{L}(f;f,\pi) = \mathcal{E}(f;\pi) + \mathcal{L}(\mathcal{T}f;f,\pi).$$

This is a standard bias-variance decomposition similar to regression. In SL regression when we predict real-valued label Y from X, we also have for any $h : \mathcal{X} \to \mathbb{R}$:

$$\mathbb{E}[(Y - h(X))^2] = \mathbb{E}[(\mathbb{E}[Y \mid X] - h(X))^2] + \mathbb{E}[(\mathbb{E}[Y \mid X] - Y)^2],$$

where the two RHS terms correspond to excess risk and inherent label noise, resp. However, in SL the existence of $\mathbb{E}[(\mathbb{E}[Y \mid X] - Y)^2]$ is not a problem even if we want to minimize the excess risk, since the inherent label noise—as its name suggests—is independent of the predictor h, and minimizing $\mathbb{E}[(Y - h(X))^2]$ is equivalent to minimizing excess risk, even if we cannot estimate the latter.

In RL, however, the corresponding term $\mathcal{L}(\mathcal{T}f;f,\pi)$ **depends on the candidate function f itself**, thus making the minimization of $\mathcal{L}(f;f,\pi)$ and $\mathcal{E}(f;\pi)$ not equivalent. This problem was identified by [9], who proposed sampling two independent next-states from each (s,a) to form an unbiased estimate of $\mathcal{E}(f;\pi)$.[10] However, this "double sampling" algorithm can only be performed in a simulator and does not apply in our setup.

Given this problem, a fix proposed by [6] is to explicitly estimate $\mathcal{L}(\mathcal{T}f;f,\pi)$ and subtract it off $\mathcal{L}(f;f,\pi)$. Noting that $\mathcal{L}(\mathcal{T}^\pi f;f,\pi)$ is the Bayes error rate of predicting $r + \gamma f(s',\pi)$ from (s,a) with $\mathcal{T}^\pi f$ as the Bayes-optimal predictor, we can write it as

$$\mathcal{L}(\mathcal{T}f;f,\pi) = \min_{g \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \mathcal{L}(g;f,\pi). \tag{9}$$

When we use a function class $\mathcal{G}$ to model g, the overall estimator for Q^π on the actual dataset $\mathcal{D}$ becomes [6]:

$$\hat{f}^\pi = \arg\min_{f\in\mathcal{F}} \widehat{\mathcal{E}}(f;\pi) \tag{10}$$
$$\text{where } \widehat{\mathcal{E}}(f;\pi) := \max_{g\in\mathcal{G}} \widehat{\mathcal{L}}(f;f,\pi) - \widehat{\mathcal{L}}(g;f,\pi).$$

Clearly, $\widehat{\mathcal{E}}(f;\pi)$ is only a good estimation of $\mathcal{E}(f;\pi)$ if $\min_{g\in\mathcal{G}} \mathcal{L}(g;f,\pi) \approx \mathcal{L}(\mathcal{T}^\pi f;f,\pi)$, which requires that $\mathcal{T}^\pi f \in \mathcal{G}$ for all $f \in \mathcal{F}$. Interestingly, if we use $\mathcal{F}$ itself as $\mathcal{G}$, this becomes $\mathcal{T}^\pi f \in \mathcal{F}$, $\forall f \in \mathcal{F}$, i.e., *Bellman completeness* in Assumption 1! For presentation purposes, **we will stick with $\mathcal{G} \equiv \mathcal{F}$ for the rest of this article** unless otherwise noted (e.g., we will still separate $\mathcal{F}$ and $\mathcal{G}$ when discussing the approximation errors).

Estimation Guarantee Under Bellman completeness, with standard concentration arguments (see e.g., [93]), we obtain that w.p. $\geq 1-\delta$, $\forall f \in \mathcal{F}$,

$$|\widehat{\mathcal{E}}(f;\pi) - \mathcal{E}(f;\pi)| \lesssim \frac{V_{\max}^2}{n} \log \frac{|\mathcal{F}||\Pi|}{\delta}, \tag{11}$$

Given $Q^\pi \in \mathcal{F}$ (a.k.a. *realizability*),[11]

$$\mathcal{E}(\hat{f}^\pi;\pi) \leq_\epsilon \widehat{\mathcal{E}}(\hat{f}^\pi;\pi) \leq \widehat{\mathcal{E}}(Q^\pi;\pi) \leq_\epsilon \mathcal{E}(Q^\pi;\pi) = 0,$$

where $\leq_\epsilon$ is $\leq$ up to a small additive term that corresponds to the RHS of Eq.(11). This further implies that

$$\mathcal{E}(\hat{f}^\pi;\pi) \lesssim \frac{V_{\max}^2}{n} \log \frac{|\mathcal{F}|}{\delta}. \tag{12}$$

The result is for finite $\mathcal{F}$; extensions to continuous $\mathcal{F}$ with bounded $\mathcal{E}_\infty$ covering numbers are feasible, but there are

[10] In regression, this corresponds to sampling two independent labels Y, Y' from the same X and estimating the excess risk as $\mathbb{E}[(h(X) - Y)(h(X) - Y')]$.

[11] Realizability is automatically implied from completeness for $\mathcal{G} = \mathcal{F}$; when $\mathcal{G} \neq \mathcal{F}$, it needs to be assumed separately on $\mathcal{F}$.

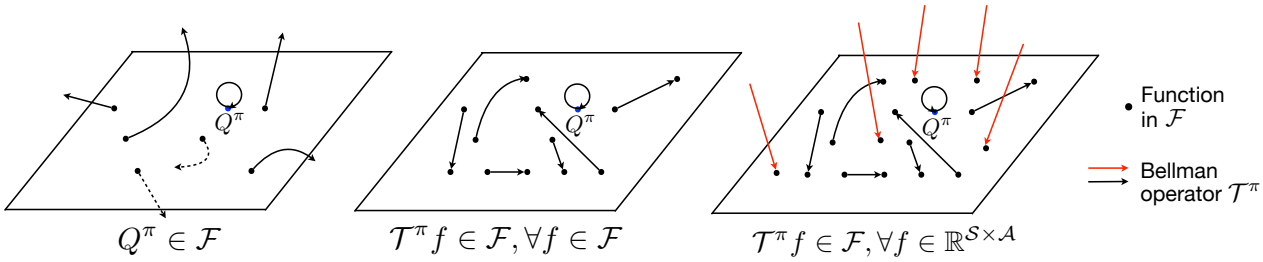

Fig 1: Figurative illustration of different expressivity assumptions on the value-function class $\mathcal{F}$. **Left:** Only realizability $Q^\pi \in \mathcal{F}$ is assumed, and Bellman operator ("$\to$") can generally take functions in $\mathcal{F}$ out of the class. **Middle:** Bellman-completeness, where $\mathcal{F}$ is closed under $\mathcal{T}^\pi$. **Right:** All functions, including those not in $\mathcal{F}$, have their image in $\mathcal{F}$ (Section 4.4).

challenges with other complexity measures due to union bounding over f in the TD target $(r + \gamma f(s', \pi))$, which we will discuss in Section 7.

For downstream analyses, it will be convenient to introduce the notation for (distribution-)weighted p-norm (i.e., $\mathcal{L}^p$ norm): given a distribution μ over a space $\mathcal{X}$,

$$\|(\cdot)\|_{p,\mu}^p := \mathbb{E}_\mu[|\cdot|^p].$$

This allows us to write Eq.(12) as

$$\|\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi\|_{2,D} \lesssim V_{\max} \sqrt{\frac{\log(|\mathcal{F}|/\delta)}{n}}. \tag{13}$$

We now hope to translate this guarantee on the Bellman error of $\hat{f}^\pi$ to a bound on $f - Q^\pi$, and finally the estimation error of $J(\pi)$. A classical result on how Bellman error translates to value error is: $\forall f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$,

$$\|f - Q^\pi\|_\infty \le \frac{\|f - \mathcal{T}^\pi f\|_\infty}{1-\gamma}. \tag{14}$$

However, just as the convergence of VI in Eq.(6), these L_∞ results are difficult to use in the setting of learning in large state spaces: Eq.(14) requires L_∞ Bellman error, but Eq.(13) only controls the much weaker weighted 2-norm. We will address this discrepancy in the next section and see how the notion of state (and action) coverage naturally arises when we depart from the classical L_∞ analyses and consider *error propagation* in Section 3.3.

A few remarks before we move on:

Related Estimators Besides BRM in Eq.(10), there are other ways to estimate $\mathcal{E}(f;\pi)$. One notable approach is to leverage Fenchel duality [20]: $\mathcal{E}(f;\pi) = \min_{g\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_D[g(s,a)(f(s,a) - r - \gamma f(s',\pi)) - \frac{1}{2}g(s,a)^2)]$. The equation still holds if we restrict $g \in \mathcal{G}$ as long as $(f - \mathcal{T}^\pi f) \in \mathcal{G}$. Just like BRM needs $\mathcal{G}$ to realize $\mathcal{T}^\pi f$, these approaches require $\mathcal{G}$ to realize Bellman error $f - \mathcal{T}^\pi f$,[12] so the sample-complexity analyses are similar.

[12] In fact it is easy to go back and forth between these two assumptions: if $\mathcal{T}^\pi f \in \mathcal{G}$, then $\mathcal{F} - \mathcal{G} := \{f - g : f \in \mathcal{F}, g \in \mathcal{G}\}$ will realize all Bellman errors, and vice versa.

Computation BRM in Eq.(10) requires solving a minimax optimization problem. When $\mathcal{F}$ is linear, the problem has a closed-form solution that coincides with LSTDQ (see Section 5.4). When $\mathcal{F}$ is neural net, however, the computational tractability of Eq.(10) becomes less clear; see Section 4.5 for a related discussion.

Approximation Errors So far we make exact expressivity assumptions (e.g., $Q^\pi \in \mathcal{F}$, $\mathcal{T}^\pi f \in \mathcal{G}$). We can consider approximate versions, where approximation errors (e.g., $\min_{f\in\mathcal{F}} \|f - Q^\pi\|$ for some $\|\cdot\|$) will enter the final error bounds; how to insert them and what norm to use are usually clear from the error propagation analyses we will see next. For Bellman completeness, apart from the standard additive error ($\max_{f\in\mathcal{F}} \min_{g\in\mathcal{G}} \|g - \mathcal{T}^\pi f\|_{2,D}$), one can also allow a form of *multiplicative* approximation in BRM analyses [105].

3.3 Guarantee under State-Action Coverage

We are finally ready to see how value-function estimation leads to better coverage. We will focus on the BRM algorithm in Section 3.2 due to its clean analysis, and will provide a sketch for FQE afterwards.

As alluded to at the end of Section 3.2, traditional $\mathcal{E}_\infty$ analysis is insufficient for us. Instead, we introduce a fine-grained version of Eq.(14) that is "distribution-aware":

Lemma 2 (Bellman error telescoping) *For any π, and any $f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$,*

$$J_f(\pi) - J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[f - \mathcal{T}^\pi f],$$

where $J_f(\pi) := \mathbb{E}_{s\sim d_0}[f(s,\pi)]$, and d^π is the discounted state-action occupancy of π: $d^\pi = (1-\gamma)\sum_{t=0}^\infty \gamma^t d_t^\pi$, with $d_t^\pi(s,a) = \Pr_\pi[s_t = s, a_t = a]$.

$J_f(\pi)$ is the estimation of $J(\pi)$ if we treat $f \approx Q^\pi$, and the error is exactly equal to the average Bellman error—without any absolute value or square on each (s,a)—on the discounted occupancy d^π, which reflects the visitation frequency of π over the state-action space.

We are very close to the final guarantee: Lemma 2 shows that we want to control Bellman error under d^π, and Eq.(13) shows we can control the (squared) Bellman error under d^D. The last piece of the puzzle is a result that allows us to convert between the errors under different distributions.

Lemma 3 (Error translation under coverage) *For any function ξ over space $\mathcal{X}$, let $\mu, \nu \in \Delta(\mathcal{X})$ and $1 \le p < \infty$, then*

$$\|\xi\|_{p,\nu}^p \le \|\nu/\mu\|_\infty \cdot \|\xi\|_{p,\mu}^p,$$

where $\|\nu/\mu\|_\infty := \max_x \nu(x)/\mu(x)$. In this article we adopt the convention that $0/0 = 0$, and a non-zero value divided by 0 is infinity.

Error propagation under state-action coverage Lemma 3 suggests a coverage assumption for BRM (Eq.(10)):

Assumption 2 *Assume $\|d^\pi/d^D\|_\infty \le C_\pi < \infty$.*

Based on this assumption, we immediately have the first error guarantee for estimating $J(\pi)$ by BRM :

$$\begin{aligned}
&|J_{\hat{f}^\pi}(\pi) - J(\pi)| \\
(15)\quad &= \left|\frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi]\right| && \text{(Lem. 2)} \\
(16)\quad &\le \frac{1}{1-\gamma}\|\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi\|_{1,d^\pi} \\
(17)\quad &\le \frac{1}{1-\gamma}\|\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi\|_{2,d^\pi} \\
&\le \frac{\sqrt{C_\pi}}{1-\gamma}\|\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi\|_{2,d^D} && \text{(Lem. 3)} \\
(18)\quad &\lesssim \frac{V_{\max}}{1-\gamma}\sqrt{\frac{C_\pi \log(|\mathcal{F}|/\delta)}{n}}. && \text{(Eq.(13))}
\end{aligned}$$

Alternative Coverage Parameters C_π from Assumption 2 is also called *concentrability coefficient* [59, 25, 15], which measures coverage of d^D over d^π by the maximum density ratio. It is essentially used to bound $\|\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi\|_{2,d^\pi}^2 / \|\hat{f}^\pi - \mathcal{T}^\pi \hat{f}^\pi\|_{2,d^D}^2$, and can be loose when the function class $\mathcal{F}$ has additional structures. In particular, since $\hat{f}^\pi \in \mathcal{F}$, this ratio can be relaxed to an a priori (i.e., data-independent) quantity, which is a drop-in improvement for C_π in the above analysis:

$$(19)\qquad C_\pi^{\text{sq}} := \max_{f \in \mathcal{F}} \frac{\|f - \mathcal{T}^\pi f\|_{2,d^\pi}^2}{\|f - \mathcal{T}^\pi f\|_{2,d^D}^2}.$$

(Recall that $\|\cdot\|_{2,d^D}^2 = \mathbb{E}_{d^D}[(\cdot)^2]$.) When $\mathcal{F}$ is the linear class induced by a feature map $\phi \in \mathbb{R}^{\mathsf{d}}$, i.e., $\mathcal{F} \subseteq \{\phi(s,a)^\top \theta\}$, Bellman completeness implies that $\mathcal{T}^\pi f \in \mathcal{F}$ and thus $f - \mathcal{T}^\pi f$ are also linear, in which case Eq.(19) has a very interpretable upper bound:

$$(20)\qquad C_\pi^{\text{sq}} \le \max_{u \in \mathbb{R}^{\mathsf{d}}} \frac{u^\top \Sigma_\pi u}{u^\top \Sigma_D u} = \sigma_{\max}(\Sigma_\pi^{1/2} \Sigma_D^{-1} \Sigma_\pi^{1/2}),$$

where $\Sigma_\pi = \mathbb{E}_{d^\pi}[\phi\phi^\top]$ and Σ_D is defined similarly for d^D, and $\sigma_{\max}$ denotes the largest eigenvalue. This coverage parameter only requires $(s,a) \sim d^D$ to hit all *feature directions* activated by d^π, and can be bounded even if $\|d^\pi/d^D\|_\infty$ is infinity.

In fact, we can further tighten the coverage parameter by directly translating Eq.(15) to on-data error $\|f - \mathcal{T}^\pi f\|_{2,d^D}$ (note that the square on the numerator is now *outside* the expectation) [21, 76]:

$$(21)\qquad C_\pi^{\text{avg}} := \max_{f \in \mathcal{F}} \frac{(\mathbb{E}_{d^\pi}[f - \mathcal{T}^\pi f])^2}{\mathbb{E}_{d^D}[(f - \mathcal{T}^\pi f)^2]}.$$

In the same linear setting as above, we have [21, 101, 106]:

$$(22)\qquad C_\pi^{\text{avg}} \le \mathbb{E}_{d^\pi}[\phi]^\top \Sigma_D^{-1} \mathbb{E}_{d^\pi}[\phi].$$

This is a very notable improvement over Eq.(20), as we now only need to cover a single direction in $\mathbb{R}^{\mathsf{d}}$, the *mean* feature under d^π! This shows that the Cauchy-Schwartz step in Eq.(17) can be surprisingly loose sometimes. That said, not all settings and methods permit the tight definition of coverage in Eq.(21), and sometimes we need to resort to Eq.(19) or similar quantities (Section 4.4). For example, the $\hat{f}^\pi$ learned by BRM under C_π^{avg} coverage only guarantees accurate $J(\pi)$ estimation; if we want stronger guarantees such as bounded $\mathbb{E}_\nu[(f - Q^\pi)^2]$ for some distribution ν (i.e., f has the correct "shape" on ν),[13] we will need d^D to cover d_ν^π (i.e., occupancy induced by ν as the initial distribution) in the C^{sq} sense.

