Information-Directed Pessimism for Offline Reinforcement Learning

Alec Koppel [1] **Sujay Bhatt** [1] **Jiacheng Guo** [2] **Joe Eappen** [3] **Mengdi Wang** [2] **Sumitra Ganesh** [1]

Abstract

Policy optimization from batch data, i.e., offline reinforcement learning (RL) is important when collecting data from a current policy is not possible. This setting incurs distribution mismatch between batch training data and trajectories from the current policy. Pessimistic offsets estimate mismatch using concentration bounds, which possess strong theoretical guarantees and simplicity of implementation. Mismatch may be conservative in sparse data regions and less so otherwise, which can result in under-performing their no-penalty variants in practice. We derive a new pessimistic penalty as the distance between the data and the true distribution using an evaluable one-sample test known as Stein Discrepancy that requires minimal smoothness conditions, and noticeably, allows a mixture family representation of distribution over next states. This entity forms a quantifier of information in offline data, which justifies calling this approach *information-directed pessimism* (IDP) for offline RL. We further establish that this new penalty based on discrete Stein discrepancy yields practical gains in performance while generalizing the regret of prior art to multimodal distributions.

1. Introduction

Reinforcement learning (RL), mathematically encapsulated by a Markov Decision Process (MDP) (Puterman, 2014), is a framework in which an autonomous agent moves through a state space, selects actions according to a policy, and incrementally receives rewards from the environment. Solution techniques for different challenges RL broadly consist of those in the 'policy' space and those in the 'value' space (Sutton et al., 2017; Bhatt et al., 2021; 2022). Sampling trajectories from the model conditioned on current policy may not always be possible, as in finance (Tamar et al., 2014), field robotics (Gregory et al., 2016) or games (Bhatt & Başar, 2020), which gives rise to the requirement of building a simulator (Todorov et al., 2012; Ardon et al., 2023) or using historical data to evaluate policies. In offline RL, data is collected by a *behavioral policy* distinct from the one whose parameters are being trained, which results in distribution mismatch. Efforts to correct this mismatch include importance weighting and *pessimistic* regularization. The former, which re-weights updates based on an estimate of the behavioral policy's induced occupancy measure, exhibits exponential variance dependence on the horizon (Gelada & Bellemare, 2019; Nachum et al., 2019; Rashidinejad et al., 2022). This issue can be stabilized through control variates (Jiang & Li, 2016; Dong et al., 2023) which can be difficult to tune in practice. Whether importance weighting achieves optimal sample complexity remains unknown. Alternatively, pessimistic penalization of model-free RL methods have been vetted experimentally (Kumar et al., 2020) in continuous domains, and can achieve near-optimal sample complexity in theory for value iteration (Rashidinejad et al., 2021) and Q learning (Shi et al., 2022) when combined with covariance information (Li et al., 2022; Yan et al., 2023) in discrete domains.

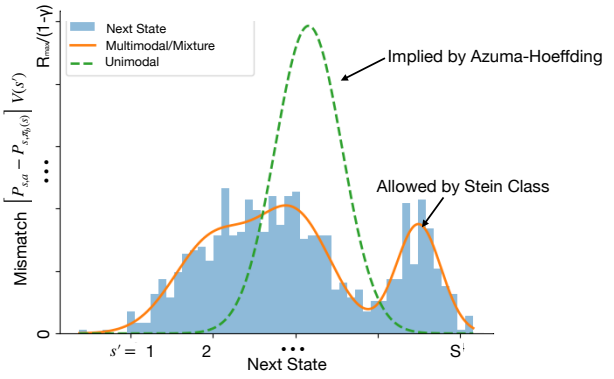

Figure 1: Visualization of distribution mismatch. Concentration bounds (Azuma-Hoeffding) implicitly require unimodality to be valid forms of pessimism. Notions introduced here enable offline RL algorithms to achieve sublinear regret when transition belongs to a mixture family.

[1]J.P. Morgan AI Research, 383 Madison Ave., 9th floor, New York, NY 10017 [2]Dept. of ECE, Princeton University, Princeton, NJ 08544, Country [3]Dept. of ECE, Purdue University, West Lafayette,IN, 47906. Correspondence to: Alec Koppel <alec.koppel@jpmchase.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

The aforementioned analyses all identify that the mismatch is bottlenecked by the difference in the value function of the current policy as compared to the optimal policy (Jin et al., 2021) that depends on the distributional distance between the transition model conditioned on the current policy and the optimal policy. This quantity is not computable in practice, which makes theory based upon it disconnected from algorithm implementation. Efforts to reduce it to tractable statistics, especially an absolute upper-bound on the likelihood ratio in the induced occupancy measures associated with the behavioral policy and the optimal policy called the *concentrability* coefficient (CC) (Xie et al., 2022), undergird the understanding of the sample complexity of offline RL. More specifically, probabilistic approximations (Shi et al., 2022; Uehara et al., 2023) to the CC, bootstraps (Nguyen-Tang & Arora, 2023; Sun et al., 2023), and model-based regularizers (Levine et al., 2020; Yu et al., 2020; Kidambi et al., 2020; Rigter et al., 2022; Karabag & Topcu, 2023) have been proposed to address mismatch, but their rates depend on information-theoretic constants that are not tractable in practice, and implicitly require empirical transition matrix to be unimodal for using concentration bounds. This context has led to a lack of understanding of the role of information in an offline data set in practice. Thus, we focus on the following:

Can we measure the information in a batch data in a way that is theoretically justified and practically computable? Can we generalize mismatch conditions underlying pre-existing sample complexity guarantees for offline RL?

We identify that "spurious correlation" (Jin et al., 2021)[Def. 4.1] is proportional to the integral probability metric (IPM) difference between the transition model conditioned on the behavioral versus optimal policy, which in general demands a two-sample test to be estimated (Gretton et al., 2012). Doing so is intractable in the offline setting where one cannot sample from the "true" MDP. Importantly, however, under an assumption that the value function of the true MDP belongs to the Stein class[1] associated with a base kernel, i.e., belongs to a mixture family associated with a certain reproducing kernel Hilbert space (RKHS) (Berlinet & Thomas-Agnan, 2011) such that the IPM may be estimated according to a *one-sample test* called the Kernelized Stein Discrepancy (KSD) (Gorham & Mackey, 2015). KSD arises from Bayesian inference (Chen et al., 2018; 2019a; Dwivedi & Mackey, 2022; Shetty et al., 2022), where it has sharpened the convergence of Monte Carlo samplers. Our motivation to employ KSD in this way both comes from the fundamental role of IPMs in defining pessimism, as well as recent work that defines KSD as the "Stein information" in lieu of mutual information (Russo & Van Roy, 2016) in online model-based bandits/RL (MBRL) (Chakraborty et al., 2023b;a) as an exploration incentive. That Stein information achieves comparable regret to information-directed sampling (IDS) while outperforming it experimentally due to its simple implementation, motivates us to use it as a way to compute mismatch in offline RL. Thus, this work puts forward a family of methods called *information-directed pessimism* (IDP), whose associated **contributions** are to:

- derive the technical machinery to make KSD operable in discrete offline RL settings (Sec. 3);
- derive new penalties for information-directed value iteration (IDP-VI) and information-directed Q-learning (IDP-Q) (pseudo-code in Appendix C);
- establish that IDP-VI (Theorem 4.3) and IDP-Q (Theorem 4.4) exhibit sublinear regret comparable to prior works, but under general multimodal transitions;
- demonstrate the practical merits of the proposed methods relative to benchmarks, especially when the behavioral policy is exploratory ("hard" settings (Kumar et al., 2019; 2020)) or MDP transitions exhibit multimodality [2].