Final comments about coverage:

1. When we relax $\{f - \mathcal{T}^\pi f : f \in \mathcal{F}\}$ to $\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ in C_π^{sq} and C_π^{avg}, we recover a definition that is based on raw densities, but slightly different from and always upper bounded by C_π: $\mathbb{E}_{d^D}[(d^\pi/d^D)^2]$ [95]. Therefore, apart from leveraging the structure of $\mathcal{F}$, C_π^{sq} and C_π^{avg} also contain an implicit improvement of using second moment instead of infinity-norm to measure coverage.
2. Another common assumption is full-rank Σ_D and $\sigma_{\min}(D)$ bounded away from 0. Combined with an upper bound on $\|\phi\|_2$, we can immediately bound C_π^{sq} and C_π^{avg} using $1/\sigma_{\min}(D)$, *regardless of* π. Compared to C_π, this assumption is weaker in that it leverages the linearity of $\{f - \mathcal{T}^\pi f\}$, but is also stronger in ignoring the properties of d^π. In fact, when $\mathsf{d} = \mathcal{S} \times \mathcal{A}$ and ϕ is the (s,a)-indicator feature,

[13] This will be desirable when we perform model selection for learning Q^π; see Section 5.3.

$\Sigma_D = \mathrm{diag}(\{d^D(s,a)\}_{(s,a)\in\mathcal{S}\times\mathcal{A}})$, so $\sigma_{\min}(\Sigma_D) = \min_{s,a} d^D(s,a)$, a type of parameter often referred to as *reachability*. Note that this quantity *necessarily* scales with $1/|\mathcal{S}\times\mathcal{A}|$, whereas C_π does not.

FQE Analysis For FQE in Section 3.1, we can establish an estimation error bound on $\|f_k - \mathcal{T} f_{k-1}\|_{2,D}$ under Bellman completeness that is similar to Eq.(13). The difference is that f_k is not Bellman-consistent with itself, but with the function in previous iteration, f_{k-1}. This makes the error propagation analysis (Section 3.3) slightly more involved for FQE, but the overall idea is similar to BRM and we give a sketch here.

The key observation is that running FQE for K iterations can be viewed as learning the non-stationary value function for a truncated finite-horizon MDP, where the return is defined as $J_K(\pi) := \mathbb{E}_\pi[\sum_{t=0}^{K-1}\gamma^t r_t]$.[14] $J_K(\pi)$ well-approximates the infinite-horizon objective $J(\pi)$ up to a residual term controlled by γ^K, and can be made arbitrarily small by choosing large K. Value function for this objective is time-dependent: $Q_k^\pi(s,a) := \mathbb{E}_\pi[\sum_{t=0}^{k-1}\gamma^t r_t \,|\, s_0 = s, a_0 = a]$, and $J_{Q_K^\pi}(\pi) = J_K(\pi)$. The non-stationary Q-function also satisfies Bellman equation: $Q_k^\pi = \mathcal{T}^\pi Q_{k-1}^\pi$, with $Q_0^\pi \equiv 0$. FQE's output $f_K, \dots, f_1$ are approximations of $Q_K^\pi, \dots, Q_1^\pi$, and small $\|f_k - \mathcal{T}^\pi f_{k-1}\|_{2,D}$ thus implies the Bellman consistency of $\{f_K, \dots, f_1\}$ as a single time-dependent function in the finite-horizon problem. Based on this understanding, we can write down the finite-horizon variant of Lemma 2

Lemma 4 *For any non-stationary policy $\pi_{K:1}$ and function $f_{K:1}$, we have $J_{f_K}(\pi_K) - J_K(\pi_{K:1}) =$*

$$\sum_{t=0}^{K-1}\gamma^t \mathbb{E}_{d_t^{\pi_{K:1}}}[f_{K-t} - \mathcal{T}^{\pi_{K-t-1}} f_{K-t-1}].$$

The form applies to a K-step non-stationary policy $\pi_{K:1}$, which takes a_t according to $\pi_{K-t}(\cdot \,|\, s)$; for now we only need $\pi_K = \cdots = \pi_1 = \pi$, and the general form will be of use later. The rest of the analysis follows similarly as Eqs.(16)–(18), except that we now need d^D to cover d_t^π separately for each t instead of covering d^π as a whole [59, 25, 95].

3.4 Policy Optimization

So far we have focused on OPE, which is crucial for holdout validation and testing (we will discuss more in Section 5). For training, however, we will need to perform policy optimization. An immediate reduction is

$$\arg\max_{\pi\in\Pi} J_{\hat{f}^\pi}(\pi), \tag{23}$$

[14] Here $f_0 \equiv 0$ is assumed, and non-zero f_0 can be added to the objective and does not affect the analysis.

(see the definition of $J_f(\pi)$ in Lemma 2) where $\hat{f}^\pi$ is estimated from Eq.(10) or other methods. Then, we can upper bound the prediction error from Eq.(18) uniformly over $\pi \in \Pi$, which immediately translates to the following policy optimization guarantee:

Theorem 5 *Let $\hat{\pi}$ be the output of Eq.(23) with $\mathcal{G} = \mathcal{F}$. Suppose Assumptions 1 and 2 hold for all $\pi \in \Pi$, then w.p. $\geq 1-\delta$, for any $\pi_{cp} \in \Pi$,*

$$J(\pi_{cp}) - J(\hat{\pi}) \lesssim \frac{V_{\max}}{1-\gamma}\sqrt{\frac{(\max_{\pi\in\Pi} C_\pi)\log(|\mathcal{F}||\Pi|/\delta)}{n}}.$$

Here π_{cp} is any policy we may wish to compete with and can be set to $\arg\max_{\pi\in\Pi} J(\pi)$; we present in this form so that it can be easily compared with the guarantee of improved algorithms in Section 4.1. The additional $|\Pi|$ comes from union bounding the event of accurate $J(\pi)$ estimation across all $\pi \in \Pi$, and can be extended to continuous policy classes with appropriate notions of covering numbers [6, 93, 106, 17]. C_π in the bound can also be replaced with tighter definitions of coverage, such as C_π^{avg} in Eq.(21).

All-policy Coverage The $\max_\pi C_\pi$ term requires that the data distribution provides sufficient coverage over *all* policies $\pi \in \Pi$ (see Figure 2), which puts a heavy burden on d^D that it needs to be very exploratory. Worse still, there are settings where $\max_{\pi\in\Pi} C_\pi$ *will be exponentially large (e.g., linear in $|\Pi|$ or exponentially in H) even for the best-case d^D*. A simple example is where the MDP is a deterministic complete tree and each path is a deterministic policy, and $\max_{\pi\in\Pi} C_\pi \geq |\Pi| = |\mathcal{A}|^H$ for any d^D [15].

That said, certain structures in the dynamics may allow for small $\max_{\pi\in\Pi} C_\pi$ even when the state space is large. An example is the low-rank MDP in Example 1 [15]: given any policy class Π, there always exists d^D such that $\max_{\pi\in\Pi} C_\pi \leq |\mathcal{A}|\mathsf{d}$. The construction is based on the observation that all $d^\pi(s)$ in a low-rank MDP is linear in $\psi(\cdot)$, and a mixture of the barycentric spanner of $\{d^\pi : \pi \in \Pi\}$ will guarantee state coverage of d, and the additional $|\mathcal{A}|$ comes from taking uniform actions.

Computationally-efficient Algorithms Eq.(23) is an information-theoretic objective: implementing it in a literal manner would require enumerating over the policy class Π, which is typically a combinatorial object and thus leads to intractable computation.

There are computationally more feasible algorithms that give comparable guarantees under similar or somewhat stronger assumptions. They are typically based on dynamic-programming (DP) algorithms such as value iteration and policy iteration.

Fitted-Q Iteration (FQI) Besides the policy-specific Bellman equation (which we used to approximate Q^π),

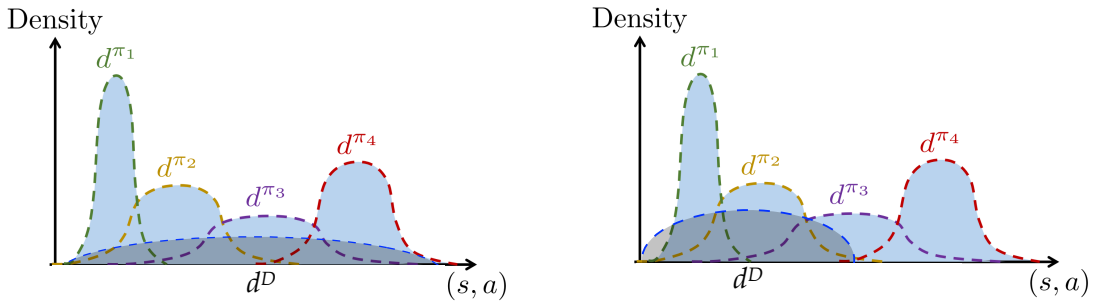

Fig 2: Figurative illustration of different coverage assumptions, adapted from [94]. **Left:** All-policy coverage. **Right:** Data only covers π_1 and π_2, and pessimistic algorithms in Section 4.1 can compete with the best among them.

there is also the Bellman optimality equation, $Q^\star = \mathcal{T}Q^\star$, where $\forall f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$,

$$(\mathcal{T}f)(s,a) := R(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}[\max_{a'\in\mathcal{A}} Q^\star(s',a')].$$

The fixed point of $\mathcal{T}$, namely $Q^\star$, induces a greedy policy $\pi_{Q^\star}(s) := \arg\max_{a\in\mathcal{A}} Q^\star(s,a)$, which achieves optimal return in this MDP for all starting states simultaneously and is also denoted as $\pi^\star$.

Therefore, we may approximate the fixed point of $\mathcal{T}$ in exactly the same way as we do to $\mathcal{T}^\pi$, i.e., either in an iterative manner as in Eq.(7), or by solving a minimax optimization problem as in Eq.(10). After obtaining $\hat{f} \in \mathcal{F}$, we finally output its greedy policy $\pi_{\hat{f}}(s) := \arg\max_{a\in\mathcal{A}} \hat{f}(s,a)$. This way, the algorithm does not maintain a standalone policy class, and instead induces it from the value function class $\mathcal{F}$ as $\Pi = \{\pi_f : f \in \mathcal{F}\}$. The iterative algorithm, known as FQI [22], is more computationally tractable and can be viewed as the theoretical prototype of empirically popular algorithms such as Q-learning (or DQN in deep RL [58]).

Despite FQI being more computationally friendly, we sketch the analysis for the minimax variant below due to its simplicity; the FQI analysis is similar in spirit [59, 60, 15]. For the minimax algorithm, an analysis similar to the one for Eq.(10) can bound $\mathbb{E}_{d^D}[(\hat{f} - \mathcal{T}\hat{f})^2]$, under a variant of the Bellman-completeness assumption

$$\mathcal{T}f \in \mathcal{F}, \forall f \in \mathcal{F}.$$

The following lemma sheds light on its error propagation (see Lemma 3.1 of [45] for the finite-horizon version):

Lemma 6 *For any $f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, $\pi, \pi' : \mathcal{S} \to \Delta(\mathcal{A})$,*

$$J(\pi') - J(\pi) = \frac{1}{1-\gamma}\big(\mathbb{E}_{s\sim d^{\pi'}}[f(s,\pi') - f(s,\pi)] + \mathbb{E}_{d^{\pi'}}[\mathcal{T}^\pi f - f] + \mathbb{E}_{d^\pi}[f - \mathcal{T}^\pi f]\big).$$

We will use this result again in its general form later, but for now we only need its corollary:[15] when we choose $\pi' = \pi^\star$ and $\pi = \pi_f$, the first term on the RHS is non-positive and $\mathcal{T}^\pi f = \mathcal{T}f$, leading to the bound that

$$J(\pi^\star) - J(\pi_f) \le \frac{1}{1-\gamma}\big(\mathbb{E}_{d^{\pi^\star}}[\mathcal{T}f - f] + \mathbb{E}_{d^{\pi_f}}[f - \mathcal{T}f]\big).$$

This result is the counterpart of Lemma 2 but for learning $Q^\star$, which tells us that $J(\pi^\star) - J(\pi_{\hat{f}})$ will be small if (1) $\mathbb{E}_{d^D}[(\hat{f} - \mathcal{T}\hat{f})^2]$ is small (which we already have from above), and (2) d^D covers both $d^{\pi^\star}$ and $d^{\pi_{\hat{f}}}$ (in the sense of bounded $C_{(\cdot)}$ or its refined versions). The former is very reasonable and inevitable: to learn the near optimal policy, it is natural that our data should contain information about it. The latter requires d^D to cover the *learned policy* itself, which is a random quantity we have no direct control over. To come up with an assumption that is independent of the data randomness, we may relax to all possible policies of the form π_f and pay $\max_{f\in\mathcal{F}} C_{\pi_f}$ in the sample complexity guarantee, which coincides with the "all-policy coverage" term $\max_{\pi\in\Pi} C_\pi$ in Theorem 5.

Fitted Policy Iteration (FPI) Besides value iteration, another fundamental planning algorithm for MDPs is called *policy iteration*: $\pi_{k+1} \leftarrow \pi_{Q^{\pi_k}}$, i.e., producing the next policy to be greedy w.r.t. the previous policy's Q-function, and FPI is the algorithm where Q^{π_k} is approximated by algorithms introduced earlier in function class $\mathcal{F}$. Similar to the FQI case, FPI also has its policy class induced by $\mathcal{F}$: $\{\pi_f : f \in \mathcal{F}\}$. The guarantee for FPI is similar to FQI, and we will also see the analysis of a variant of FPI in Section 4.3.

4. PESSIMISTIC POLICY OPTIMIZATION

The policy optimization guarantees we have seen so far all require all-policy coverage and incur dependence on $\max_{\pi\in\Pi} C_\pi$ (C_π can be replaced by its refined versions).

[15] Another notable special case of this lemma is when $f = Q^\pi$. The Bellman error terms vanish and we recover the performance difference (PD) lemma [46]

This requires the dataset to be exploratory which may not hold in practice, and sometimes even implies restrictions on the underlying MDP dynamics (Section 3.4). Therefore, a natural question, which has become a central consideration for offline RL research, is whether we can develop algorithms and guarantees that work for an arbitrary offline dataset (see Figure 2 right).

Below we will show that by using *pessimistic* algorithms, we can relax all-policy coverage to *single-policy* coverage: we can compete with any policy that receives good coverage by the offline data.

4.1 Pessimism in face of uncertainty

Consider a simple multi-armed bandit problem, which can be viewed as an MDP with only one state (so transition always returns to the state itself) and each action (or arm) yields a stochastic reward. For each $a \in \mathcal{A}$ we denote the true mean reward as $R(a)$. The candidate policies correspond to choosing different arms deterministically. On this problem instance, the various policy-optimization algorithms in the previous section all reduce to the following simple procedure:

1. Estimate the mean reward for each arm ($\widehat{R}(a)$).
2. Output the arm with the highest estimated mean.

All-policy coverage requires that each arm is sampled a sufficient number of times in the offline data, so that the reward estimation is accurate for all arms. When it fails to hold, we may choose an arm with low reward if it has received very few samples due to random fluctuation, even when we have abundant samples for another high-reward arm; see Figure 3.

The key to addressing this issue is *uncertainty quantification*: instead of just looking at the point estimates, we should also consider *confidence intervals* (CIs) for the estimate. In the bandit case, we may form $[R^-(a), R^+(a)]$ for each arm by concentration inequalities, and guarantee that the true mean $\mu(a) \in [R^-(a), R^+(a)]$. To reliably compete with a high-sample, high-reward arm, we may choose the arm with the highest *lower-confidence bound* (LCB), $R^-(a)$. The guarantee of algorithms based on point estimates pays for the worst-case estimation error across all arms; in comparison, the LCB algorithm only pays for the uncertainty on the optimal arm: let $\hat{a} = \arg\max_{a\in\mathcal{A}} R^-(a)$ and $a^\star = \arg\max_{a\in\mathcal{A}} R(a)$,

$$
\begin{aligned}
(24) \qquad & R(a^\star) - R(\hat{a}) \\
& \le R^-(a^\star) - R^-(\hat{a}) + R(a^\star) - R^-(a^\star) \\
& \le R(a^\star) - R^-(a^\star).
\end{aligned}
$$

This a general lesson that applies broadly: **when we optimize lower confidence bounds, the guarantee only pays for how much we under-estimate the optimal policy.** In fact, even when the optimal arm has a large CI, we may still compete with any other arm that has good (albeit suboptimal) mean reward and a small CI.