An overview of related rate analyses for offline RL in terms of model-free/ model-based, infinite-horizon/episodic, and tabular/parameterized settings, the relative dependence on data coverage conditions and defined notions of pessimistic penalties may be found in Table 3 (Appendix A).

2. Problem Formulation

We consider state and action spaces, respectively, $\mathcal{S}$ and $\mathcal{A}$ as finite discrete sets, i.e., $|\mathcal{S}| = S$ and $|\mathcal{A}| = A$. Starting from a state $s \in \mathcal{S}$, an action $a \in \mathcal{A}$ causes a transition to the next state s' according to conditional distribution $\mathbb{P}(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ that depends on the current state and action. Here $\Delta(\cdot)$ denotes the probability simplex over the set in its argument, i.e., the set of vectors with non-negative weights that sum to unit. This transition probability in the tabular setting can be succinctly represented as a matrix in $\mathbb{R}^{SA \times S}$, with $P_{s,a} = \mathbb{P}(\cdot \mid s, a) \in \mathbb{R}^{1 \times S}$ as a distribution over next states s'. Further denote individual entries of this matrix as $P_{s,a,s'} := \mathbb{P}(s' \mid s, a)$.

At each time t, the agent executes an action $a_t \in \mathcal{A}$ given the current state $s_t \in \mathcal{S}$, following a possibly stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, i.e., $a_t \sim \pi(\cdot \mid s_t)$. Then, given the state-action pair (s_t, a_t), the agent observes a reward $r_t = R(s_t, a_t)$. The goal is for the agent to accumulate the

[1] We develop algorithms that alleviate this condition via "optimistic" transition estimates (Auer et al., 2008) in Sec. 3.2 following Prop. 3.1.

[2] Code is available here: `https://github.com/jeappen/idp-offline-rl`

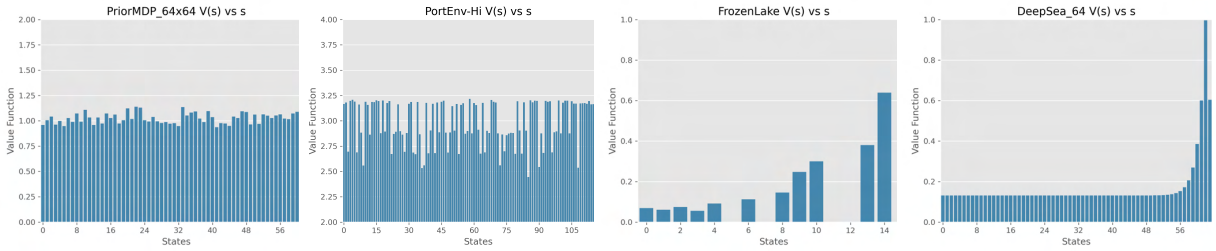

Figure 2: Empirical cumulative return of optimal policy over initial state, which is an inner-product between (empirical) occupancy measure and vectorized reward. Thus, it drives the structure of mismatch, which in PriorMDP (Markou & Rasmussen, 2019) and Portfolio environments (Suttle et al., 2022) are slow-decaying without central tendency. Frozen Lake (Brockman et al., 2016) and DeepSea (Osband & Van Roy, 2017a) exhibit unimodality.

most reward in the long term on average, a quantity called *value*. Thus, under any policy π that maps states to actions, one can define the value function $V_\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_\pi(s) = \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathbb{P}(\cdot|s_t,a_t)} \left(\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s \right), \tag{2.1}$$

which quantifies the long-term expected accumulation of rewards discounted by $\gamma \in (0,1)$. The goal is to find the policy π that maximizes the long-term return $V_\pi(s_0 = s)$, i.e., to solve the following optimization problem

$$\max_{\pi \in \Pi} \; V_\pi(s), \tag{2.2}$$

when the model, i.e., the transition probability $\mathbb{P}$ and reward function R, are unknown to the agent. Subsequently denote as $\pi^\star$ the maximizer of (2.2).

Subsequently, denote the distribution $\mu_\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s \mid s_0, \pi)$ as the *discounted state-occupancy measure*, which is a valid probability measure over the state space $\mathcal{S}$ – see (Sutton et al., 2000). For notational convenience, we let $\mu_\pi(s,a) = \mu_\pi(s) \cdot \pi(a \mid s)$, which denotes the *discounted state-action occupancy measure*. Further define $\mu_{\pi^\star}(s,a)$ as the state-action occupancy measure of the optimal policy. Define the action-value, or Q-function, as $Q_\pi(s,a) = \mathbb{E}\big(\sum_{t=0}^{\infty} \gamma^t r_t \,\big|\, s_0 = s, a_0 = a, a_t \sim \pi(s_t)\big)$ as the value $V_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ conditioned on an initial action, when following policy π thereafter.

We further hypothesize that initial state s_0 is sampled from initial state distribution and define a scalarized performance objective for policy π as its expected accumulated value:

$$J(\pi) := \mathbb{E}_{s \sim \rho}\Big[V_\pi(s)\Big]. \tag{2.3}$$

2.1. Offline Reinforcement Learning

Consider a fixed data batch $\mathcal{D} = \{s_{u-1}, a_{u-1}, r_{u-1}, s_u\}_{u=1}^{T}$ sampled from distribution μ_b (Xie et al., 2021a):

$$\mu_b(s,a) = \frac{1}{1-\gamma} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s, a_t = a \,|\, s_0, a_t \sim \pi_b(s_t)) \tag{2.4}$$

is the state-action occupancy measure, i.e., distribution of the Markov chain induced by the product between the long-run probability of being in a state when following behavioral policy π_b. Denote as $\tau = (s_0, a_0, s_1, a_1 \cdots, s_H, a_H)$ a collection of state-action tuples of length H from $\mathcal{D}$, and let $p_\pi(\tau) = \rho(s_0) \prod_{j=0}^{H} \pi(a_t|s_t) P_{s_t,\pi(s_t)}$ denote the distribution over the collection. Given H episodes $\{s_i, a_i, R(s_i,a_i), s_i'\}_{i=1}^{H}$ in the dataset $\mathcal{D}$, with $N(s,a) = \sum_{i=1}^{H} \mathbb{1}((s_i,a_i) = (s,a))$, define empirical reward and transition matrix elements:

$$\hat{R}(s,a) = \frac{1}{N(s,a)} \sum_{i=1}^{H} R(s_i,a_i) \mathbb{1}((s_i,a_i) = (s,a))$$

$$\hat{P}_{s,a,s'} = \begin{cases} \sum_{i=1}^{H} \frac{\mathbb{1}((s_i,a_i,s_i') = (s,a,s'))}{N(s,a)}, & \text{if } N(s,a) > 0 \\ \frac{1}{S}, & \text{else.} \end{cases}$$