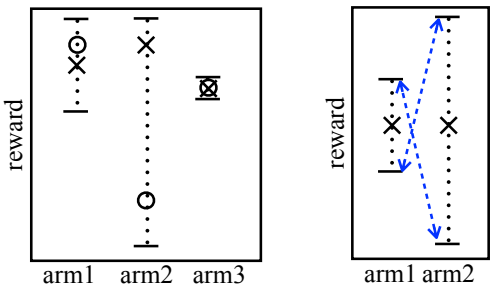

Fig 3: Uncertainty in Multi-armed bandits (MABs). "O" is true mean and "X" is point estimate. **Left:** Example where greedy w.r.t. point estimate chooses arm2 and suffers large loss. Instead, pessimism chooses arm3 with suboptimality bounded by the uncertainty of the best arm (arm1). **Right:** Example where return optimization chooses a different pessimistic policy (arm1) than regret minimization. Both arms have the same regret (height of double-headed arrows), and randomizing between them only incurs half of the regret.

This is an example of the principle of *pessimism in face of uncertainty*, which is the opposite to the well-known optimistic principle in online RL: optimism encourages exploration (i.e., leaving the current data distribution), whereas pessimism encourages exploitation (i.e., staying within the current data distribution).

The next question is how to instantiate this principle for MDPs. One natural idea is to consider uncertainty quantification for $J(\pi)$: if we know $J(\pi)$ falls into an interval $[J^-(\pi), J^+(\pi)]$ computed from data (e.g., the CI for each arm in the bandit example), we may choose the policy that maximizes the lower confidence bound: $\arg\max_{\pi\in\Pi} J^-(\pi)$. One such interval is directly given by the guarantee of BRM, i.e.,

$$
J^{\pm}_{\mathrm{cov}}(\pi) := J_{\hat{f}^\pi}(\pi) \pm \mathrm{eb}(C_\pi),
$$

where $\mathrm{eb}(C_\pi) = O\left(\frac{V_{\max}}{1-\gamma}\sqrt{\frac{C_\pi \log(|\mathcal{F}||\Pi|/\delta)}{n}}\right)$ is the error bound from Eq.(18) after union bounding over Π. Recall that this is a valid bound whenver we have Bellman completeness (Assumption 1), $\mathcal{T}^\pi f \in \mathcal{F}, \forall f \in \mathcal{F}, \pi \in \Pi$. By the same logic as in Eq.(24), the policy $\hat{\pi}$ that optimizes $J^-_{\mathrm{cov}}(\pi)$ can directly compete with any policy $\pi_{\mathrm{cp}} \in \Pi$ (such as $\arg\max_{\pi\in\Pi} J(\pi)$) under the desired single-policy coverage condition:

$$
\begin{aligned}
J(\pi_{\mathrm{cp}}) - J(\hat{\pi}) &\le J^-_{\mathrm{cov}}(\pi) + \mathrm{eb}(C_{\pi_{\mathrm{cp}}}) - J^-_{\mathrm{cov}}(\hat{\pi}) \\
&\le \mathrm{eb}(C_{\pi_{\mathrm{cp}}}).
\end{aligned}
$$

The last line follows from the fact that $\hat{\pi}$ maximizes $J^-_{\mathrm{cov}}(\cdot)$. Compared to Theorem 5, the bound successfully replaced $\max_{\pi\in\Pi} C_\pi$ with $C_{\pi_{\mathrm{cp}}}$.

The problem with the approach is that the error bound $\mathrm{eb}(C_\pi)$ depends on C_π. C_π is known in the bandit example (which is the inverse of the probability of choosing an arm in the offline data), but is generally inaccessible as it involves the discounted occupancy d^π that depends on MDP dynamics (see Lemma 2). Next we show how to obtain the same guarantee without knowing C_π.

4.2 Version-space pessimism

Recall from Eq.(10) that we minimized the following loss over $f \in \mathcal{F}$ to obtain $\hat{f}^\pi$:

$$\widehat{\mathcal{E}}(f;\pi) := \max_{g\in\mathcal{F}} \widehat{\mathcal{L}}(f;f,\pi) - \widehat{\mathcal{L}}(g;f,\pi),$$

as $\widehat{\mathcal{E}}(f;\pi) \approx \mathcal{E}(f;\pi) = \|f - \mathcal{T}f\|^2_{2,D}$ is small for $f = Q^\pi$. This immediately implies that w.p. $\geq 1-\delta$, $\forall \pi \in \Pi$,

$$(25) \qquad Q^\pi \in \mathcal{F}^\pi_{\epsilon_0} := \{f \in \mathcal{F} : \widehat{\mathcal{E}}(f;\pi) \leq \epsilon_0\},$$

where ϵ_0 is the RHS of Eq.(11) (plus union-bounding over Π) to make sure that Q^π is not excluded for all $\pi \in \Pi$. We call $\mathcal{F}^\pi_{\epsilon_0}$ the *version space* for Q^π. This allows us to come up with a lower confidence bound for $J(\pi)$:

$$(26) \qquad J^-_{\mathrm{VS}}(\pi) := \min_{f\in\mathcal{F}^\pi_{\epsilon_0}} J_f(\pi) \leq J_{Q^\pi}(\pi) = J(\pi).$$

and we optimize this objective over $\pi \in \Pi$. Similar to the analysis of $\arg\max_{\pi\in\Pi} J^-_{\mathrm{cov}}(\pi)$, we will pay for an upper bound on the difference between $J(\pi_{\mathrm{cp}})$ and $J^-_{\mathrm{VS}}(\pi_{\mathrm{cp}})$. To bound $J(\pi_{\mathrm{cp}}) - J^-_{\mathrm{VS}}(\pi_{\mathrm{cp}})$, we combine the definition of $\mathcal{F}^\pi_{\epsilon_0}$ with Eq.(11) and immediately have that

$$\mathcal{E}(f;\pi) \leq 2\epsilon_0, \ \forall f \in \mathcal{F}^\pi_{\epsilon_0}.$$

This also holds for $f^\pi_{\min} := \arg\min_{f\in\mathcal{F}^\pi_{\epsilon_0}} J_f(\pi)$ since $f^\pi_{\min} \in \mathcal{F}^\pi_{\epsilon_0}$. Using the same telescoping and error translation analysis as Eqs.(15)–(18), we have

$$J(\pi_{\mathrm{cp}}) - J^-_{\mathrm{VS}}(\pi_{\mathrm{cp}}) \leq \frac{\sqrt{C_\pi}}{1-\gamma}\sqrt{2\epsilon_0}.$$

Plugging in the expression for ϵ_0, we have the more formal guarantee:

Theorem 7 *Fix any $\pi_{cp} \in \Pi$. Assume (1) Assumptions 2 and 1 for $\pi = \pi_{cp}$, and (2) $Q^\pi \in \mathcal{F}$ for all $\pi \in \Pi$. Then, the policy $\hat{\pi}$ that maximizes $J^-_{VS}(\pi)$ over $\pi \in \Pi$ enjoys the following guarantee: w.p. $\geq 1-\delta$,*

$$J(\pi_{cp}) - J(\pi) \lesssim \frac{V_{\max}}{1-\gamma}\sqrt{\frac{C_{\pi_{cp}}\log(|\mathcal{F}||\Pi|/\delta)}{n}}.$$

Similar to before, $C_{\pi_{\mathrm{cp}}}$ can be replaced with its refined versions such as $C^{\mathrm{avg}}_{\pi_{\mathrm{cp}}}$. Compared to Theorem 5, we have relaxed all-policy coverage ($\max_\pi C_\pi$) to single-policy coverage ($C_{\pi_{\mathrm{cp}}}$); in fact, the guarantee can be non-vacuous even when the optimal policy is not covered, as we can simply choose any other comparator policy π_{cp} that is covered by data.

Theorem 7 has also relaxed the expressivity assumptions: it only requires Bellman-completeness for π_{cp}, as that guarantees tightness of the lower bound $J^-_{\mathrm{VS}}(\pi_{\mathrm{cp}})$ which we pay in the guarantee. On the other hand, for any other policy π, all we need is the *validity* of $J^-_{\mathrm{VS}}(\pi)$, i.e., that it is actually a lower bound of $J(\pi)$. As long as $Q^\pi \in \mathcal{F}$, we have $\mathcal{T}^\pi Q^\pi = Q^\pi \in \mathcal{F}$, which implies that $\mathcal{E}(Q^\pi;\pi) \approx 0$ even when $\mathcal{F}$ does not satisfy Bellman completeness for π. As a result, Q^π will never be eliminated from $\mathcal{F}^\pi_{\epsilon_0}$, which guarantees the validity of $J^-_{\mathrm{VS}}(\pi)$.

4.3 An Oracle-efficient Algorithm: PSPI

Despite the significant improvement in coverage, the algorithm is information-theoretic (in the same way as Eq.(23)), and here we introduce a more computationally friendly version. In RL with function approximation, computational efficiency is often stated in the form of *oracle efficiency*, i.e., the computation is efficient if we are given blackbox oracles for certain optimization subroutines that (1) we can reasonably approximate in practice, and (2) the oracle itself can be efficiently implemented (without further assumptions) in special cases such as tabular and linear function classes.

The FQE and FQI algorithms are examples where the oracle assumption is very straightforward: least-square regression oracles over $\mathcal{F}$. For pessimism, however, we require the oracle of computing the worst-case Q-function for any given policy π:

$$(27) \qquad f^\pi_{\min} := \arg\min_{f\in\mathcal{F}^\pi_{\epsilon_0}} J_f(\pi).$$

Recall that $f \in \mathcal{F}^\pi_{\epsilon_0} \iff \widehat{\mathcal{E}}(f;\pi) \leq \epsilon_0$, so this is essentially a constrained optimization problem. [93] showed that this problem (and its "Lagrangian"-like version $\min_{f\in\mathcal{F}} J_f(\pi) + \lambda\widehat{\mathcal{E}}(f;\pi)$)[16] is fully efficient when $\mathcal{F}$ is linear in d-dimensional feature map $\phi(s,a)$, as $\lambda\widehat{\mathcal{E}}(f;\pi)$ has a close-form expression that is quadratic in ϕ with a PSD Hessian, with a close connection to LSTDQ (Section 5.4). Given that $J_f(\pi)$ is also linear in f (thus in ϕ), the overall problem is an instance of convex quadratic programming.[17]

PSPI Given such an oracle, [93] propose the following oracle-efficient algorithm that achieves guarantees under single-policy coverage:

[16] This is not strictly speaking a Lagrangian because λ is fixed and not being optimized. This reflects a more general situation in RL theory, that computationally efficient algorithms are often *not* designed by (1) taking an information-theoretic algorithm, and (2) implementing or approximating it with bounded computation. Rather, the algorithm will take inspiration from the info-theoretic algorithm but have distinct statistical behaviors, which require new analyses.

[17] Similar to Footnote 9, here we also need norm constraints on the linear coefficients to control the boundedness of functions in $\mathcal{F}$, but that only adds a few more convex constraints to the convex program.

1. Initialize π_1 as uniformly random over $\mathcal{A}$.
2. **For** $k = 1, 2, \ldots, K$,
   a) Compute $f_k := f^{\pi_k}_{\min}$ using the oracle.
   b) $\pi_{k+1}(a \mid s) \propto \pi_k(a \mid s) \exp(\eta f_k(s, a))$.
3. Output uniform mixture of $\pi_1, \ldots \pi_K$, Unif$[\pi_{1:K}]$.

The "uniform mixture" is *trajectory-level* mixture, i.e., when executing the policy, we will randomly sample an integer i uniformly between 1 to K, and roll out the trajectory with policy π_i. The number i only gets re-sampled when another trajectory is generated. In general, such trajectory-level mixture creates non-Markov and history-dependent policies, even if the "base policies" $(\pi_1, \ldots, \pi_K)$ themselves are Markov, so this can be viewed as an instance of *improper learning*, which is common when no-regret algorithms are incorporated in RL [71]. As a direct consequence, $J(\text{Unif}[\pi_{1:K}]) = 1/K \sum_{k=1}^K J(\pi)$.

Computationally, apart from calling the oracle in Eq.(27), we also need to execute the policy update step 2b). Here we no longer use a standalone policy class, but have the policy class induced from the value function class $\mathcal{F}$ similar to FQI and FPI in Section 3.4. The difference is that instead of hard argmax (i.e., greedy) over $f \in \mathcal{F}$ we are now taking softmax policies over mixture of $f \in \mathcal{F}$, with the following implicit policy class:

$$\Pi = \{\pi(\cdot \mid s) \propto \exp(\eta \textstyle\sum_{i=1}^k f^{(i)}(s, \cdot)) : 1 \le k \le K, f_{1:k} \in \mathcal{F}\}.$$

Naïve computation of this policy will require storing its tabular representation (i.e., $|\mathcal{S} \times \mathcal{A}|$ numbers), which is clearly inefficient in large state spaces. However, to compute $f_{k+1} = f^{\pi_{k+1}}_{\min}$ we only need to evaluate π_{k+1} on data points in $\mathcal{D}$. Hence, "lazy evaluation" suffices, where we calculate $\pi(\cdot \mid s)$ on demand for any queried s. Since $\pi_{k+1}(\cdot \mid s) \propto \exp(\eta \sum_{i=1}^k f_i(s, \cdot))$, we can compute $\pi(a \mid s)$ as long as we store (the model parameters of) $f_i \in \mathcal{F}$ for previous i. The normalization step will incur linear-in-$|\mathcal{A}|$ computational complexity.

Error decomposition We now provide the analysis of PSPI, which will explain the design of the policy update step 2b). Let π' be any comparator policy π_{cp}, and π be the mixture policy we output, then

$$J(\pi_{\text{cp}}) - J(\text{Unif}[\pi_{1:K}]) = 1/K \textstyle\sum_{k=1}^K (J(\pi_{\text{cp}}) - J(\pi_k)). \tag{28}$$

Now we invoke Lemma 6 on each k, by choosing $f = f_k$:

$$J(\pi_{\text{cp}}) - J(\pi_k) = \frac{1}{1-\gamma}\Big(\mathbb{E}_{s \sim d^{\pi_{\text{cp}}}}[f_k(s, \pi_{\text{cp}}) - f_k(s, \pi_k)] + \mathbb{E}_{d^{\pi_{\text{cp}}}}[\mathcal{T}^{\pi_k} f_k - f_k] + \mathbb{E}_{d^{\pi_k}}[f_k - \mathcal{T}^{\pi_k} f_k]\Big).$$

First, we apply Lemma 2 "reversely" and the 3rd term becomes

$$\mathbb{E}_{d^{\pi_k}}[\mathcal{T}^{\pi_k} f_k - f_k] = J(\pi_k) - J_{f_k}(\pi_k) \le 0.$$

"≤ 0" follows directly from that $f_k = f^{\pi_k}_{\min}$ and $J_{f_k}(\pi_k) = J^-_{\text{VS}}(\pi_k)$ is a pessimistic estimation (Eq.(26)). For the 2nd term, note that any function in $\mathcal{F}^{\pi_k}_{\epsilon_0}$, including $f_k = f^k_{\min}(\pi_k)$, has well controlled $\mathbb{E}_D[(f - \mathcal{T}^{\pi_k} f)^2]$.[18] This translates to the error w.r.t. $d^{\pi_{\text{cp}}}$ under single-policy coverage, i.e., bounded $C^{\text{avg}}_{\pi_{\text{cp}}}$.

Therefore, the pessimistic design of $f_k = f^{\pi_k}_{\min}$ automatically handles the 2nd and the 3rd terms. We only need to take care of the first term now, which is

$$\mathbb{E}_{s \sim d^{\pi_{\text{cp}}}}[f_k(s, \pi_{\text{cp}}) - f_k(s, \pi_k)]. \tag{29}$$

So far we have not used any properties of π_k. Hence, the only design goal for π_k, i.e., the policy update rule, is to make Eq.(29) as negative as possible. Ideally we would like to choose π_k to be greedily w.r.t. f_k, but this is against the causal order: $f_k = f^{\pi_k}_{\min}$ is calculated *given* π_k *as input*!

Mirror Descent Note that Eq.(29) is taking expectation w.r.t. $d^{\pi_{\text{cp}}}$, so we can aggregate such terms across k in Eq.(28):

$$\mathbb{E}_{s \sim d^{\pi_{\text{cp}}}}\left[\frac{1}{K} \textstyle\sum_{k=1}^K (f_k(s, \pi_{\text{cp}}) - f_k(s, \pi_k))\right].$$

Instead of trying to come up with π_k that maximizes $f_k(s, \cdot)$ (which is against the causal order, as noted above), we now have an easier task: design $\pi_{1:K}$ to maximize $\sum_k f_k(s, \cdot)$ and compete with a fixed benchmark π_{cp}. For a fixed s, this design problem fits a classical setting in no-regret learning: given a discrete space $\mathcal{X}$,

Online Linear Optimization in the Simplex.
For round $k = 1, 2, \ldots, K$,

1. Learner proposes distribution $p_k \in \Delta(\mathcal{X})$.
2. Nature chooses a (bounded and possibly adversarial) function $f_k \in \mathbb{R}^{\mathcal{X}}$.