The goal of offline RL is to use the offline dataset $\mathcal{D}$ to compute a policy $\hat{\pi}$ that is close to the optimal $\pi^\star$, i.e., to ensure the sub-optimality w.r.t optimal policy $\pi^\star$:

$$J(\pi^\star) - J(\hat{\pi}) \le \mathcal{O}(T^{-k}) \tag{2.5}$$

for some $k > 0$, and the left-hand side is either in expectation or with high probability w.r.t to batch data [cf. (2.4)].

2.2. Pessimism in Offline RL

Observe that one cannot evaluate the objective [cf. (2.5)] under an arbitrary policy, and in particular the one associated with optimal trajectories $p_{\pi^\star}(\tau)$, since $\pi^\star$ may require visitation to states that are not contained in the offline dataset. This issue leads to distribution mismatch, as expectation or probability with respect to (2.4) is with respect to the behavioral policy of the prior dataset. If unaddressed, mismatch leads to overestimating the value function during training, resulting in possibly spurious action choices (Kumar et al., 2019).

To address this gap, pessimistic (Kumar et al., 2020; Rashidinejad et al., 2021) offsets may improve the cumulative return on test trajectories, when training from offline

data by subtracting a penalty associated with the probability that the estimated return deviates from the true return of the current state-action pair. Under an implicit unimodal hypothesis, this probability is well-encapsulated by concentration bounds used to define lower confidence-bound (LCB)-offsets to RL algorithms of various types (Rashidinejad et al., 2021; Jin et al., 2021; Xie et al., 2021a; Shi et al., 2022; Yan et al., 2023; Li et al., 2022). However, experimentally (see Fig 2), we find that cumulative returns possess this attribute only for highly structured environments; for MDPs whose transition models possess higher volatility, unimodality appears invalid. Thus, LCB-based penalties are simple to compute and well-capture the worst-case deviation of the value function computed from offline data w.r.t. true value function, although they may be overly conservative in sparse data regions, and the insufficiently so for densely sampled data.

We propose a new penalty that is inspired by information directed sampling (IDS) in bandits and MDPs literature (Russo & Van Roy, 2018; Lu et al., 2023), but redefines the notion of information to instead be based on distributional distance. Doing so allows us to define a penalty that is distinct for each state-action pair, and that measures the discrepancy of the next state transitions between the data and the true model, under some specific structural hypothesis on the class of value functions (which we are able to relax – see Sec 3.2).

3. Information-Directed Pessimism

In (Yan et al., 2023)[Lemma 2] and (Shi et al., 2022)[Lemma A.1], among others (Kumar et al., 2020; Jin et al., 2021), pessimism is designed to annihilate a "spurious correlation," which appears as a gap between the value function under the true MDP transition model and its empirical estimate constructed via offline data:

$$|(P_{s,a} - \hat{P}_{s,a})V|. \tag{3.1}$$

This *bias* in the value estimate is then aggregated across the entire state space w.r.t. $s' \in \mathcal{S}$ and offline data set to quantify the expected difference in the value function under the empirical offline and true transition dynamics in prior regret analyses, especially (Yan et al., 2023; Shi et al., 2022). The form of (3.1), combined with the fact that regret analysis tends to aggregate over states and samples in $\mathcal{D}$, suggests employing distributional distance such as an integral probability metric (IPM).

On LCB, IPM, KSD, and IDS. Before shifting focus to defining the way this machinery may be employed to define a new pessimistic penalty, we expand upon their connection to IDS (Russo & Van Roy, 2018), IPMs (Gretton et al., 2012), and concentration bounds (Hoeffding, 1994; Bernstein, 1924). From equation 3.1, it is clear that integral probability metrics (IPMs) are the appropriate mathematical machinery to quantify distribution mismatch; however, their intractability in general has led prior works (Yan et al., 2023; Rashidinejad et al., 2021; Li et al., 2022) to approximate the probability of deviation instead using concentration bounds making unimodal assumptions. However these are frequently violated in practice, meaning that the mismatch may not have central tendency across the state and action spaces (Moerland et al.). This motivates exploring discrepancy measures for pessimism.

On the other hand, using Kernelized Stein Discrepancy (KSD) to evaluate mismatch between a nominal and target measure has seen recent success as an exploration incentive in model-based RL (MBRL) (Chakraborty et al., 2023a), where KSD is defined as the "**Stein information**" in lieu of mutual information (Lu et al., 2021; Russo & Van Roy, 2018). Mutual information appears in the lower-bound on the achievable regret in Bayesian bandits (Russo & Van Roy, 2016) and MBRL (Lu et al., 2021). However, algorithms that achieve this lower-bound by introducing intrinsic curiosity in the form of mutual information between the current and optimal policy to augment posterior sampling by uncertainty about the optimal policy, i.e., IDS and its variants (Russo & Van Roy, 2018; Hao & Lattimore, 2022), require estimating mutual information which is not efficient in practice. "Stein information," as coined by (Chakraborty et al., 2023a), provides an alternate exploration incentives in practice that matches the regret of IDS. Stein information also turns out to be a suitable quantifier of mismatch in offline RL, and hence methods introduced next are referred to as *information-directed* pessimism (IDP).

3.1. Discrete Stein Discrepancy (DSD)

Computation of DSD hinges upon evaluating the score function of the target distribution, which is analogous to the gradient of the log-likelihood in continuous settings (Liu et al., 2016; Chen et al., 2018). We begin with a method to construct the discrete score function of the true MDP transition model, and then develop a way to operate with only "optimistic" estimates of it from offline data. Proceed then by defining permutation and inverse permutation, which are required to formalize the notion of a discrete score function (Yang et al., 2018). Specifically, we augment definitions from (Yang et al., 2018) to address conditional distributions in the form of $\mathbb{P}(\cdot \mid s, a)$ associated with the row vector $P_{s,a}$ defined in Sec. 2.

Let $\vee$ denote the cyclic permutation[3] for set $\mathcal{S}$, such that

[3]Cyclic Permutation (Yang et al., 2018): For a finite discrete set $\mathcal{X}$, a cyclic permutation $\vee : \mathcal{X} \mapsto \mathcal{X}$ is a bijective function s.t. for some ordering $x^{[1]}, \cdots, x^{[|\mathcal{X}|]}$, $\vee x^{[i]} = x^{[(i+1) \bmod |\mathcal{X}|]}$, for all $i \leq |\mathcal{X}|$.

for $s' \in \mathcal{S}$, $\vee_i s'$ is the vector that undergoes a cyclic permutation at its i-th component.[4] Let the inverse permutation operator $\wedge$ be such that $\vee(\wedge(s')) = \wedge(\vee(s')) = s'$. The partial difference operator w.r.t any function $f : \mathcal{S} \to \mathbb{R}$ for $i = 1, \ldots, |\mathcal{S}|$ is defined as

$$\triangle_{s'_i} f(s') := f(s') - f(\vee_i s') \text{ for } s' \in \mathcal{S}, \tag{3.2}$$

and we denote the difference operator associated with the inverse permutation as

$$\triangle^*_{s'_i} f(s') := f(s') - f(\wedge_i s') \text{ for } s' \in \mathcal{S}. \tag{3.3}$$

Let the conditional *discrete score function* of the distribution $P_{s,a}$ be defined as in (Yang et al., 2018), for $i = 1, 2, \cdots, |\mathcal{S}|$,

$$\mathbb{S}_{P_{s,a}}(s')_i = \frac{\triangle_{s'_i} P_{s,a,s'}}{P_{s,a,s'}} = 1 - \frac{P_{s,a,\vee_i s'}}{P_{s,a,s'}} \tag{3.4}$$

but now for conditional distributions. Here $P_{s,a,s'}$ denotes the true transition matrix. Denote as $\mathbb{A}_{P_{s,a}}$ the difference Stein operator of $P_{s,a}$:

$$\mathbb{A}_{P_{s,a}} f(s') := \mathbb{S}_{P_{s,a}}(s') f(s') - \Delta^* f(s'), \tag{3.5}$$

for any function f. The discrete Stein discrepancy (DSD) between data $\mathcal{D}^{(s,a)} = \{(s, a, s') | s' \in \mathcal{D}\}$ and the probability of s' identified by the $(s, a)^{th}$ row of the true transition matrix P is defined as

$$\text{DSD}(\mathcal{D}^{(s,a)}, P_{s,a}) := \sup_{f \in \mathcal{F}} \mathbb{E}_{s' \sim \mathcal{D}^{(s,a)}} [\mathbb{A}_{P_{s,a}} f(s')].$$

The discrepancy in this form is not evaluable, but restriction of the function class to RKHS leads to a closed-form evaluation (Yang et al., 2018). For $s' \in \mathcal{S}$, let l denote the positive definite (exponential) Hamming kernel, i.e., $l(s', \check{s}) = \exp\{-\frac{1}{d} \sum_{i=1}^{d} \mathcal{I}\{s_i' \neq \check{s}_i\}\}$, where d is the dimension of the state representation. Let $\mathcal{H}$ denote the RKHS associated with the kernel l. With $\mathcal{F}$ as the unit ball in the RKHS $\mathcal{H}$, the (kernelized) DSD is:

$$\text{DSD}(\mathcal{D}^{(s,a)}, P_{s,a}) := \sup_{f \in \mathcal{H},\ \|f\|_{\mathcal{H}} \leq 1} \mathbb{E}_{s' \sim \mathcal{D}^{(s,a)}} [\mathbb{A}_{P_{s,a}} f(s')].$$

The discrepancy may be computed in closed-form as a conditional plug-in variant of (Yang et al., 2018)[Theorem 7]:

$$\textbf{DSD}(\mathcal{D}^{(s,a)}, P_{s,a}) := \sqrt{\mathbb{E}_{s', \check{s} \sim \mathcal{D}^{(s,a)}} \left[\kappa_{P_{s,a}}(s', \check{s}) \right]}, \tag{3.6}$$

where the *discrete Stein kernel* w.r.t $P_{s,a}$ is given as

$$\begin{aligned} \kappa_{P_{s,a}}(s', \check{s}) = & \mathbb{S}_{P_{s,a}}(s')^T l(s', \check{s}) \mathbb{S}_{P_{s,a}}(\check{s}) - \mathbb{S}_{P_{s,a}}(s')^T \triangle^*_{\check{s}} l(s', \check{s}) \\ & - \triangle^*_{s'} l(s', \check{s})^T \mathbb{S}_{P_{s,a}}(\check{s}) + \texttt{trace}(\triangle^*_{s', \check{s}} l(s', \check{s})). \end{aligned} \tag{3.7}$$

Here s' and $\check{s}$ are distinct next-states associated with conditional distribution $\mathbb{P}(\cdot \mid s, a)$ sampled from dataset $\mathcal{D}^{(s,a)}$.

[4] Regarding the subscript i, we note the state $s \in \mathcal{S}$ may be represented as a vector of length $|\mathcal{S}|$, e.g., s may be represented through its one-hot encoding, each of which is a vector in $\mathbb{R}^{|\mathcal{S}|}$ with 1 in the position of i and 0 elsewhere.

3.2. Info-directed pessimistic algorithms

Before we derive the penalty for the two flavors of offline RL algorithms, we bound on the spurious correlation using the DSD, which represents the discrepancy between the data set $\mathcal{D}$ and true MDP transition model P. Intuitively, the penalty should be larger for state-action regions where there is more mismatch due to insufficient samples in the batch data set. That this is so is formalized next.

Proposition 3.1. *Suppose the rewards are bounded $r \in [0, 1]$ implying $0 \leq V(s) \leq \frac{1}{1-\gamma}$ for all s. For all $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have*

$$|(P_{s,a} - \hat{P}_{s,a}) V| \leq \frac{1}{1-\gamma} \sqrt{\mathbb{E}_{s', \check{s} \sim \mathcal{D}^{(s,a)}} \left[\kappa_{P_{s,a}}(s', \check{s}) \right]},$$

with probability 1*, where $\hat{P}_{s,a}$ is computed from $\mathcal{D}^{(s,a)}$.*

The deviation (proof in Appendix D.1) derived using Proposition 3.1 provides a *deterministic* upper bound for the deviation of the empirical model from the data, unlike high-probability bounds derived using concentration inequalities in (Rashidinejad et al., 2021; Li et al., 2022; Shi et al., 2022).

Suppose that we have access to sampled values of the score function $\mathbb{S}_{P_{s,a}}$ [cf. Eq. 3.4] of the generating process underlying the MDP, meaning $\mathbb{P}(\cdot \mid s_t, a_t)$ belongs to a family of mixture models associated with the discrete Stein class. We may evaluate the discrepancy between samples $(s, a, s') \sim \mathcal{D}$ from the offline dataset and the true transition distribution $\mathbb{P}(\cdot \mid s, a)$ through the DSD [cf. Eq. 3.6]. The knowledge of the 'true' score function is feasible in certain financial applications (Limmer & Horvath, 2023), where access to a generative model is provided by a physics or market simulation engine.