Goal: minimize regret $\sum_{k=1}^K (E_p[f_k] - E_{p_k}[f_k])$ against any static benchmark $p \in \Delta(\mathcal{X})$.

Mapping this protocol onto our problem, we have $\mathcal{X} \longleftarrow \mathcal{A}$, $p_k \longleftrightarrow \pi_k(\cdot \mid s)$, $f_k \in \mathbb{R}^{\mathcal{X}} \longleftrightarrow f_k(s, \cdot) \in \mathbb{R}^{\mathcal{A}}$, and $p \in \Delta(\mathcal{X}) \longleftrightarrow \pi_{\text{cp}}(\cdot \mid s) \in \Delta(\mathcal{A})$. Therefore, classical algorithm that provides regret bound, such as *mirror descent*, can be directly applied to our problem. Indeed, the policy update rule in PSPI (step 2b) is literally running mirror descent on each state individually.[19] With appropriate learning rate η, mirror descent enjoys the guarantee

[18] This requires Bellman-completeness under π_k. Given that π_k is random, we relax it to $\mathcal{T}^\pi f \in \mathcal{F}, \forall f \in \mathcal{F}, \pi \in \Pi$. Hence, PSPI makes stronger expressivity assumptions than Theorem 7 (which only needs completeness for π_{cp}), and is more similar to Theorem 5.

[19] This is a version of Natural Policy Gradient (NPG) [47, 3].

[34, 93]:

$$\frac{1}{K}\sum_{k=1}^{K}(f_k(s,\pi_{\text{cp}})-f_k(s,\pi_k)) \lesssim \frac{V_{\max}}{1-\gamma}\sqrt{\frac{\log|\mathcal{A}|}{K}}.$$

This shows that we can control Eq.(29) by increasing the number of iterations K. This way, we have all terms coming out of Eq.(28) under control, under Bellman completeness and single-policy coverage.

4.4 Pointwise Pessimism: PEVI

The PSPI algorithm requires a nontrivial optimization oracle of computing $f^{\pi}_{\min}$. Despite its efficiency in the linear setting, its general feasibility is unclear. Therefore, it is worth asking if we can enjoy single-policy coverage under more straightforward computational oracles.

One such example is PEVI [45], which requires a regression oracle that fits $(s,a)\mapsto r+\gamma v(s')$ for any $v\in\mathbb{R}^{\mathcal{S}}$ with **pointwise uncertainty quantification**: let $f^\star$ be the Bayes-optimal predictor and $\hat{f}\in\mathcal{F}$ be the learned predictor, we also need $b\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}_{\geq 0}$ (often called a *bonus term*), such that with high probability,

$$f^\star(s,a)\in[\hat{f}(s,a)-b(s,a),\hat{f}(s,a)+b(s,a)].$$

PEVI [45] Given the oracle, we can modify FQI to incorporate uncertainty $b(s,a)$ to ensure pessimism: let $f_0^-\equiv 0$,

For $k=1,2,\dots,K$,

1. Fit $(s,a)\mapsto r+\max_{a'} f_{k-1}^-(s',a')$ on $\mathcal{D}$. Let $\hat{f}_k\in\mathcal{F}$ and b_k be oracle outputs.
2. $f_k^-(s,a):=\hat{f}_k(s,a)-b(s,a)$.

Output: non-stationary policy $\pi_{K:1}:=\{\pi_K,\dots,\pi_1\}$, where $\pi_k:=\pi_{f_k^-}$, i.e., the greedy policy w.r.t. f_k^-.

Error propagation and coverage definition The first step of the analysis is to show pessimism, that $f_k^-(s,a)\leq Q_k^{\pi_{k-1:1}}$ (recall the non-stationary setup in the FQE analysis in Section 3.1). Throughout the analysis we assume the high-probability event that all uncertainty quantification is valid.

Pessimism can be shown by induction, based on the monotone property of Bellman operators: for any $f\leq f'$ (pointwise) and π, we have $\mathcal{T}^\pi f\leq\mathcal{T}^\pi f'$. Therefore, if we know that $f_{k-1}^-\leq Q_{k-1}^{\pi_{k-2:1}}$ (the base case at $k=1$ holds trivially with $f_0^-\equiv Q_0^{(\cdot)}\equiv 0$), then

$$f_k^-\leq\mathcal{T}f_{k-1}^-=\mathcal{T}^{\pi_{k-1}}f_{k-1}^-\leq\mathcal{T}^{\pi_{k-1}}Q_{k-1}^{\pi_{k-2:1}}=Q_k^{\pi_{k-1:1}}.$$

The first inequality follows from the validity of uncertainty quantification and the Bayes-optimal predictor for the k-th regression is $f_k^\star=\mathcal{T}f_{k-1}^-$; the second follows from the greediness of π_k w.r.t. f_{k-1}^-; the third from monotonicity of $\mathcal{T}^{\pi_k}$, and the last is the finite-horizon version of Bellman equation. Now, invoking the finite-horizon variant of Lemma 6 (c.f. Eq. (4)) with $f_{K:1}$ playing the role of f, the first term on the RHS vanishes due to greediness of $\pi_{K:1}$ w.r.t. $f_{K:1}^-$, and the third term vanishes due to pessimism. Therefore, we are only left with the second term, that

$$J_K(\pi_{\text{cp}})-J_K(\pi_{K:1})\leq\sum_{t=0}^{K-1}\gamma^t\mathbb{E}_{d_t^{\pi_{\text{cp}}}}[\mathcal{T}f_{K-t-1}^- - f_{K-t}^-].$$

We can bound the Bellman error inside expectation using uncertainty quantification again:

$$\mathcal{T}f_{K-t-1}^-=f_{K-t}^\star\leq\hat{f}_{K-t}+b=f_{K-t}^-+2b,$$

which yields [45]

$$J_K(\pi_{\text{cp}})-J_K(\pi_{K:1}) \leq 2\textstyle\sum_{t=0}^{K-1}\gamma^t\mathbb{E}_{(s,a)\sim d_t^{\pi_{\text{cp}}}}[b(s,a)]. \tag{30}$$

Expressivity Assumptions and Guarantees The guarantee we obtain above depends on $b(s,a)$, i.e., the tightness of the bonus term. Next we consider concrete settings where $b(s,a)$ can be explicitly computed and the RHS of Eq.(30) can be further bounded by expressions comparable to results in previous sections.

First of all, just like FQI/FQE, we will need assumptions to ensure that each regression we solve is realizable. Note that Bellman completeness for $\mathcal{T}$ ($\mathcal{T}f\in\mathcal{F},\forall f\in\mathcal{F}$) is *insufficient* here, because we are not backing up functions from $\mathcal{F}$, but $f_k^-=\hat{f}_k-b$, which can be outside the function class depending on the form $b(s,a)$ takes. In the literature, it is often assumed that $\mathcal{T}f\in\mathcal{F},\forall f\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ to avoid the problem, as illustrated in Figure 1.

Second, we will need pointwise confidence interval for our prediction. A canonical setting where this is feasible is linear regression: when $\mathcal{F}$ is the linear class induced by feature map ϕ, $\mathcal{F}_\phi=\{\langle\phi,\theta\rangle:\theta\in\mathbb{R}^d\}$, we can simply run (ridge) linear regression for the point estimate $\hat{f}_k$, with a quadratic uncertainty term:

$$b(s,a)=\frac{\beta}{\sqrt{n}}\sqrt{\phi(s,a)^\top(\Sigma_{\mathcal{D}}^{\text{ridge}})^{-1}\phi(s,a)}, \tag{31}$$

where β may depend on quantities such as d and horizon. $\Sigma_{\mathcal{D}}^{\text{ridge}}=\frac{1}{n}(\sum_{(s,a)\in\mathcal{D}}\phi(s,a)\phi(s,a)^\top+I)$.

As a remark, linear MDPs in Example 1 satisfies all conditions needed above. In fact, they are *equivalent*:

Proposition 8 *If $\mathcal{F}=\mathcal{F}_\phi$ is the linear class induced by ϕ and $\mathcal{T}f\in\mathcal{F},\forall f\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, then the MDP must be a linear MDP with $\phi^\star=\phi$ as its features.*[20]

[20] Proposition 2 of [99] claims a very similar result but replaces $\mathcal{T}f\in\mathcal{F},\forall f\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ with Bellman completeness $\mathcal{T}f\in\mathcal{F},\forall f\in\mathcal{F}$.

One can compare the linear MDP setting with the "representation learning in low-rank MDP" setting in Example 1, where $\phi^\star$ is unknown and must be learned from some class Φ. PEVI does not apply to the latter setting due to the lack of linearity of the concatenated class $\bigcup_\phi \mathcal{F}_\phi$, despite that the setting is information-theoretically tractable and can be handled by PSPI subject to the efficiency of the $f^\pi_{\min}$ oracle. That said, when there is also a class for realizing $\psi^\star$, model-based algorithms can also handle the setting and some use quadratic bonuses similar to PEVI [85].

Comparison of coverage Plugging Eq.(31) into Eq.(30), we have $J_K(\pi_{\mathrm{cp}}) - J_K(\pi_{K:1})$ bounded by $\frac{2\beta}{(1-\gamma)\sqrt{n}}$ times the following quantity:

$$(1-\gamma)\sum_{t=0}^{K-1}\gamma^t \mathbb{E}_{(s,a)\sim d_t^{\pi_{\mathrm{cp}}}}[\sqrt{\phi(s,a)^\top(\Sigma_{\mathcal{D}}^{\mathrm{ridge}})^{-1}\phi(s,a)}].$$

This expression plays the role of coverage in PEVI. Up to some subtle differences, its square is roughly

$$\left(\mathbb{E}_{(s,a)\sim d^{\pi_{\mathrm{cp}}}}[\sqrt{\phi(s,a)^\top \Sigma_D^{-1}\phi(s,a)}]\right)^2, \tag{32}$$

which is very close to $\sigma_{\max}(\Sigma_\pi^{1/2}\Sigma_D^{-1}\Sigma_\pi^{1/2})$, the upper bound of $C^{\mathrm{sq}}_{\pi_{\mathrm{cp}}}(\mathcal{F}_{\phi^\star})$ given in Eq.(20), Section 3.3. In fact, if we move the square-root of Eq.(32) outside, the expression will become $\mathrm{tr}(\Sigma_\pi^{1/2}\Sigma_D^{-1}\Sigma_\pi^{1/2})$, and trace is no smaller than largest eigenvalue $\sigma_{\max}$ and can be a factor of d larger in the worst case. On the other hand, having square-root inside makes Eq.(32) smaller than $\mathrm{tr}(\Sigma_\pi^{1/2}\Sigma_D^{-1}\Sigma_\pi^{1/2})$, so a definitive relation to Eq.(20) is unclear. That said, what is clear is that PEVI's guarantee requires Σ_D to cover all directions hit by Σ_π and does not apply when Σ_D only covers $\mathbb{E}_{d^\pi}[\phi]$ (this corresponds to $C^{\mathrm{sq}}_{\pi_{\mathrm{cp}}}$ in Eq. (22)), which is a weaker condition that still enables Theorem 7 and PSPI's guarantee.

4.5 Pessimism in Deep RL

The two algorithms, PSPI and PEVI, are both computationally efficient in linear settings. One may wonder whether they can be implemented with practical function approximation schemes, such as deep neural networks. We briefly review the theory-practice gaps below:

PSPI The problem with PSPI is that the oracle in Eq.(27) can be difficult to implement due to its minimax nature. Generally, most algorithms that attempt at minimizing Bellman errors need minimax optimization to tackle the double sampling problem (we will see other examples in Section 6), and there have been empirical attempts at implementing them with deep neural networks. Despite some reported successes [20, 66], the minimax can be difficult to tune and has very different optimization properties compared to dynamic-programming (DP) algorithms; the latter have received extensive study and empirical experimentation since the early days of deep RL [58], and various practical tricks and improvements have contributed to their reliability and stability [50]. It is possible that further empirical research in BRM-type algorithms may enable a more faithful implementation of version-space-based pessimistic algorithms such as PSPI.

In practice, empirical implementation of PSPI [17] still uses DP/TD-style updates for value estimation. This, however, breaks the theoretical guarantees: in finite-horizon problems, DP is bottom-up and freezes the value functions at later time stages before fitting earlier stages, which makes it difficult for pessimism at the initial time step to influence value estimation in later time steps.

Another problem with PSPI is in its NPG policy update, and the requirement of storing all previous iterates of value functions can be a significant burden in practice. Furthermore, the NPG step does not allow for a standalone policy class (see more discussion at the end of this subsection) and cannot scale to large action spaces computationally.

PEVI Unlike PSPI, PEVI is designed to be a DP algorithm, and we can easily swap out linear regression for regression with neural-nets. The problem is the bonus term that performs pointwise uncertainty quantification, which can be difficult to obtain beyond the simple linear case. A common heuristic is to do regression with the neural-net, and use the last-layer feature as ϕ to compute the quadratic bonus [8], which again breaks the theoretical guarantee. On the other hand, PEVI can directly benefit from better uncertainty quantification for standard regression problems.

Behavior Regularization Empirically, it is also common to take a standard DP algorithm and add *behavior regularization*, often in the form of a regularization term $\log \pi(a \mid s)/\pi_D(a \mid s)$. This corresponds to the KL divergence between the action distributions of the learned policy π and behavior policy π_D [29], as a way to encourage π to stay within the offline data distribution. However, restricting $\pi(a \mid s)/\pi_D(a \mid s)$ can only control $\prod_{t=1}^H \pi(a_t \mid s_t)/\pi_D(a_t \mid s_t)$, which is the notion of coverage used by IS (Section 2.1). Furthermore, $\log(\cdot)$ is a very weak regularizer as it allows the term inside to be exponentially large, and recent works in RLHF for large language models also show that such terms alone are insufficient for preventing out-of-distribution policies (see Section 4.2 of [100]).

This is incorrect, as pointed out in Proposition 3 of [104]: Bellman completeness + linear $\mathcal{F}$ cannot imply linear MDP. A counterexample is bisimulation: given an arbitrarily complex transition P, if we let $R \equiv 0$, then aggregating all states together is a perfect bisimulation abstraction, and the corresponding linear class (see Section 5.2) has dimension $|\mathcal{A}|$ and is Bellman complete [15], but P is still unrestricted and may not have a low-rank factorization.

Standalone Policy Class Empirically, offline RL algorithms are often tested on control problems with continuous action spaces. In these problems, it is natural to have a standalone policy class Π (instead of having it induced from $\mathcal{F}$), often consisting of stochastic policies. While the information-theoretic algorithm of maximizing $J_{\mathrm{VS}^-}(\pi)$ can handle this setting, both PSPI and PEVI crucially rely on being able to optimize the policy's action distribution separately for each state. The challenge with optimizing over a standalone Π is that we will inevitably face trade-offs among optimization errors in different states. Roughly speaking, we will need to control $E_{d^{\pi_{\mathrm{cp}}}}[f(s,\pi_{\mathrm{cp}}) - f(s,\pi)]$ (Lemma 6) where f is the pessimistic value function, but $d^{\pi_{\mathrm{cp}}}$ is unknown. We may minimize the function under d^D by $\arg\max_\pi \mathbb{E}_{d^D}[f(s,\pi)]$, but $f_k^-(s,\pi_{\mathrm{cp}}) - f_k^-(s,\pi)$ is not a one-sided error and can be negative in some states while positive in others. As a consequence, Lemma 3 does not apply and error under d^D may not translate to that under $d^{\pi_{\mathrm{cp}}}$ even with coverage assumptions. How to handle standalone policy classes in a computationally efficient manner is still an open problem.

4.6 Model-based RL and Further Discussions

We conclude this section by briefly introducing model-based algorithms. The model-based formulation will also make it easy for further discussions on pessimism.

Model-based Algorithms In RL, "model-based" refers to explicitly learning and using the dynamics of the MDP in the task, and by this standard all algorithms we introduced so far are model-free. Below we sketch a typical algorithmic framework for model-based offline RL, and we will see that it can be analyzed using the tools we have introduced so far.