Alleviating Score Function Access. Knowledge of the score function may not be viable in all offline RL problems. Thus, we propose a method to 'estimate' the score function from the offline dataset to be used in DSD computation [cf. Eq. 3.6] based upon classical transition estimation techniques (Strehl & Littman, 2005; Jaksch et al., 2010). We briefly describe the method, with further details in Appendix C.1. Consider an ordering of states $\mathcal{S} := \{s'_1, s'_2, \cdots, s'_S\}$ based on values as $V_{t-1}(s'_1) \geq V_{t-1}(s'_2) \geq \cdots \geq V_{t-1}(s'_n)$. The 'estimated' model is computed as

$$\tilde{P}_{s,a} := \underset{\tilde{p} \in \mathcal{P}(s,a)}{\arg\max} \sum_{s' \in \mathcal{S}} \tilde{p}(s') V_{t-1}(s'), \tag{3.10}$$

where, for $n_t(s, a) := |\{\tau : \tau \leq t,\ s_\tau = s, a_\tau = a\}|$

$$\mathcal{P}(s, a) := \left\{ \tilde{p} : \|\tilde{p} - \hat{P}^t_{s,a}\|_1 \leq \sqrt{\frac{14 S \log(2At/\delta)}{\max\{1, n_t(s, a)\}}} \right\},$$

Table 1: Penalties for Value Iteration $b^v(\cdot,\cdot)$ and Q-learning $b^q(\cdot,\cdot)$

$$b_t^v(s,a) := \begin{cases} \alpha \cdot \sqrt{\frac{1}{n^2(s,a)} \sum_{s',\check{s}\in\mathcal{D}^{(s,a)}} \kappa_{\tilde{P}_{s,a}}(s',\check{s})}, & \text{if } (s,a) \in \mathcal{D}^{(s,a)} \\ \max_{(s,a)} \left\{ \alpha \cdot \sqrt{\frac{1}{n^2(s,a)} \sum_{s',\check{s}\in\mathcal{D}^{(s,a)}} \kappa_{\tilde{P}_{s,a}}(s',\check{s})} \right\}, & \text{otherwise.} \end{cases} \tag{3.8}$$

$$b_t^q(s,a) := \max\left\{ \sqrt{\frac{n_t(s,a)-1}{n_t(s,a)}} b_{t-1}^q(s,a), \alpha \cdot \sqrt{\frac{1}{n_t^2(s,a)} \sum_{s',\check{s}\in\mathcal{D}_t^{(s,a)}} \kappa_{\tilde{P}_{s,a}}(s',\check{s})} \right\}. \tag{3.9}$$

with $\hat{P}_{s,a}^t$ denoting the empirical transition matrix at t, which is simply $\hat{P}$ with $n_t(s,a)$ in place of $N(s,a)$. Moreover, $V_{t-1}(\cdot)$ is any estimate of the value function $t-1$. The optimization in Eq. 3.10 is such that the estimated transition probabilities have larger weight allocated to states with maximum values at the expense of those states with smaller values. Considering that these values already[5] incorporate a penalty (Eq. 3.11 and Eq. 3.12), we have a new transition function in the convex polytope $\mathcal{P}(s,a)$ that best represents the offline data. This estimated distribution $\tilde{P}_{s,a}$ is used in score function computation [cf. Eq. 3.4].

The penalty based on DSD can be substituted into the notion of pessimistic penalty in any algorithm in principle to potentially improve empirical performance and relax the unimodal assumptions implicit in how mismatch is quantified. Next, we develop pessimistic value iteration (Rashidinejad et al., 2021) and pessimistic asynchronous Q-learning (Yan et al., 2023), with specific variants of the DSD [cf. (3.7)] determined by the regret analysis given in Table 1.

- *Information-Directed Pessimistic Value Iteration.* The algorithm first estimates the (possibly optimistic) transition matrix to enable score function evaluation. Then, we set an initial value function estimate V_0 and action-value function estimate Q_0, and we calculate the penalty term $b_t^v(s,a)$ as in Eq. 3.8 $\forall (s,a) \in \mathcal{D}$ using the 'true' model estimate $\tilde{P}_{s,a}$ in Eq. 3.10. Finally, we update the value estimates Q_t and V_t for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ using a fixed-point iteration:

$$Q_t(s,a) \leftarrow \hat{R}_t(s,a) - b_t^v(s,a) + \gamma \hat{P}_{s,a}^t \cdot V_t,$$
$$V_t(s) \leftarrow \max_a Q_t(s,a), \ \ \forall\, s. \tag{3.11}$$

  See Alg. 2 in the appendix for details. The penalty b_t^v is derived by seeking a decrement on the value error, via Prop. 3.1, which is defined via equation D.7.

- *Information-Directed Pessimistic Q-Learning.* With an initial action-value function estimate Q_0, we sample a transition (s,a,r,s') from $\mathcal{D}$ following which we calculate the penalty term $b_t^q(s,a)$ using Eq. 3.9. Here $n_t(s,a)$ denotes the number of occurrences of (s,a) within the initial t samples of the dataset. For asynchronous Q learning, one samples tuples $\{s_{t-1},a_{t-1},r_{t-1},s_t\} \in \mathcal{D}$ from the offline data and updates the values as follows:

$$Q_t(s_{t-1},a_{t-1}) = (1-\eta) Q_{t-1}(s_{t-1},a_{t-1}) \tag{3.12}$$
$$+ \eta\Big[\hat{R}(s_{t-1},a_{t-1}) + \gamma \max_{a'} Q(s_t,a') - b_t^q(s_{t-1},a_{t-1})\Big],$$

  and $Q_t(s,a) = Q_{t-1}(s,a)$ for $(s,a) \neq (s_{t-1},a_{t-1})$. Here η represents the learning rate and we need a certain degree of monotonicity for the penalty since we need to use the t-th offset to bound the estimated value error for the first $t-1$ iterations. penalty b_t^q is similarly derived from Prop. 3.1 via seeking a decrement condition on the accumulated value error equation F.6 up to iteration $t-1$ in terms of the offset at step t : $\sqrt{i} b_i^q \leq \sqrt{n} b_n^q$. See Alg. 3 in the appendix for details.

Computational Effort. Evaluating equation 3.8 and equation 3.9 requires estimating the transition matrix at computational cost of $O(TS^2A)$ for score function computation, where T is total sample size. To calculate penalty for N total training steps for batch size b, each step of computing equation 3.8 takes $O(b^3)$ computation steps for evaluation of Stein kernel, and similarly for equation 3.9. With $Nb = T$, one pass over the dataset at the end of N training epochs incurs $O(Nb^3) = O(Tb^2)$, which is improvable by summing over data "coresets" (Chen et al., 2019b;a).