In theoretical analyses, the log-loss as in Maximum Likelihood Estimation[21] is often considered for model learning:[22] given a class of candidate dynamics $\mathcal{P}$, the loss for $\widetilde{P} \in \mathcal{P}$ is

$$\text{log-loss}_{\mathcal{D}}(\widetilde{P}) = 1/|\mathcal{D}| \sum_{(s,a,s')\in\mathcal{D}} -\log \widetilde{P}(s' \mid s,a).$$

Concentration inequalities for MLE [109, 2] show that the minimizer of log-loss, $\widehat{P}$, satisfies

$$\mathbb{E}_{(s,a)\sim d^D}[\|P(\cdot \mid s,a) - \widehat{P}(\cdot \mid s,a)\|_1^2] \lesssim \frac{\log(|\mathcal{P}|/\delta)}{n}.$$

Now, if we output $Q^\pi_{\widehat{P}}$, the Q-function of π in the MDP defined by $(R,\widehat{P})$, we can show that it has bounded Bellman error w.r.t. the true model:

$$\begin{aligned}
&|Q^\pi_{\widehat{P}}(s,a) - (\mathcal{T}^\pi Q^\pi_{\widehat{P}})(s,a)| \\
&= |(\mathcal{T}^\pi_{\widehat{P}} Q^\pi_{\widehat{P}})(s,a) - (\mathcal{T}^\pi Q^\pi_{\widehat{P}})(s,a)| \\
&= |R(s,a) + \gamma\langle \widehat{P}(\cdot \mid s,a), V^\pi_{\widehat{P}}\rangle \\
&\qquad - (R(s,a) + \gamma\langle P(\cdot \mid s,a), V^\pi_{\widehat{P}}\rangle)| \\
&= |\gamma\langle \widehat{P}(\cdot \mid s,a) - P(\cdot \mid s,a), V^\pi_{\widehat{P}}\rangle| \\
&\le \gamma \|\widehat{P}(\cdot \mid s,a) - P(\cdot \mid s,a)\|_1 V_{\max}.
\end{aligned}$$

The first step follows from that $Q^\pi_{\widehat{P}}$ satisfies the Bellman equation in $(R,\widehat{P})$, and the last step follows from Hölder's inequality. Now, taking expectation over d^D of the squared Bellman error and plugging in the earlier concentration result, we have

$$\mathbb{E}_{d^D}[(Q^\pi_{\widehat{P}} - \mathcal{T}^\pi Q^\pi_{\widehat{P}})^2] \lesssim V_{\max} \frac{\log(|\mathcal{P}|/\delta)}{n}.$$

The LHS is precisely $\mathcal{E}(Q^\pi_{\widehat{P}};\pi)$, so the guarantee is exactly analogous to Eq.(12) for BRM estimation of Q^π, and the subsequent analysis in Section 3.2 can be applied as-is here.

Similarly, the information-theoretic pessimistic algorithm in Section 4.2 also has its model-based counterpart. Define the version space of models as:

$$\mathcal{P}_{\mathcal{D}} := \{\widetilde{P} \in \mathcal{P} : \text{log-loss}_{\mathcal{D}}(\widetilde{P}) \le_\epsilon \text{log-loss}_{\mathcal{D}}(\widehat{P})\},$$

where $\le_\epsilon$ is $\le$ up to a statistical threshold term similar to ϵ_0 in Eq.(25). The loss of MLE model $\widehat{P}$ is playing a similar role to the variance correction term (Eq.(9)) for value-based learning, making sure that we do not eliminate the true dynamics P and only keep around candidate models whose excess risk is small on the data distribution. Then, we have the model-based analogue of $\arg\max_{\pi\in\Pi} J^-_{\mathrm{VS}}(\pi)$:

$$\arg\max_{\pi\in\Pi} \min_{\widetilde{P}\in\mathcal{P}_{\mathcal{D}}} J_{\widetilde{P}}(\pi), \tag{33}$$

which enjoys a similar guarantee to Theorem 7. An interesting difference is that the model-based guarantee do not need to separately pay for $\log|\Pi|$, as the model estimation is independent of the policies.

Return, Regret, and General Objectives In most areas of RL, the following two objectives are equivalent performance measures:

$$\text{Return: } J(\pi) \quad \text{vs.} \quad \text{Regret: } J(\pi^\star) - J(\pi),$$

where $\pi^\star$ is the optimal policy in the MDP. This is because maximizing return is the same as minimizing regret, as they only differ by a negative sign and a constant shift. A perhaps surprising fact is that they become *different* when pessimism is involved: for example, consider

$$\arg\max_{\pi\in\Pi} \min_{\widetilde{P}\in\mathcal{P}_{\mathcal{D}}} (J_{\widetilde{P}}(\pi^\star_{\widetilde{P}}) - J_{\widetilde{P}}(\pi)).$$

[21] One caveat of this formulation is that practical model-based algorithms in deep RL seldom use log-loss [33, 102, 49, 14]; see [38] for further discussion.

[22] We assume reward function is known, as learning reward only involves a standard regression and can be easily incorporated.

This will be a policy that minimizes *worst-case* regret, which will be generally different from Eq.(33) that maximizes *worst-case* return; see Figure 3 for a concrete bandit example. These are design choices that we must make to reflect our preferences, and we should generally analyze an algorithm under the performance measure it uses (see e.g., [14]'s discussion of [92]).

In fact, these are not the only choices. One can also consider a "relative" return [14]:

$$J(\pi) - J(\pi_{\text{ref}}), \tag{34}$$

which leads to different behaviors with different π_{ref}. (Note that regret is not a special case of this, because here π_{ref} is fixed but $\pi^\star_{\widetilde{P}}$ in the definition of worst-case regret changes with $\widetilde{P} \in \mathcal{P}$.) [14] shows that using Eq.(34) to replace return in Eq.(33) achieves similar guarantees under the additional assumption that π_{ref} is covered. It also comes with the additional benefit that the learned policy is never worse than π_{ref} as long as $P \in \mathcal{P}_{\mathcal{D}}$, and this non-degenerate property can be desirable in industrial applications. Another benefit is that optimizing a difference can be numerically more stable than optimizing an absolute quantity. Furthermore, when $\pi_{\text{ref}} = \pi_D$, the coverage of π_{ref} is automatically satisfied ($C_{\pi_D} = 1$ if $d^D = d^{\pi_D}$), and the algorithm also has a model-free version by transforming Eq.(34) with the PD lemma (Ft 15) [17].

Connection to Robust MDPs The computational problem in Eq.(33) takes the form of the robust MDP problem [91], generally written as

$$\arg\max_{\pi\in\Pi} \min_{M\in\mathcal{M}} J_M(\pi),$$

where $\mathcal{M}$ is called the uncertainty set. Robust MDP (RMDP) is concerned with approximating this problem in a computationally efficient manner, and the performance of an algorithm output $\hat{\pi}$ is evaluated by the gap between $\min_{M\in\mathcal{M}} J_M(\hat{\pi})$ and the optimal objective value of the robust MDP problem. Solving the problem computationally often involves computing value functions and perform (robust) Bellman updates, which are similar to the model-free pessimistic algorithms (like PEVI).

The RMDP literature is often motivated by the fact that the true model dynamics are unknown and data-driven estimation has uncertainties, which coincides with the setting of offline RL. The difference is that RMDP assumes that the uncertainty set $\mathcal{M}$ is given as input and focus on the computational problem of maximizing pessimistic return, whereas offline RL takes the formation of uncertainty set as a component of the end-to-end problem and provide guarantees under coverage conditions.

In terms of problem settings, RMDPs typically consider the tabular setting (with some recent exceptions [115]), where the number of states is small and $\mathcal{M}$ is factored, in the sense that the uncertainty for transition and reward is independent across the state-action space. This factorization structure is similar to the pointwise uncertainty quantification in Section 4.4, and enables classical algorithms such as robust VI which PEVI resembles [91].

An interesting possibility is to apply PSPI-type algorithms for RMDPs with large state spaces and arbitrary uncertainty set, when we are given the pessimistic oracle $M(\pi) := \arg\min_{M\in\mathcal{M}} J_M(\pi)$, which is sometimes also considered in RMDPs.[23] However, a straightforward adaptation shows that PSPI only enjoys guarantees when $\mathcal{M}$ only has reward uncertainty: let $\pi^\star_{\text{rb}}$ be the optimal robust policy and $\hat{\pi}$ be the output of PSPI (the uniform mixture over iterates $\pi_{1:K}$),

$$\begin{aligned}
&J_{M(\pi^\star_{\text{rb}})}(\pi^\star_{\text{rb}}) - J_{M(\hat{\pi})}(\hat{\pi}) \\
&= J_{M(\pi^\star_{\text{rb}})}(\pi^\star_{\text{rb}}) - \tfrac{1}{K}\textstyle\sum_{k=1}^K J_{M(\hat{\pi})}(\pi_k) \\
&\le \textstyle\sum_{k=1}^K \left(J_{M(\pi_k)}(\pi^\star_{\text{rb}}) - J_{M(\pi_k)}(\pi_k)\right) \\
&= \tfrac{1}{K(1-\gamma)} \textstyle\sum_{k=1}^K \mathbb{E}_{d^{\pi^\star_{\text{rb}}}_{M(\pi_k)}}[Q^{\pi_k}_{M(\pi_k)}(s,\pi^\star_{\text{rb}}) - Q^{\pi_k}_{M(\pi_k)}(s,\pi_k)].
\end{aligned}$$

If all $M \in \mathcal{M}$ share the same transition dynamics, $d^{\pi^\star_{\text{rb}}}_{M(\pi_k)}$ will be the same for all k, and we can push $\sum_{k=1}^K$ inside the expectation and the rest of the analysis will be the same as PSPI. However, this is not doable in more general settings with transition uncertainty.[24] Similar difficulties are encountered when PSPI-like algorithms are applied to offline 2-player 0-sum Markov games [110]. How to reconcile this inapplicability with the intimate relation between RMDP and offline RL, as well as how to overcome this difficulty, will be interesting directions for future investigation.

5. LEARNING UNDER REALIZABILITY AND VALUE-FUNCTION SELECTION

Value-function estimation has been a core component to all previous sections, where we learn Q^π or $Q^\star$ from a potentially large and rich class $\mathcal{F}$.

Now consider a "simpler" version of the problem:

[23] What we really need is $Q^\pi_{M(\pi)}$, which is similar to the $f^\pi_{\min}$ oracle in Eq.(27).

[24] Another way that may circumvent the issue is to assume strong coverage assumption on the initial distribution, which is a common condition in policy gradient analyses [3]; c.f. Assumption 6 in [115] and how it is used in the proofs of their Theorems 8 and 9.

Value-function Selection
Input: $f_1, \dots, f_m \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$.
Output: f_i such that $f_i \approx Q^\pi$.

Despite its deceptive simplicity, **none of the methods so far are applicable even with $m = 2$ and one of f_1, f_2 is exactly Q^π,** and algorithms like BRM require a separate helper class ($\mathcal{G}$ in Section 3.2) that realizes $\{\mathcal{T} f_i\}_{i=1}^m$ to handle double sampling; the situation is similar for $Q^\star$. This means even with good data coverage (that is, we temporarily put away consideration of pessimism), we cannot perform hyperparameter tuning and model selection for FQI (e.g., selecting neural architecture). If we use FQE to estimate the performance of trained policies, FQE's own hyperparameters will be left untuned, and so on.

The other side of the same coin is that no estimators so far work under straight realizability, $Q^\pi \in \mathcal{F}$; had we had such an algorithm, applying it to $\mathcal{F} = \{f_i\}_{i=1}^m$ would solve the selection problem. Indeed, we can make progress on this difficult problem by switching back and forth between the two views: the *learning view*, where $\mathcal{F}$ is large and possibly structured, and the *selection view*, where $\mathcal{F} = \{f_1, \dots, f_m\}$ is unstructured but has a small size. Studying this issue also allows us to visit several less known ideas in the literature.

5.1 The Non-expansive Projection Argument

We start with the learning view, which asks the question: can we learn Q^π from $\mathcal{F}$ under realizability, without assuming Bellman completeness? To answer the question we revisit the counterexample in Proposition 1: when there is infinite data, each iteration of FQE becomes

$$f_k \leftarrow \mathrm{Proj}_{\mathcal{F}} \mathcal{T}^\pi f_{k-1}, \tag{35}$$

where $\mathrm{Proj}_{\mathcal{F}} f := \arg\min_{f' \in \mathcal{F}} \|f - f'\|_{2,d^D}$ and corresponds to linear regression for linear $\mathcal{F}$. To simplify presentation, in this section we assume d^D is supported on the entire $\mathcal{S} \times \mathcal{A}$, so that $\mathrm{Proj}_{\mathcal{F}}$ is unique for linear $\mathcal{F}$.

Without the projection step, Eq.(35) is textbook value-iteration, which enjoys convergence based on the ℓ_∞ contraction of $\mathcal{T}^\pi$:

$$\|f_k - Q^\pi\|_\infty = \|\mathcal{T}^\pi f_{k-1} - \mathcal{T}^\pi Q^\pi\|_\infty \le \gamma \|f_{k-1} - Q^\pi\|_\infty.$$

Therefore, the divergence in Proposition 1 can be attributed to $\mathrm{Proj}_{\mathcal{F}}$ destroying the contraction of $\mathcal{T}^\pi$.

So a natural idea for "fixing" the analysis is to make sure that $\mathrm{Proj}_{\mathcal{F}}$ is *non-expansive*, that for any f, f', $\|\mathrm{Proj}_{\mathcal{F}} f - \mathrm{Proj}_{\mathcal{F}} f'\| \le \|f - f'\|$ for some appropriate norm $\|\cdot\|$ (see [13], Assumption 6.3), with the hope that composing a non-expansion with a contraction still yields a contraction. In fact, $\mathrm{Proj}_{\mathcal{F}}$ *is* non-expansive when $\mathcal{F}$ is linear and $\|\cdot\| = \|\cdot\|_{2,d^D}$. Then why can the divergence in Proposition 1 still happen to linear $\mathcal{F}$?

The answer is that $\mathcal{T}^\pi$ is contraction under ℓ_∞ and $\mathrm{Proj}_{\mathcal{F}}$ is non-expansion under $\|\cdot\|_{2,d^D}$, and the mismatch between norms prevents us from combining their contraction/non-expansion properties. That said, $\mathcal{T}^\pi$ is also a contraction under weighted 2-norm, but under a special distribution: for any $f, f' \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ and a distribution $\nu \in \Delta(\mathcal{S} \times \mathcal{A})$ to be determined later,

$$\begin{aligned}
&\|\mathcal{T}^\pi f - \mathcal{T}^\pi f'\|_{2,\nu}^2 \\
&= \gamma^2 \mathbb{E}_{(s,a)\sim d^D}\big[\big(\mathbb{E}_{s'\sim P(\cdot|s,a)}[f(s',\pi)] \\
&\quad - \mathbb{E}_{s'\sim P(\cdot|s,a)}[f'(s',\pi)]\big)^2\big] \\
&\le \gamma^2 \mathbb{E}_{\substack{(s,a)\sim d^D, \\ s'\sim P(\cdot|s,a), a'\sim\pi(\cdot|s)}}[(f - f')^2] \\
&=: \gamma^2 \|f - f'\|_{2,P(\nu)\times\pi}^2.
\end{aligned}$$

We start with ν but obtain a different distribution at the end, $P(\nu) \times \pi$. To make this a contraction property, let $P(\nu) \times \pi = \nu$, i.e., ν is an *invariant distribution* w.r.t. policy π. To further allow combination with $\mathrm{Proj}_{\mathcal{F}}$'s non-expansion, we need to let $d^D = \nu$; this is an "on-policy" assumption,[25] that data is sampled from the target policy we want to evaluate, beating the basic premise of OPE.

5.2 State Abstractions

To mitigate the mismatch in norms, instead of having $\mathcal{T}^\pi$ to be a contraction under weighted 2-norm, we can instead ask $\mathrm{Proj}_{\mathcal{F}}$ to be a non-expansion under ℓ_∞ to align with the ℓ_∞ contraction of $\mathcal{T}^\pi$. [31] shows that this is indeed the case when $\mathcal{F}$ is *piecewise constant*. Let $\mathcal{F}$ be a linear class with feature $\phi \in \mathbb{R}^d$. We say $\mathcal{F}$ is piecewise constant if ϕ is "1-hot", that $\phi(s,a)$ equals 1 in exactly one of the coordinates and 0 elsewhere. In this case, we indeed have $\|\mathrm{Proj}_{\mathcal{F}} f - \mathrm{Proj}_{\mathcal{F}} f'\|_\infty \le \|f - f'\|_\infty$.

Piecewise-constant $\mathcal{F}$ is closely related to the literature of state abstractions (or aggregations) [55].[26] A large part of this literature focuses on bisimulation, which satisfies Bellman completeness; thus, its theoretical understanding is largely subsumed by the more general analyses we have seen in Section 3.3. In comparison, the fact that abstractions enjoy guarantees under realizability alone[27] is what truly makes them different from other function-approximation schemes.

The above analysis is under ℓ_∞, which is incompatible with learning in large state spaces. To have a more

[25] The invariant-distribution assumption here is stronger than the typical on-policy condition. To satisfy it with trajectory data (Section 2.1), we need d_0 to be such an invariant distribution to start with. In comparison, "on-policy" IS (i.e., $\pi_D = \pi$ and $C_{\mathcal{A}} = 1$) reduces to Monte-Carlo policy evaluation and can work with arbitrary d_0.

[26] Strictly speaking, 1-hot ϕ induces a partition over $\mathcal{S} \times \mathcal{A}$, which may not correspond to a well-defined partition over $\mathcal{S}$; we ignore this subtle difference in the discussion.

[27] This is known as Q^π- or $Q^\star$-irrelevant abstractions [55].

distribution-aware analysis similar to Section 3.3, we define an ϕ-aggregated MDP, M_ϕ with transition

$$P_\phi(s' \mid s,a) = \sum_{\widetilde{s},\widetilde{a}:\phi(\widetilde{s},\widetilde{a})=\phi(s,a)} d_D^\phi(\widetilde{s},\widetilde{a}) P(s' \mid \widetilde{s},\widetilde{a}).$$

and reward R_ϕ is defined similarly. Here $d_D^\phi(s,a) \propto d^D(s,a)$ but is normalized within a partition (that is, all (s,a) pairs that share the same $\phi(s,a)$). Two key properties: (1) $Q^\pi = Q^\pi_{M_\phi}$, that M_ϕ preserves the true Q^π, and (2) $\mathrm{Proj}_{\mathcal{F}} \mathcal{T}^\pi f = \mathcal{T}^\pi_{M_\phi} f$, so projected Bellman update is essentially the true Bellman update in M_ϕ. Then we can essentially carry out an analysis similar to Section 3.3 but in M_ϕ. This results in coverage assumptions that d_D covers $d^\pi_{M_\phi}$, where the discounted occupancy d^π is defined w.r.t. the dynamics of M_ϕ [96, 37].

5.3 BVFT

The result in Section 5.2 removes Bellman-completeness, at the cost of imposing piecewise constant $\mathcal{F}$, exhibiting an expressivity-structure trade-off. One can argue though that the result is not too useful for either the "learning view" (piecewise-constant too restrictive) or the "selection view" (the result does not seem to help with the selection problem).

We now show that the benefit of strong structures can be "lifted" to unstructured $\mathcal{F}$ *for free*, at least information-theoretically. For this we must switch to the selection view, and consider the minimal problem posed at the beginning of Section 5: $m = 2$ and $Q^\pi \in \{f_1, f_2\}$. Perhaps surprisingly, only using these pieces of information, we can already create a piecewise-constant $\mathcal{F}$ with bounded dimension (i.e., the size of the partition, which determines the statistical capacity of $\mathcal{F}$), without making further assumptions or leveraging additional side information.

To describe the construction, it suffices to give the "group identifier" for each (s,a), which specifies the partition that will induce $\mathcal{F}$. To do so, we first discretize the value range $[0, V_{\max}]$ to a regular grid with grid size ϵ, and discretize $f_1(s,a)$ and $f_2(s,a)$ accordingly, denoted as $\bar{f}_1(s,a), \bar{f}_2(s,a)$. This pair of integer is the group identifier, i.e., (s,a) and (s',a') are aggregated if $\bar{f}_1(s,a) = \bar{f}_1(s',a')$ and $\bar{f}_2(s,a) = \bar{f}_2(s',a')$. The resulting piecewise constant $\mathcal{F}$ satisfies $\bar{f}_1, \bar{f}_2 \in \mathcal{F}$, which means $Q^\pi \in_\epsilon \mathcal{F}$ up to a small discretization error. On the other hand, the dimensionality of $\mathcal{F}$ (the number of groups) is $O((V_{\max}/\epsilon)^2)$, independent of the size of the state-action space.

The algorithm, known as BVFT [96],[28] also extends to selecting m functions via a tournament procedure of pairwise comparisons, and has been empirically tested [108]. The error bound has $\log m$ dependence, which means a $\log|\mathcal{F}|$ information-theoretic result for the learning setting, though the computation seems not friendly to optimization.

Hardness Results against $Q^\pi \in \mathcal{F}$ + Bounded C_π The analysis of BVFT inherits the coverage assumption from Section 5.2, that we need bounded $d^\pi_{M_\phi}/d^D$ for all partitions ϕ created in the pairwise process, which is a complicated assumption. Ideally, the cleanest assumptions would be $Q^\pi \in \mathcal{F}$ and the standard coverage parameter C_π. The $Q^\star$ counterpart of this result was conjectured to be impossible [15], though subsequent negative results were w.r.t. different and often weaker data assumptions [89, 5, 104]. The gap was eventually closed by [28] (see also [37]) which confirmed the earlier conjecture, that accurate estimation of $J(\pi)$ under $Q^\pi \in \mathcal{F}$ and bounded C_π is information-theoretically intractable.

5.4 LSTDQ

Besides state abstractions, there is another setting where learning Q^π only requires realizability: the LSTD algorithms. LSTD is derived by taking a TD-style algorithm with linear function approximation and directly writing down its fixed point. For example, if we take FQE with a linear $\mathcal{F}$, its fixed point, LSTDQ, is $f_\theta(s,a) = \phi(s,a)^\top\theta$, with $\theta = A^{-1}B$ and

$$A = \Sigma_D - \gamma \mathbb{E}_D[\phi(s,a)\phi(s',\pi)^\top], B = \mathbb{E}_D[\phi(s,a)\cdot r].$$

Here A and B are both population statistics, and the actual algorithm uses their finite-sample estimation $\widehat{A}$, $\widehat{B}$ to compute $\widehat{\theta}$. Note that the algorithm is only well-defined when A is invertible, and numerically stable when $\sigma_{\min}(A)$ is bounded away from 0 (which guarantees boundedness of $\hat{\theta}$); here $\sigma_{\min}$ is the smallest singular value since A is not necessarily symmetric. Despite that LSTD is inspired by TD algorithms, there are cases where TD diverges but LSTD outputs bounded solutions, and LSTD algorithms generally work under more lenient conditions [67].

Interestingly, many algorithms, such as BRM (Section 3.2) and MIS (Section 6), reduce to LSTDQ when using linear classes [6, 83, 93]. Hence, LSTDQ can also be analyzed under the assumptions made in those sections, which require additional expressivity assumptions other than $Q^\pi \in \mathcal{F}$.

But this is not necessary, and the simple linear-algebraic structure of LSTDQ admits an alternative condition. It turns out that all we need is $\sigma_{\min}(A)$! The idea is simple: (1) the linear coefficient of Q^π (i.e., $Q^\pi = \phi(s,a)^\top\theta^\pi$) is a solution to $A\theta = B$, and (2) if A is invertible ($\sigma_{\min}(A) > 0$), the solution is unique. In the finite-sample case, we can use standard concentration arguments to bound $\|A - \widehat{A}\|_2$ (operator norm), $\|B - \widehat{B}\|_2$, and

$$\|\theta^\pi - \widehat{\theta}\|_2 = \|A^{-1}A\theta^\pi - A^{-1}A\widehat{\theta}\|_2$$

[28] The original work learns $Q^\star$ using a similar argument.

$$= \|A^{-1}B - A^{-1}A\widehat{\theta}\|_2 \le \|B - A\widehat{\theta}\|_2/\sigma_{\min}(A)$$
$$= \|B - A\widehat{\theta} + \widehat{A}\widehat{\theta} - \widehat{B}\|_2/\sigma_{\min}(A)$$
$$\le (\|B - \widehat{B}\|_2 + \|A - \widehat{A}\|_2\|\widehat{\theta}\|_2)/\sigma_{\min}(A).$$

Coverage in LSTDQ All guarantees we have seen so far rely on three types of conditions: (1) expressivity of function class, (2) structure of function class, and (3) coverage of data. For LSTDQ, $Q^\pi \in \mathcal{F}$ is the expressivity condition, linear $\mathcal{F}$ is the structural condition, so the remaining $\sigma_{\min}(A)$ must correspond to coverage. Indeed, one can show that in the strictly on-policy case, that is, when d^D is an invariant distribution w.r.t. π (Section 5.1), $\sigma_{\min}(A)$ is positive and controlled by $(1-\gamma)$. Roughly speaking, this is because $\phi(s',\pi)$ has the same marginal distribution as $\phi(s,a)$,[29] so $\Sigma_D = \mathbb{E}_D[\phi(s,a)\phi(s,a)^\top]$ dominates the spectrum of $\mathbb{E}_D[\phi(s,a)\phi(s',\pi)^\top]$. See [52, 68] for more detailed analyses of LSTD in the on-policy case.

Unfortunately, our understanding of $\sigma_{\min}(A)$ outside the on-policy setting is very limited, and the parameter itself is not a very clean coverage parameter and mixes together many other factors. For example, $\sigma_{\min}(A)$ can be 0 if there are simply redundant features in ϕ. Another problem is that $\sigma_{\min}(A)$ is not scale-invariant: if we scale ϕ by a constant (and scale θ^π accordingly), $\sigma_{\min}(A)$ will also grow or shrink superficially. Even if we fix these issues, there is still very limited interpretability of the condition outside the on-policy case. This is a general theme with algorithms that only require realizability, that error propagation and coverage are conceptually much more complicated and hard to interpret.

As a final remark, just as BVFT "lifts" the benefit of state abstractions to unstructured function classes, a similar procedure can also lift LSTDQ, which produces an algorithm that is potentially simpler than BVFT for learning Q^π.

6. MARGINALIZED IMPORTANCE SAMPLING

Density ratio d^π/d^D has played an important role in the analyses of previous sections. In this section we show that they can also be explicitly learned by algorithms for OPE or even policy optimization. These algorithms, generally known as "marginalized importance samplling" (MIS), require different assumptions than the ones introduced previously and have complementary properties.

6.1 OPE via Value Functions

We start with learning value functions for OPE. Previous guarantees for value-based OPE all require Bellman completeness. Below we show an alternative algorithm and analyses that only require realizability. In addition to $Q^\pi \in \mathcal{F}$, we will also require a *weight function class* $\mathcal{W}$ such that $w^\pi \in \mathcal{W}$, where $w^\pi(s,a) = d^\pi(s,a)/d^D(s,a)$. (These assumptions will be relaxed later, but for now we use them for a clean derivation.) It starts from Lemma 2 and only takes a few steps [83, 39]:

$$|J_f(\pi) - J(\pi)| = \left|\frac{1}{1-\gamma}\mathbb{E}_{d^\pi}[f - \mathcal{T}^\pi f]\right|$$
$$= \left|\frac{1}{1-\gamma}\mathbb{E}_{d^D}[d^\pi/d^D \cdot (f - \mathcal{T}^\pi f)]\right|$$
$$\le \max_{w\in\mathcal{W}}\left|\frac{1}{1-\gamma}\mathbb{E}_{d^D}[w \cdot (f - \mathcal{T}^\pi f)]\right| =: \max_{w\in\mathcal{W}} L_q(w,f).$$

The MQL algorithm minimizes (the estimation of) the above loss over $f \in \mathcal{F}$ and estimate $J(\pi) \approx J_f(\pi)$ for the learned f. Here, since the ultimate goal is to make sure $J_f(\pi) \approx J(\pi)$, the derivation starts with their difference and hope to explicitly control it. Lemma 2 tells us that the difference is the *average Bellman error* under d^π; there are two key properties of this quantity $\mathbb{E}_{d^\pi}[f - \mathcal{T}^\pi f]$:

1. $f - \mathcal{T}^\pi f$ can be positive or negative, and the positive/negative errors can cancel with each other in a plain expectation $\mathbb{E}_{(.)}[f - \mathcal{T}^\pi f] = 0$.
2. d^π is an unknown distribution.

Previous algorithms, such as BRM, handle these challenges by squaring the Bellman error, which makes it one-sided (non-negative), so that controlling the (squared) error on d^D indirectly controls it on any covered distribution, which also solves the problem of unknown d^π. The cost is the double-sampling issue and the introduction of the completeness assumption.

In comparison, MIS handles it very differently. It does not square the error, and $L_q(w,f)$ is directly amendable to statistical estimation and free from the double-sampling issue: $L_q(w,f) =$

$$\left|\frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^D}[w(s,a)(f(s,a) - r - \gamma f(s',\pi))]\right|.$$

The problem is that we cannot just minimize $|\mathbb{E}_{d^D}[f - \mathcal{T}^\pi f]|$, as the lack of non-negativity does not allow error bound to translate from d^D to a covered distribution. The solution, as Lemma 2 suggests, is to use importance weights d^π/d^D to reweight the data distribution to be precisely d^π. Of course, d^π is unknown, so we reweight using a rich class of weight functions $w \in \mathcal{W}$, and take the worst-case reweighted error among them. The extra functions in $\mathcal{W}$ does not hurt, in the sense that $f = Q^\pi$ satisfies $f - \mathcal{T}^\pi f \equiv 0$ so $L_q(w, Q^\pi) \equiv 0, \forall w$.

A more formal guarantee is as follows:

Theorem 9 (Guarantee of MQL [83]) *Assume $w^\pi \in \mathcal{W}$ and $Q^\pi \in \mathcal{F}$. Let $\hat{f} := \arg\min_{f\in\mathcal{F}} \max_{w\in\mathcal{W}} \widehat{L}_q(w,f)$,*

[29] For stochastic policy we can replace $\phi(s',\pi)$ in the definition of A with $\phi(s',a')$, $a' \sim \pi(\cdot \mid s')$, and this claim will hold.

where $\widehat{L}_q(\cdot)$ is the empirical estimation of $L_q(\cdot)$. Then, w.p. $\geq 1-\delta$,

$$|J_{\hat{f}}(\pi) - J(\pi)| \lesssim \frac{V_{\max}\|\mathcal{W}\|_\infty}{1-\gamma}\sqrt{\frac{\log(|\mathcal{F}||\mathcal{W}|/\delta)}{n}},$$

where $\|\mathcal{W}\|_\infty := \max_{w\in\mathcal{W}}\|w\|_\infty$.

The guarantee is similar to Eq.(18) for BRM, except that the coverage parameter C_π seems missing from the bound. The reality is that C_π is hidden in $\|\mathcal{W}\|_\infty$, the boundedness of the $\mathcal{W}$ class, as $C_\pi = \|w^\pi\|_\infty \leq \|\mathcal{W}\|_\infty$. In fact, $\|\mathcal{W}\|_\infty$ will be generally greater than C_π if it includes w with $\|w\|_\infty > \|w^\pi\|_\infty$.

For now let's treat $\|\mathcal{W}\|_\infty \approx C_\pi$. Another difference is that Eq.(18) scales with $\sqrt{C_\pi}$, but Theorem 9 scales linearly with C_π. This is a looseness and can be tightened with some additional assumptions/pre-processing. Since $w \in \mathcal{W}$ models the importance weights, we can assume that $\mathbb{E}_{d^\mathcal{D}}[w] \approx 1$,[30] in which case $\mathbb{E}_{d^\mathcal{D}}[w^2] \leq \|w\|_\infty$ and Bernstein's inequality can give the square-root improvement just as in the analysis of IS .

Do We Really Need $w^\pi \in \mathcal{W}$? A somewhat common belief is that MIS has a disadvantage that it requires boundedness of C_π as its coverage parameter and cannot enjoy refined version such as C_π^{sq} or C_π^{avg}. On a related note, in structural models such as linear MDPs (Example 1), MIS is believed to be not applicable without a separate $\mathcal{W}$ class as we do not know the form of w^π and it may not be linear in $\phi^\star$.

This is actually not true. Inspecting the derivations, we see that what we really need is a function w^π_{eff} such that

$$\mathbb{E}_{d^\pi}[f - \mathcal{T}^\pi f] = \mathbb{E}_{d^\mathcal{D}}[w^\pi_{\text{eff}}(f - \mathcal{T}^\pi f)].$$

When $f - \mathcal{T}^\pi f$ is linear in some features ϕ (the algorithm does not need to know ϕ, so this is not imposing Bellman-completeness on $\mathcal{F}$), a sufficient condition is that [111]

$$\mathbb{E}_{d^\pi}[\phi] = \mathbb{E}_{d^\mathcal{D}}[w^\pi_{\text{eff}} \cdot \phi].$$

This is known as the *(kernel) mean matching* problem [32]. While $w^\pi_{\text{eff}} = w^\pi$ is a solution, it can have better-behaved solutions, such as

$$w^\pi_{\text{eff}}(s,a) = \phi(s,a)^\top \Sigma_D^{-1}\mathbb{E}_{d^\pi}[\phi].$$

Interestingly, the 2nd moment of w^π_{eff} (which controls the $\sqrt{n}$ term in Bernstein's inequality) is $\mathbb{E}_{d^\mathcal{D}}[(w^\pi_{\text{eff}})^2] = C_\pi^{\text{avg}}$! Furthermore, w^π_{eff} constructed above is linear in $\phi(s,a)$, which makes it directly applicable in linear MDPs. The slight disadvantage of MIS is that we also pay $\|w^\pi_{\text{eff}}\|_\infty$ in the $1/n$ term in Bernstein's, which can be mitigated by using median-of-means estimators [53].