4. Convergence

In this section, we analyze the rate at which the value function sub-optimality [cf. Eq. 2.5] decays with the size T of the dataset, the cardinality of the state and action spaces, respectively $|\mathcal{S}|$ and $|\mathcal{A}|$, and other problem-dependent constants. We impose the technical conditions that are standard (Jin et al., 2021; Yan et al., 2023).

Assumption 4.1. *(Single-Policy Concentrability) With $\mu^\star(s,a)$ as the occupancy measure associated with the op-*

[5] In offline RL, *not* incorporating pessimism results in value overestimation in regions without sufficient data (Kumar et al., 2020).

timal policy $\pi^\star$ [cf. Eq. 2.2], we suppose there exists some constant $C^\star \geq 1$ such that

$$\mu_{\pi^\star}(s,a)/\mu_b(s,a) \leq C^\star \quad \textit{for all } s \in \mathcal{S}, a \in \mathcal{A}. \quad (4.1)$$

Assumption 4.2. *(Uniform Ergodicity) The behavioral policy π_b is stationary, and the transition distribution $\mathbb{P}(\cdot \mid s, \pi_b(s))$ under policy π_b mixes to its induced occupancy $\pi_b(s,a)$ measure at exponential rate:*

$$d_{TV}\left(\mathbb{P}(\cdot \mid s, \pi_b(s)), \mu_b(s_t, a_t)\right) \leq M\beta^t \quad \textit{for all } t. \quad (4.2)$$

Single-policy concentrability holds whenever the behavioral policy π_b is highly exploratory, or otherwise is an expert policy close-to-optimal $\pi^\star$. In the case of an ergodic Markov chain (Assumption 4.2), one can define the mixing time $t_{\text{mix}}(\zeta)$ for any $\zeta \in (0,1)$ as

$$t_{\text{mix}}(\zeta) := \min\left\{ \max_{s_0 \in \mathcal{S}, a_0 \in \mathcal{A}} d_{\text{TV}}\big(\mathbb{P}^t(\cdot \mid s_0, a_0), \mu_b\big) \leq \zeta \right\}, \quad (4.3)$$

where $\mathbb{P}^t(\cdot \mid s_0, a_0)$ denotes the transition distribution of s_t, a_t when starting from the pair (s_0, a_0). Assumption 4.2 is a sufficient condition for a Markov process to have finite mixing time, which is required for any RL method to converge. To our knowledge, all prior analyses of offline RL require finite mixing time, which implicitly requires Assumption 4.2, which may be estimated in practice according to the spectral gap of the transition matrix (Hsu et al., 2019; Dorfman & Levy, 2022).

In subsequent results, $\lesssim$ subsumes constant and poly-logarithmic factors. For the next theorem, we impose Assumption 4.1 only for $C^\star \geq 1$, let $\alpha = \frac{1}{1-\gamma}$ in (3.8) and (3.9), and suppose the reward is bounded $r \in [0,1]$.

Theorem 4.3. *The policy $\hat{\pi}$ returned by Alg. 2 satisfies, under Assumption 4.1:*

$$\mathbb{E}_{\mathcal{D}}\Big[J(\pi^*) - J(\hat{\pi})\Big] \lesssim \frac{SC^\star}{(1-\gamma)^4 T} + \sqrt{\frac{SC^\star}{(1-\gamma)^4 T}}. \quad (4.4)$$

Here $\mathbb{E}_{\mathcal{D}}$ is w.r.t. batch data $\mathcal{D}$ [cf. (2.4)]

See Appendix D for proof. Though the algorithm the offline value iteration is similar to (Rashidinejad et al., 2021), the information-directed penalty results in the following differences: (i) the bound on spurious correlation holds with probability 1 and the penalty of DSD is computed as a V-statistic, which requires a slightly different analysis; (ii) the KSD penalty improves the regret of LCB penalization (Yan et al., 2023) by a factor of $1/(1-\gamma)$; it achieves this sharper regret when transition is in a mixture family, rather than unimodal.

Theorem 4.4. *Let $t_{mix} := t_{mix}(1/4)$ [cf. Eq. 4.3]. Policy $\widehat{\pi}$ returned by Alg. 3, under Assumptions 4.1-4.2, satisfies:*

$$J(\pi^*) - J(\hat{\pi}) \lesssim \sqrt{\frac{C^\star S \iota^2}{T(1-\gamma)^5}} + \frac{C^\star S t_{\text{mix}} \iota}{T(1-\gamma)^2} + \frac{C^\star t_{\text{mix}} \iota^2}{T(1-\gamma)^3},$$

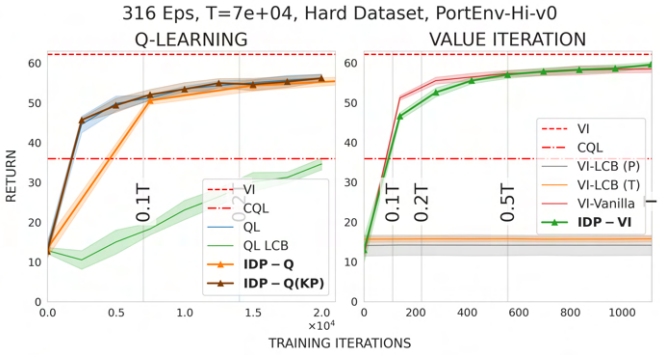

Figure 3: Test cumulative return from offline training for Portfolio Environment under '**Hard**' sampling. For fixed batch size T, IDP performance is competitive with optimal benchmarks, and improves with transition model access.

with probability at least $1-\delta$, with $\iota = \log(ST/\delta)$, when run with learning rate $\eta_t = (H+1)/(H+t)$

See Appendix F for proof. Algorithm 3 is similar to (Yan et al., 2023)[Alg. 1]. The primary distinction lies in using (3.9) in lieu of LCB, which requires modifying the analysis to incorporate the V-statistic of the DSD penalty. This result allows the transitions to be in a mixture family, which is strictly more general than unimodal conditions in prior art, which may explain experimental upsides in volatile environments (Fig. 2). Asymptotic convergence of Algs. 2-3 is implied by dividing the regret by T and sending $T \to \infty$ under attenuating step-size $\eta_T \to 0$.

5. Experiments

Offline RL requires a dataset of sampled transitions, which under single-policy concentrability (Rashidinejad et al., 2021), mandates the data contains sufficient trajectories from optimal policy (separately estimated via value iteration (Sutton & Barto, 1998)). We then create a dataset by sampling three policies for N_{ep} episodes each and concatenating their trajectories: (i) the optimal policy, (ii) a random policy, and an (iii) ϵ-greedy policy (with $\epsilon = 0.3$). Following (Kumar et al., 2020), our experiments span three dataset sampling ratios: '**Easy**' (1:1:1), '**Hard**' (0:1:0.1) and '**Random**' (0:1:0). We spotlight some representative sampling ratios here, with alternates in appendices. We experiment with a Portfolio Optimization task (Neuneier, 1997; Moody & Saffell, 2001), Frozen Lake (Brockman et al., 2016), DeepSea (Osband & Van Roy, 2017a), PriorMDP (Markou & Rasmussen, 2019) – see Appendix G for detailed descriptions, and Appendix G.2.5 for additional experiments with a random walk MDP. To our knowledge, there is little prior experimental analysis of LCB approaches, which motivated us to implement theoretically specified parameters as well as fine-tuned via grid search (see Appendix G.)