As a final remark, due to the convexity of $L_q(w,f)$ in w, w^π (or w^π_{eff}) $\in \mathcal{W}$ can be relaxed to $w^\pi \in \text{conv}(\mathcal{W})$, where $\text{conv}(\cdot)$ is the convex hull [83].

[30] If not we can replace w with $w/\mathbb{E}_\mathcal{D}[w]$. See also Footnote 4.

Computational Efficiency Similar to BRM, the MIS estimator also requires minimax optimization. When the $\mathcal{W}$ class, which plays the role of a "discriminator", is an RKHS, one can show that the gradient of $\max_{w\in\mathcal{W}} L_q(w,f)^2$ w.r.t. f's parameters has a closed-form solution[31] [56, 26], and the objective can be effectively optimized by a single SGD over f. However, if we want to leverage neural-nets for $\mathcal{W}$, the optimization can become difficult and this contributes to MIS mehtods' lack of empirical popularity.

Uncertainty Quantification We can also derive valid upper and lower bounds on $J(\pi)$ similar to $J^-_{\text{vs}}(\pi)$ in Eq.(26). All we need to do is to replace the loss $\widehat{\mathcal{E}}(f;\pi)$ with $\max_{w\in\mathcal{W}} L_q(w,f)$ and set the appropriate statistical threshold [39, 27]. As long as $Q^\pi \in \mathcal{F}$, $\min J_f(\pi)$ over the version space is always a true lower bound, and it will be tight if we further have $w^\pi \in \mathcal{W}$. This immediately leads to pessimistic policy optimization algorithms that parallel those introduced in Sections 4.1 and 4.3 [39, 116]. For example, information-theoretic MIS algorithm can achieve a similar guarantee to Theorem 7, under the assumptions that $Q^\pi \in \mathcal{F}, \forall\pi \in \Pi$, and $w^{\pi_{\text{cp}}} \in \mathcal{W}$ [39].

6.2 OPE via Weight Functions

MQL learns a value function using marginalized importance weights w as "discriminators". However, their roles can be swapped: we can learn w^π using $\mathcal{F}$ as discriminators. In fact, MIS was first introduced in this form [56, 98] and the discovery of value learning methods and unification happened later [83, 26, 39].

The derivation of weight learning methods is similar. First, once we obtain $w^\pi = d^\pi/d^D$, we can estimate return as $J(\pi) = \mathbb{E}_{(s,a)\sim d^\pi}[R(s,a)]/(1-\gamma) = \mathbb{E}_{d^\mathcal{D}}[w^\pi \cdot r]/(1-\gamma) =: J_{w^\pi}(\pi)$, with a slight abuse of notation that $J_f(\pi)$ and $J_w(\pi)$ correspond to different expressions that depend on the nature of the subscript. Similar to Lemma 2, we now need a error decomposition lemma that translates the error of $J(\pi) - J_w(\pi)$ for an arbitrary w into some kind of one-step Bellman-like error:

Lemma 10 *Given any $w \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, $J(\pi) - J_w(\pi) =$*

$$\mathbb{E}_{s\sim d_0}[Q^\pi(s,\pi)] + \mathbb{E}_D[w(\gamma Q^\pi(s',\pi) - Q^\pi(s,a))]/(1-\gamma).$$

Just as the RHS of Lemma 2 contains the Bellman error w.r.t. $\mathcal{T}^\pi$ (which defines Q^π), the RHS of Lemma 10 is the violation of Bellman *flow* equation for d^π:[32] $d^\pi(s,a) = (1-\gamma)d_0(s,\pi) + \gamma\sum_{s',a'} d^\pi(s',a')P(s \mid s',a')\pi(a \mid s)$. In fact, when $w = w^\pi$, the RHS will always be 0 even if

[31] The square in $L_q(w,f)^2$ is outside the expectation and absolute value, so it does not change the statistical properties of the algorithm.

[32] Interestingly, Lemma 2's proof uses the Bellman flow equation for d^π, and Lemma 10's proof uses the Bellman equation for Q^π.

we replace Q^π with any other function f, which leads to the following loss: $L_w(w,f) :=$

$$\left|\mathbb{E}_{s\sim d_0}[f(s,\pi)] + \mathbb{E}_D[w\cdot(\gamma f(s',\pi) - f(s,a))]/(1-\gamma)\right|.$$

The MWL algorithm, $\arg\min_{w\in\mathcal{W}}\max_{f\in\mathcal{F}}\widehat{L}_w(w,f)$, has similar and symmetric properties w.r.t. MQL. For unification between the two class of methods and their duality, we refer the readers to [39, 63] for further reading.

Squared-loss Algorithms for Learning w^π We have seen algorithms that learn Q^π under Bellman completeness (FQE and BRM), and algorithms that model Q^π and w^π jointly (MIS). Naturally, there are also algorithms that learn w^π under a form of completeness w.r.t. the Bellman flow operator, both in DP style [35] and BRM style [62]. As notable difference between learning Q^π and w^π under the respective completeness assumptions, the existence of Q^π does not depend on the property of the offline data distribution d^D, but the existence of w^π does, so sometimes the learning target does not even exist for w^π estimation when the data lacks sufficient coverage. To handle this problem, [35] shows that for an arbitrary data distribution, one can always estimate a clipped version of w^π, which can also be used for pessimistic evaluation. Another benefit of learning w^π directly is that it allows the plug-in use of more general objective functions (and constraints) of the occupancy measure d^π, for which the expected return $J(\pi) = \mathbb{E}_{d^\pi}[R]/(1-\gamma)$ is a special case (linear objective) [103, 61].

6.3 Policy Optimization

The symmetry between value-function and weight learning is closely related to the Linear Programming (LP) duality [70]. In fact, it is well known that finding the optimal policy in an MDP can also be cast as an LP, where the saddle point of the primal-dual form corresponds to $d^{\pi^\star}$ and $V^\star(s) = Q^\star(s,\pi^\star)$, respectively, from which the optimal policy $\pi^\star$ can be extracted. This observation leads to the hope that instead of evaluating each policy and choosing the best among them, which requires assumptions like $Q^\pi \in \mathcal{F}, \forall\pi\in\Pi$, the LP for $\pi^\star$ may enable an MIS algorithm that directly solves $(d^{\pi^\star}, V^\star)$ without policy evaluation, and provide single-policy coverage guarantee under only two realizability assumptions: $d^{\pi^\star}/d^D \in \mathcal{W}$, $V^\star \in \mathcal{V}$. This is roughly what is proved by [107], except that strong regularization needs to be added to prevent degenerate solution in the function-approximation setting, which leads to a relatively slow rate of $1/\epsilon^6$ for the optimality guarantee. Improvement of the rate often comes with the cost of additional assumptions [16, 65, 116].

7. EMERGING DIRECTIONS AND DISCUSSIONS

We conclude the article by discussing emerging directions and challenges.

Connection to Online RL Online RL, which emphasizes efficient exploration and active data collection, is another core area of RL. Despite its largely parallel development to offline RL, more and more similarities and connections have been found. To start with, the function-approximation assumptions induced in this article are also highly relevant to online RL [41, 43]. Moreover, the central challenge in online RL is to *explore*, i.e., finding policies that visit states and actions *not* covered by existing data, which is often achieved by *optimistic* algorithms [7]. This exhibits an interesting symmetry with pessimism in offline RL. Indeed, the key behind both optimism and pessimism is uncertainty quantification, and both version-space and pointwise pessimism have their optimistic counterparts in online RL [41, 43, 44].

That said, the coverage conditions extensively discussed in this article seem a purely offline concept, since it depends on an offline data distribution which does not exist in the online setting. In online RL, what is needed is often structural assumptions on the dynamics (e.g., low-rankness in Example 1) [41]. While structural dynamics and coverage may seem unrelated, we have seen in Section 3.4 that the *existence* of d^D that allows all-policy coverage (low $\max_{\pi\in\Pi} C_\pi$) may imply restriction on the environment dynamics, leading to the suspicion that such a condition may be useful for online exploration. Indeed, recent work of [94] shows that *coverability*, defined as $\inf_{d^D}\max_{\pi\in\Pi} C_\pi$, enables sample-efficient online exploration under Bellman completeness under $\mathcal{T}$. The quantity is inspired by offline RL coverage, but is purely a structural property of the MDP dynamics and the policy class (note the $\inf_{d^D}$).

Besides connections on theoretical understanding and learnability conditions, there is also significant interest in exploring more "hybrid" learning protocols between online and offline RL, e.g., when online RL can benefit from some offline data [97, 88, 76] , and when offline RL can benefit from additional online experimentation budget [36].

Intersection with Deep Learning Theory Throughout this article we have been using naïve complexity measures for in-distribution generalization error bounds, the log cardinality of finite classes. While many analyses can be extended to handle infinite classes with bounded covering numbers, there are also significant challenges in incorporating more modern complexity measures, especially for deep neural networks. For example, when analyzing DP-based algorithms (e.g., FQI or its online variant, such as DQN [58]), we need to deal with the loss $\widehat{\mathcal{L}}(f';f,\pi)$. Fixing f and π, minimizing the loss is a standard regression problem, and analyses from deep learning theory can be directly adopted. However, unlike standard supervised learning where the labels are independent of each other, here the regression label is $r + \gamma f(s',\pi)$,

where f and π may be the outcome of previous iterations (e.g., $f = f_{k-1}$ in FQE/FQI) which depends on the entire dataset. Replacing union bound over $f \in \mathcal{F}, \pi \in \Pi$ with proper complexity measures has proved difficult. These issues are often circumvented by using fresh samples in each iteration [23], so that the labels can be viewed as independent for the current regression. [6] has addressed a related problem by proposing a new complexity measure called VC-crossing dimension, but the definition is somewhat restrictive and has not seen further development.

Multi-agent Setting Multi-agent RL (MARL) is an extension of standard (single-agent) RL, where there are multiple agents acting simultaneously with possibly misaligned or even competitive objectives.[33] Central to the setting are game-theoretic concepts such as equilibria. A perhaps surprising fact is that even for zero-sum two-player games, coverage of the equilibrium policy $(\mu^\star, \nu^\star)$ (which are policies for each player/agent, respectively) is insufficient for offline learning. A stronger and sufficient condition is called unilateral coverage, where data covers $(\mu^\star, \nu)$ and $(\mu, \nu^\star)$ for all μ and ν in each player's policy class [18, 19].

Information-theoretically, algorithms with guarantees in the general function-approximation settings have also been developed, which build on and extend the ideas behind version-space pessimism (Theorem 7). To find a (say) Nash equilibrium, a formulation amendable to learning and optimization is to minimize the equilibrium gap, e.g., $\mathrm{Gap}(\mu, \nu) = \max_{\mu^\dagger} J(\mu^\dagger, \nu) - \min_{\nu^\dagger} J(\mu, \nu^\dagger)$, which measures how much the players/agents want to deviate from a candidate solution. [110] shows that we can have a conservative estimate of the gap:

$$\mathrm{Gap}(\mu, \nu) \le \max_{\mu^\dagger} J^+_{\mathrm{VS}}(\mu^\dagger, \nu) - \min_{\nu^\dagger} J^-_{\mathrm{VS}}(\mu, \nu^\dagger),$$

where J^+_{VS} is similar to J^-_{VS} but we look for the most optimistic estimate. The estimation of these upper and lower bounds is no different from Section 4.1, since once the players' policies are fixed, the game-theoretic aspects simply disappear in the policy evaluation subproblem. Minimizing such a upper bound on the gap leads to provable guarantees under unilateral coverage.

Partial Observability Another extension of the standard MDP framework is to consider partial observability (non-Markovianity), often modelled as Partially Observable MDPs (POMDPs). POMDPs and the related PSRs formulations have been studied in the literature, but mostly from a computational perspective and often in the tabular setting. To handle large observation spaces, one idea is to view POMDPs as MDPs with history of observations and actions as the state representation; this perspective makes MDP algorithms and analyses directly applicable under appropriate function approximation over histories. However, the notion of coverage, especially that based on state density ratio C_π, becomes the probability ratio on *histories*, which is the exactly the cumulative importance weights of IS (Section 2.1), thus erasing the advantage of most approaches introduced in this article compared to IS. Obtaining nontrivial OPE guarantees under general function approximation is a challenging task and the investigation has started relatively recently [84, 111], revealing that potentially different coverage conditions are needed for partially observable environments (c.f. belief and outcome coverage in [111]).

The POMDP formulation can also be used to model data confoundedness, an important consideration in causal inference from observational data [80, 73]. A typical setting is that the behavior policy can depend on the latent state which is not logged in the data. OPE in such a *confounded* POMDP setting is related to but also very different from the previous (unconfounded) POMDP setting, and the existence of latent confounders bring significant challenges and require technical tools from the causal inference literature.

[33] A large part of empirical MARL is concerned with fully cooperative settings with emphases on decentralized training and/or execution, which is a very different area; see [4] for a survey.

REFERENCES

[1] AFSAR, M. M., CRUMP, T. and FAR, B. (2022). Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys* **55** 1–38.

[2] AGARWAL, A., KAKADE, S., KRISHNAMURTHY, A. and SUN, W. (2020). FLAMBE: Structural Complexity and Representation Learning of Low Rank MDPs. *arXiv preprint arXiv:2006.10814*.

[3] AGARWAL, A., KAKADE, S. M., LEE, J. D. and MAHAJAN, G. (2020). Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory* 64–66. PMLR.

[4] AMATO, C. (2024). (A Partial Survey of) Decentralized, Cooperative Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2405.06161*.

[5] AMORTILA, P., JIANG, N. and XIE, T. (2020). A Variant of the Wang-Foster-Kakade Lower Bound for the Discounted Setting. *arXiv preprint arXiv:2011.01075*.

[6] 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** 89–129.

[7] AUER, P., CESA-BIANCHI, N. and FISCHER, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning* **47** 235–256.

[8] BAI, C., WANG, L., YANG, Z., DENG, Z.-H., GARG, A., LIU, P. and WANG, Z. (2022). Pessimistic Bootstrapping for Uncertainty-Driven Offline Reinforcement Learning. In *International Conference on Learning Representations*.

[9] BAIRD, L. (1995). Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995* 30–37. Elsevier.

[10] BARRETO, A., PRECUP, D. and PINEAU, J. (2011). Reinforcement learning using kernel-based stochastic factorization. *Advances in Neural Information Processing Systems* **24**.

[11] BARRETO, A. D. M. S., PINEAU, J. and PRECUP, D. (2014). Policy iteration based on stochastic factorization. *Journal of Artificial Intelligence Research* **50** 763–803.

[12] BASSEN, J., BALAJI, B., SCHAARSCHMIDT, M., THILLE, C., PAINTER, J., ZIMMARO, D., GAMES, A., FAST, E. and MITCHELL, J. C. (2020). Reinforcement learning for the adaptive scheduling of educational activities. In *Proceedings of the 2020 CHI conference on human factors in computing systems* 1–12.

[13] BERTSEKAS, D. P. and TSITSIKLIS, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

[14] BHARDWAJ, M., XIE, T., BOOTS, B., JIANG, N. and CHENG, C.-A. (2023). Adversarial Model for Offline Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[15] CHEN, J. and JIANG, N. (2019). Information-Theoretic Considerations in Batch Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning* 1042–1051.

[16] CHEN, J. and JIANG, N. (2022). Offline reinforcement learning under value and density-ratio realizability: the power of gaps. In *Uncertainty in Artificial Intelligence* 378–388. PMLR.

[17] CHENG, C.-A., XIE, T., JIANG, N. and AGARWAL, A. (2022). Adversarially trained actor critic for offline reinforcement learning. *International Conference on Machine Learning*.

[18] CUI, Q. and DU, S. S. (2022). When are Offline Two-Player Zero-Sum Markov Games Solvable? *Advances in Neural Information Processing Systems* **35** 25779–25791.

[19] CUI, Q. and DU, S. S. (2022). Provably efficient offline multi-agent reinforcement learning via strategy-wise bonus. *Advances in Neural Information Processing Systems* **35** 11739–11751.

[20] DAI, B., SHAW, A., LI, L., XIAO, L., HE, N., LIU, Z., CHEN, J. and SONG, L. (2018). Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *International Conference on Machine Learning* 1133–1142.

[21] DUAN, Y., JIA, Z. and WANG, M. (2020). Minimax-optimal off-policy evaluation with linear function approximation. In *International Conference on Machine Learning* 2701–2709. PMLR.

[22] ERNST, D., GEURTS, P. and WEHENKEL, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research* **6** 503–556.

[23] FAN, J., WANG, Z., XIE, Y. and YANG, Z. (2020). A theoretical analysis of deep Q-learning. In *Learning for dynamics and control* 486–489. PMLR.

[24] FARAHMAND, A.-M. and SZEPESVÁRI, C. (2011). Model selection in reinforcement learning. *Machine learning* **85** 299–332.

[25] FARAHMAND, A.-M., SZEPESVÁRI, C. and MUNOS, R. (2010). Error Propagation for Approximate Policy and Value Iteration. In *Advances in Neural Information Processing Systems* 568–576.

[26] FENG, Y., LI, L. and LIU, Q. (2019). A kernel loss for solving the bellman equation. In *Advances in Neural Information Processing Systems* 15430–15441.

[27] FENG, Y., REN, T., TANG, Z. and LIU, Q. (2020). Accountable Off-Policy Evaluation With Kernel Bellman Statistics. In *Proceedings of the 37th International Conference on Machine Learning (ICML-20)*.

[28] FOSTER, D. J., KAKADE, S. M., QIAN, J. and RAKHLIN, A. (2021). The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*.

[29] FUJIMOTO, S., MEGER, D. and PRECUP, D. (2019). Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning* 2052–2062.

[30] GABBIANELLI, G., NEU, G. and PAPINI, M. (2024). Importance-weighted offline learning done right. In *International Conference on Algorithmic Learning Theory* 614–634. PMLR.

[31] GORDON, G. J. (1995). Stable function approximation in dynamic programming. In *Proceedings of the twelfth international conference on machine learning* 261–268.

[32] GRETTON, A., SMOLA, A., HUANG, J., SCHMITTFULL, M., BORGWARDT, K. and SCHÖLKOPF, B. (2008). Covariate shift by kernel mean matching.

[33] HAFNER, D., LILLICRAP, T., BA, J. and NOROUZI, M. (2019). Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

[34] HAZAN, E. et al. (2016). Introduction to online convex optimization. *Foundations and Trends® in Optimization* **2** 157–325.

[35] HUANG, A., CHEN, J. and JIANG, N. (2023). Reinforcement Learning in Low-rank MDPs with Density Features. In *Proceedings of the 40th International Conference on Machine Learning*. *Proceedings of Machine Learning Research* **202** 13710–13752. PMLR.

[36] HUANG, A., GHAVAMZADEH, M., JIANG, N. and PETRIK, M. (2023). Non-adaptive Online Finetuning for Offline Reinforcement Learning. In *NeurIPS 2023 Workshop on Generalization in Planning*.

[37] JIA, Z., RAKHLIN, A., SEKHARI, A. and WEI, C.-Y. (2024). Offline Reinforcement Learning: Role of State Aggregation and Trajectory Data. *arXiv preprint arXiv:2403.17091*.

[38] JIANG, N. (2024). A Note on Loss Functions and Error Compounding in Model-based Reinforcement Learning. *arXiv preprint arXiv:2404.09946*.

[39] JIANG, N. and HUANG, J. (2020). Minimax Value Interval for Off-Policy Evaluation and Policy Optimization. *Advances in Neural Information Processing Systems* **33**.

[40] JIANG, N., KRISHNAMURTHY, A., AGARWAL, A., LANGFORD, J. and SCHAPIRE, R. E. (2016). Contextual decision processes with low Bellman rank are PAC-learnable. *arXiv preprint arXiv:1610.09512*.

[41] JIANG, N., KRISHNAMURTHY, A., AGARWAL, A., LANGFORD, J. and SCHAPIRE, R. E. (2017). Contextual Decision Processes with low Bellman rank are PAC-Learnable. In *Proceedings of the 34th International Conference on Machine Learning* **70** 1704–1713.

[42] JIANG, N. and LI, L. (2016). Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning* **48** 652–661.

[43] JIN, C., LIU, Q. and MIRYOOSEFI, S. (2021). Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems* **34** 13406–13418.

[44] JIN, C., YANG, Z., WANG, Z. and JORDAN, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory* 2137–2143. PMLR.

[45] JIN, Y., YANG, Z. and WANG, Z. (2020). Is Pessimism Provably Efficient for Offline RL? *arXiv preprint arXiv:2012.15085*.

[46] KAKADE, S. and LANGFORD, J. (2002). Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning* **2** 267–274.

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

[48] KEARNS, M. and SINGH, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine learning* **49** 209–232.

[49] KIDAMBI, R., RAJESWARAN, A., NETRAPALLI, P. and JOACHIMS, T. (2020). Morel: Model-based offline reinforcement learning. *arXiv preprint arXiv:2005.05951*.

[50] KUMAR, A., FU, J., SOH, M., TUCKER, G. and LEVINE, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems* **32**.

[51] KUMAR, P. R. and VARAIYA, P. (2015). *Stochastic systems: Estimation, identification, and adaptive control*. SIAM.

[52] LAZARIC, A., GHAVAMZADEH, M. and MUNOS, R. (2012). Finite-sample analysis of least-squares policy iteration. *The Journal of Machine Learning Research* **13** 3041–3074.

[53] LERASLE, M. (2019). Lecture notes: Selected topics on robust statistical learning theory. *arXiv preprint arXiv:1908.10761*.

[54] LI, L., CHU, W., LANGFORD, J. and SCHAPIRE, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web* 661–670. ACM.

[55] LI, L., WALSH, T. J. and LITTMAN, M. L. (2006). Towards a unified theory of state abstraction for MDPs. In *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics* 531–539.

[56] 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* 5356–5366.

[57] MANDLEKAR, A., XU, D., WONG, J., NASIRIANY, S., WANG, C., KULKARNI, R., FEI-FEI, L., SAVARESE, S., ZHU, Y. and MARTÍN-MARTÍN, R. (2022). What Matters in Learning from Offline Human Demonstrations for Robot Manipulation. In *Conference on Robot Learning* 1678–1690. PMLR.

[58] MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A. A., VENESS, J., BELLEMARE, M. G., GRAVES, A., RIEDMILLER, M., FIDJELAND, A. K., OSTROVSKI, G. et al. (2015). Human-level control through deep reinforcement learning. *Nature* **518** 529–533.

[59] MUNOS, R. (2007). Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization* **46** 541–561.

[60] MUNOS, R. and SZEPESVÁRI, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research* **9** 815–857.

[61] MUTTI, M., DE SANTI, R., DE BARTOLOMEIS, P. and RESTELLI, M. (2023). Convex Reinforcement Learning in Finite Trials. *Journal of Machine Learning Research* **24** 1–42.

[62] NACHUM, O., CHOW, Y., DAI, B. and LI, L. (2019). Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems* **32**.

[63] NACHUM, O. and DAI, B. (2020). Reinforcement Learning via Fenchel-Rockafellar Duality. *arXiv preprint arXiv:2001.01866*.

[64] NIE, A., REUEL, A.-K. and BRUNSKILL, E. (2023). Understanding the Impact of Reinforcement Learning Personalization on Subgroups of Students in Math Tutoring. In *International Conference on Artificial Intelligence in Education* 688–694. Springer.

[65] OZDAGLAR, A. E., PATTATHIL, S., ZHANG, J. and ZHANG, K. (2023). Revisiting the linear-programming framework for offline rl with general function approximation. In *International Conference on Machine Learning* 26769–26791. PMLR.

[66] PATTERSON, A., WHITE, A. and WHITE, M. (2022). A generalized projected bellman error for off-policy value estimation in reinforcement learning. *Journal of Machine Learning Research* **23** 1–61.

[67] PERDOMO, J. C., KRISHNAMURTHY, A., BARTLETT, P. and KAKADE, S. (2023). A Complete Characterization of Linear Estimators for Offline Policy Evaluation. *Journal of Machine Learning Research* **24** 1–50.

[68] PIRES, B. A. and SZEPESVÁRI, C. (2012). Statistical linear estimation with penalized estimators: an application to reinforcement learning. *arXiv preprint arXiv:1206.6444*.

[69] PRECUP, D., SUTTON, R. S. and SINGH, S. P. (2000). Eligibility Traces for Off-Policy Policy Evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning* 759–766.

[70] PUTERMAN, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[71] ROSS, S., GORDON, G. and BAGNELL, D. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* 627–635.

[72] SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A. and KLIMOV, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[73] SHI, C., UEHARA, M., HUANG, J. and JIANG, N. (2022). A minimax learning approach to off-policy evaluation in confounded partially observable markov decision processes. In *International Conference on Machine Learning* 20057–20094. PMLR.

[74] SHIRANTHIKA, C., CHEN, K.-W., WANG, C.-Y., YANG, C.-Y., SUDANTHA, B. and LI, W.-F. (2022). Supervised optimal chemotherapy regimen based on offline reinforcement learning. *IEEE Journal of Biomedical and Health Informatics* **26** 4763–4772.

[75] SILVER, D., SCHRITTWIESER, J., SIMONYAN, K., ANTONOGLOU, I., HUANG, A., GUEZ, A., HUBERT, T., BAKER, L., LAI, M., BOLTON, A., CHEN, Y., LILLICRAP, T. P., HUI, F., SIFRE, L., VAN DEN DRIESSCHE, G., GRAEPEL, T. and HASSABIS, D. (2017). Mastering the game of go without human knowledge. *nature* **550** 354–359.

[76] SONG, Y., ZHOU, Y., SEKHARI, A., BAGNELL, D., KRISHNAMURTHY, A. and SUN, W. (2022). Hybrid RL: Using both offline and online data can make RL efficient. In *The Eleventh International Conference on Learning Representations*.

[77] SUTTON, R. S. and BARTO, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[78] SWAMINATHAN, A. and JOACHIMS, T. (2015). Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research* **16** 1731–1755.

[79] TANG, S. and WIENS, J. (2021). Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Machine Learning for Healthcare Conference* 2–35. PMLR.

[80] TENNENHOLTZ, G., SHALIT, U. and MANNOR, S. (2020). Off-policy evaluation in partially observable environments. In *Proceedings of the AAAI Conference on Artificial Intelligence* **34** 10276–10283.

[81] THOMAS, P. S. and BRUNSKILL, E. (2016). Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning* 2139–2148.

[82] TSITSIKLIS, J. N. and VAN ROY, B. (1996). Feature-based methods for large scale dynamic programming. *Machine Learning* **22** 59–94.

[83] UEHARA, M., HUANG, J. and JIANG, N. (2020). Minimax Weight and Q-Function Learning for Off-Policy Evaluation. In *Proceedings of the 37th International Conference on Machine Learning* 1023–1032.

[84] UEHARA, M., KIYOHARA, H., BENNETT, A., CHERNOZHUKOV, V., JIANG, N., KALLUS, N., SHI, C. and SUN, W. (2022). Future-Dependent Value-Based Off-Policy Evaluation in POMDPs. *arXiv preprint arXiv:2207.13081*.

[85] UEHARA, M., ZHANG, X. and SUN, W. (2022). Representation Learning for Online and Offline RL in Low-rank MDPs. In *International Conference on Learning Representations*.

[86] VAN HASSELT, H., DORON, Y., STRUB, F., HESSEL, M., SONNERAT, N. and MODAYIL, J. (2018). Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*.

[87] VINYALS, O., BABUSCHKIN, I., CZARNECKI, W. M., MATHIEU, M., DUDZIK, A., CHUNG, J., CHOI, D. H., POWELL, R., EWALDS, T., GEORGIEV, P., OH, J., HORGAN, D., KROISS, M., DANIHELKA, I., HUANG, A., SIFRE, L., CAI, T., AGAPIOU, J. P., JADERBERG, M., VEZHNEVETS, A. S., LEBLOND, R., POHLEN, T., DALIBARD, V., BUDDEN, D., SULSKY, Y., MOLLOY, J., PAINE, T. L., GÜLÇEHRE, Ç., WANG, Z., PFAFF, T., WU, Y., RING, R., YOGATAMA, D., WÜNSCH, D., MCKINNEY, K., SMITH, O., SCHAUL, T., LILLICRAP, T. P., KAVUKCUOGLU, K., HASSABIS, D., APPS, C. and SILVER, D. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575** 350–354.

[88] WAGENMAKER, A. and PACCHIANO, A. (2023). Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning* 35300–35338. PMLR.

[89] WANG, R., FOSTER, D. P. and KAKADE, S. M. (2020). What are the Statistical Limits of Offline RL with Linear Function Approximation? *arXiv preprint arXiv:2010.11895*.

[90] WANG, R., WU, Y., SALAKHUTDINOV, R. and KAKADE, S. (2021). Instabilities of offline rl with pre-trained neural representation. In *International Conference on Machine Learning* 10948–10960. PMLR.

[91] WIESEMANN, W., KUHN, D. and RUSTEM, B. (2013). Robust Markov decision processes. *Mathematics of Operations Research* **38** 153–183.

[92] XIAO, C., WU, Y., MEI, J., DAI, B., LATTIMORE, T., LI, L., SZEPESVARI, C. and SCHUURMANS, D. (2021). On the optimality of batch policy optimization algorithms. In *International Conference on Machine Learning* 11362–11371. PMLR.

[93] XIE, T., CHENG, C.-A., JIANG, N., MINEIRO, P. and AGARWAL, A. (2021). Bellman-consistent Pessimism for Offline Reinforcement Learning. *arXiv preprint arXiv:2106.06926*.

[94] XIE, T., FOSTER, D. J., BAI, Y., JIANG, N. and KAKADE, S. M. (2023). The Role of Coverage in Online Reinforcement Learning. In *The Eleventh International Conference on Learning Representations*.

[95] XIE, T. and JIANG, N. (2020). Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence* 550–559. PMLR.

[96] XIE, T. and JIANG, N. (2021). Batch value-function approximation with only realizability. In *International Conference on Machine Learning* 11404–11413. PMLR.

[97] XIE, T., JIANG, N., WANG, H., XIONG, C. and BAI, Y. (2021). Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems* **34** 27395–27407.

[98] XIE, T., MA, Y. and WANG, Y.-X. (2019). Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. *Advances in neural information processing systems* **32**.

[99] YANG, L. F. and WANG, M. (2019). Sample-optimal parametric Q-learning with linear transition models. *arXiv preprint arXiv:1902.04779*.

[100] YE, C., XIONG, W., ZHANG, Y., JIANG, N. and ZHANG, T. (2024). A theoretical analysis of nash learning from human feedback under general kl-regularized preference. *arXiv preprint arXiv:2402.07314*.

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

[102] 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*.

[103] ZAHAVY, T., O'DONOGHUE, B., DESJARDINS, G. and SINGH, S. (2021). Reward is enough for convex mdps. *Advances in Neural Information Processing Systems* **34** 25746–25759.

[104] ZANETTE, A. (2020). Exponential Lower Bounds for Batch Reinforcement Learning: Batch RL can be Exponentially Harder than Online RL. *arXiv preprint arXiv:2012.08005*.

[105] ZANETTE, A. (2023). When is realizability sufficient for off-policy reinforcement learning? In *International Conference on Machine Learning* 40637–40668. PMLR.

[106] ZANETTE, A., WAINWRIGHT, M. J. and BRUNSKILL, E. (2021). Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems* **34** 13626–13640.

[107] ZHAN, W., HUANG, B., HUANG, A., JIANG, N. and LEE, J. (2022). Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory* 2730–2775. PMLR.

[108] ZHANG, S. and JIANG, N. (2021). Towards hyperparameter-free policy selection for offline reinforcement learning. *Advances in Neural Information Processing Systems* **34** 12864–12875.

[109] ZHANG, T. (2006). From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation.

[110] ZHANG, Y., BAI, Y. and JIANG, N. (2023). Offline learning in markov games with general function approximation. In *International Conference on Machine Learning* 40804–40829. PMLR.

[111] ZHANG, Y. and JIANG, N. (2024). On the Curses of Future and History in Future-dependent Value Functions for Off-policy Evaluation. *arXiv preprint arXiv:2402.14703*.

[112] ZHAO, Y., KOSOROK, M. R. and ZENG, D. (2009). Reinforcement learning design for cancer clinical trials. *Statistics in medicine* **28** 3294–3315.

[113] ZHAO, Y., ZENG, D., SOCINSKI, M. A. and KOSOROK, M. R. (2011). Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics* **67** 1422–1433.

[114] ZHENG, G., ZHANG, F., ZHENG, Z., XIANG, Y., YUAN, N. J., XIE, X. and LI, Z. (2018). DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 world wide web conference* 167–176.

[115] ZHOU, R., LIU, T., CHENG, M., KALATHIL, D., KUMAR, P. and TIAN, C. (2024). Natural Actor-Critic for Robust Reinforcement Learning with Function Approximation. *Advances in neural information processing systems* **36**.

[116] ZHU, H., RASHIDINEJAD, P. and JIAO, J. (2023). Importance weighted actor-critic for optimal conservative offline reinforce-

ment learning. *Advances in Neural Information Processing Systems* **36**.