Env-Data	$\mathbf{N_{ep}}$	T ($\times 10^3$)	VI	VI-LCB (P)	VI-LCB (Td.)	IDP-VI (Ours)	VI Vanilla	CQL	QL	IDP-Q (Ours)	Q LCB
FL-H	1000	8.8	0.73	0.01	0.17	**0.52**	0.33	0.09	0.69	**0.73**	0.53
PF-H	316	69.0	62.16	14.22	15.69	**59.74**	58.44	35.93	56.13	**56.29**	34.54
D-R	3162	420.0	2.14	0.00	2.27	**2.40**	2.13	0.32	0.03	0.09	0.08
P-R	3162	320.0	29.61	-3.27	12.93	**23.14**	12.90	3.41	15.81	15.50	7.41
R-R	3162	316.4	37.03	1.44	31.04	**35.80**	31.83	4.04	23.03	**26.65**	2.41

Table 2: Results across tasks for fixed batch size & sampling regime. **Env-Data**: Environment (FL: FrozenLake, PF: Portfolio, D: DeepSea, P: Prior MDP, R: Random MDP) - Dataset used (H: '**Hard**', R: '**Random**'), $\mathbf{N_{ep}}$: Number of episodes used to create the dataset, **T**: Total dataset size i.e. number of (s, a, r, s) tuples, **VI** : Value Iteration (Sutton & Barto, 1998) with oracle access to true MDP, **VI-LCB (P)**: VI-LCB (Rashidinejad et al., 2021) w/ paper reported constants, **VI-LCB (Td.)**: VI-LCB (Rashidinejad et al., 2021) w/ tuned constants, **IDP-VI (Ours)**: Our IDP-VI algorithm, **VI Vanilla**: VI-Vanilla with no pessimism penalty term, **CQL**: Conservative Q-learning after 30 epochs (Kumar et al., 2020), **QL**: Q-learning (Watkins & Dayan, 1992), **IDP-Q (Ours)**: Our IDP-Q algorithm, **Q-LCB**: Q-learning with LCB (Shi et al., 2022). Experiments with other data sampling are in Table 6, 7, 8, 9 and 10.

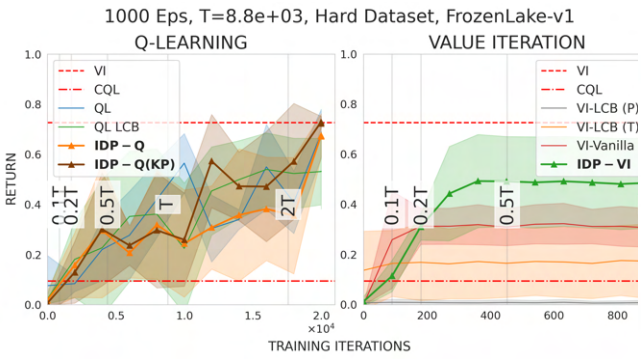

Figure 4: Test performance over training epoch on Frozen Lake (Brockman et al., 2016) a given data set size T with '**Hard**' sampling. Returns using DSD penalty are competitive with optimal benchmarks, is not true of LCB.

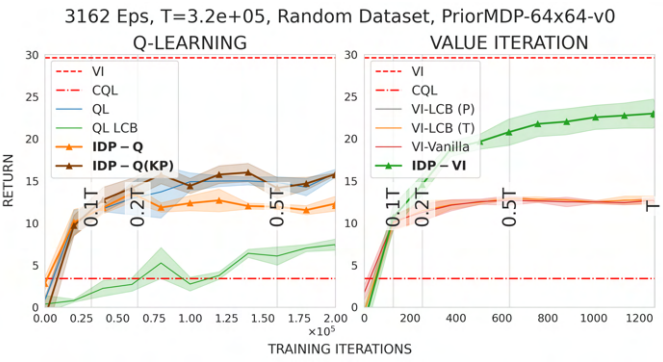

Figure 5: Test performance on PriorMDP (Markou & Rasmussen, 2019) with 64 States and 64 Actions for the '**Random**' dataset. This data set has a large concentrability coefficient [cf. (4.1)], meaning training data is less informative of optimal actions. Vertical lines denote fractions of sample size T during training. For this volatile environment, IDP outperforms LCB and tends toward optimal benchmarks.

5.1. Algorithm 2 (IDP-VI) Performance

Performance comparison is between (i) Value iteration with oracle access to true MDP transitions model and serves as the upper bound on the average return. (ii) VI-LCB, the pessimistic value iteration algorithm with a lower confidence bound based penalty from (Rashidinejad et al., 2021). This algorithm is shown to achieve state-of-the-art performance in terms of theoretical complexity (iii) IDP-VI, our algorithm proposed in this paper (Algorithm 2) and (iv)VI-Vanilla, the Offline Value Iteration algorithm without a penalty term (zero penalty). For VI-LCB we report results using specified constants in the reference as well as a tuned version L_c. We further run a grid search over different values of α in $[0.1, 1, 10]$. Parameter selection details are in Table 4. Results across different environments are in **Figure 3, 4, 5, and 6** on the **right-hand side**, with final reported values in Table 2. Ablation studies with respect to batch size are in Table 6, 7, 8, 9, and 10, respectively. Observe that IDP exhibits advantages in the portfolio environment but is comparable to prior art for DeepSea, corroborating our hypothesis regarding the structural importance of mismatch.

5.2. Algorithm 3 (IDP-Q) Performance

We compare (i) Asynchronous Q-learning; (ii) Q-learning with LCB pessimism (Yan et al., 2023), (iii) IDP-Q (Alg. 3), (iv) CQL (Kumar et al., 2020) which is fundamentally a batch method, and hence report performance after 30 training epochs. We present two variants of IDP-Q: where the transition matrix P is known (denoted as 'Known P') and one where it is estimated. For all approaches we run a grid search over a range of learning rates while also for Q Learning with LCB we search over a range of C_b values and α values for our method IDP-Q. See Table 5 for parameter selections details. Q-learning results are visualized for Portfolio Optimization, Frozen Lake, PriorMDP, and DeepSea, respectively in **Figure 3, 4, 5, and 6** on the **left-hand side** – see also Tables 6, 8, 9, 7, and 2, respectively.

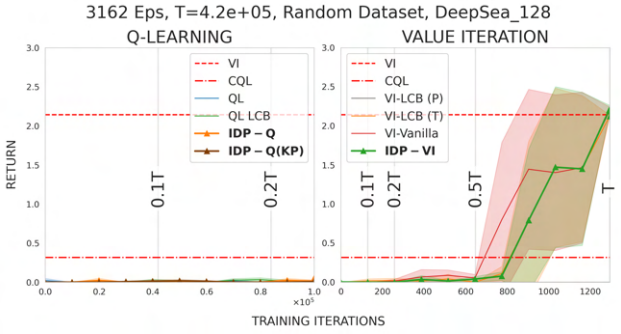

Figure 6: Test performance obtained from training on DeepSea (Osband & Van Roy, 2017a) with 128 States for the '**Random**' dataset. This data set has a large concentrability coefficient [cf. (4.1)], meaning batch data is less informative regarding optimal actions. Vertical lines denote fractions of total sample size T used during training. IDP is comparable to alternatives here.

Experimental upsides of including DSD-based penalty are more pronounced in problem instances with smaller batch data sets (Frozen Lake), and in the presence of multimodality (portfolio optimization, PriorMDP, and random MDP) – see Fig. 2. In Frozen Lake and DeepSea, larger step-sizes were required to obtain competitive performance, which results in more volatile learning than portfolio and PriorMDP.

6. Conclusions

The theory and practice gap in offline tabular RL with pessimism motivated us to adopt an information-theoretic lens to capture mismatch between offline data and the true distribution through DSD. Algorithms based upon it can effectively operate with multimodal distributions both in theory and practice, suggesting that Stein discrepancy may have a broader role to play in offline RL.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Disclaimer

This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JP Morgan Chase & Co and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

Markov decision process (mdp) toolbox for python. `https://github.com/sawcordwell/pymdptoolbox`, 2015.

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

Ardon, L., Vann, J., Garg, D., Spooner, T., and Ganesh, S. Phantom-a rl-driven multi-agent framework to model complex systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 2742–2744, 2023.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Berlinet, A. and Thomas-Agnan, C. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Bernstein, S. On a modification of chebyshev's inequality and of the error formula of laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math*, 1(4):38–49, 1924.

Bhatt, S. and Başar, T. Streisand games on complex social networks. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 1122–1127. IEEE, 2020.

Bhatt, S., Mao, W., Koppel, A., and Başar, T. Semiparametric information state embedding for policy search under imperfect information. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 4501–4506. IEEE, 2021.

Bhatt, S., Fang, G., Li, P., and Samorodnitsky, G. Regret analysis for rl using renewal bandit feedback. In *2022 IEEE Information Theory Workshop (ITW)*, pp. 137–142. IEEE, 2022.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., John, S., Jie, T., and Wojciech, Z. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Chakraborty, S., Bedi, A. S., Koppel, A., Wang, M., Huang, F., and Manocha, D. Steering: Stein information directed exploration for model-based reinforcement learning. In *International Conference on Machine Learning*. PMLR, 2023a.

Chakraborty, S., Bedi, A. S., Tokekar, P., Koppel, A., Sadler, B., Huang, F., and Manocha, D. Posterior coreset construction with kernelized stein discrepancy for model-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6980–6988, 2023b.

Chen, W. Y., Mackey, L., Gorham, J., Briol, F.-X., and Oates, C. Stein points. In *International Conference on Machine Learning*, pp. 844–853. PMLR, 2018.

Chen, W. Y., Barp, A., Briol, F.-X., Gorham, J., Girolami, M., Mackey, L., and Oates, C. Stein point markov chain monte carlo. In *International Conference on Machine Learning*, pp. 1011–1021. PMLR, 2019a.

Chen, W. Y., Barp, A., Briol, F.-X., Gorham, J., Girolami, M., Mackey, L., and Oates, C. Stein point markov chain monte carlo. In *International Conference on Machine Learning*, pp. 1011–1021. PMLR, 2019b.

Cheng, C.-A., Xie, T., Jiang, N., and Agarwal, A. Adversarially trained actor critic for offline reinforcement learning. In *International Conference on Machine Learning*, pp. 3852–3878. PMLR, 2022.

Dong, K., Flet-Berliac, Y., Nie, A., and Brunskill, E. Model-based offline reinforcement learning with local misspecification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Dorfman, R. and Levy, K. Y. Adapting to mixing time in stochastic optimization with markovian data. In *International Conference on Machine Learning*, pp. 5429–5446. PMLR, 2022.

Dwivedi, R. and Mackey, L. Generalized kernel thinning. In *Tenth International Conference on Learning Representations (ICLR 2022)*, 2022.

Gelada, C. and Bellemare, M. G. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3647–3655, 2019.

Gorham, J. and Mackey, L. Measuring sample quality with stein's method. *Advances in Neural Information Processing Systems*, 28, 2015.

Gregory, J., Fink, J., Stump, E., Twigg, J., Rogers, J., Baran, D., Fung, N., and Young, S. Application of multi-robot systems to disaster-relief scenarios with limited communication. In *Field and Service Robotics: Results of the 10th International Conference*, pp. 639–653. Springer, 2016.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*, pp. 2829–2838. PMLR, 2016.

Hao, B. and Lattimore, T. Regret bounds for information-directed reinforcement learning. *Advances in Neural Information Processing Systems*, 35:28575–28587, 2022.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pp. 409–426, 1994.

Hsu, D., Kontorovich, A., Levin, D. A., Peres, Y., Szepesvári, C., and Wolfer, G. Mixing time estimation in reversible markov chains from a single sample path. *The Annals of Applied Probability*, 29(4):2439–2480, 2019.

Jaakkola, T., Jordan, M., and Singh, S. Convergence of stochastic iterative dynamic programming algorithms. *Advances in neural information processing systems*, 6, 1993.

Jaksch, T., Ortner, R., and Auer, P. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010. URL `http://jmlr.org/papers/v11/jaksch10a.html`.

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

Jiao, J., Han, Y., and Weissman, T. Minimax estimation of the $l\_\{1\}$ distance. *IEEE Transactions on Information Theory*, 64(10):6672–6706, 2018.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Karabag, M. O. and Topcu, U. On the sample complexity of vanilla model-based offline reinforcement learning with dependent samples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Kostrikov, I., Fergus, R., Tompson, J., and Nachum, O. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pp. 5774–5783. PMLR, 2021.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

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

Li, G., Shi, L., Chen, Y., Gu, Y., and Chi, Y. Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. *Advances in Neural Information Processing Systems*, 34:17762–17776, 2021a.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. Sample complexity of asynchronous q-learning: Sharper analysis and variance reduction. *IEEE Transactions on Information Theory*, 68(1):448–473, 2021b.

Li, G., Shi, L., Chen, Y., Chi, Y., and Wei, Y. Settling the sample complexity of model-based offline reinforcement learning. *arXiv preprint arXiv:2204.05275*, 2022.

Limmer, Y. and Horvath, B. Robust hedging gans. *Available at SSRN 4489029*, 2023.

Liu, Q., Lee, J., and Jordan, M. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pp. 276–284. PMLR, 2016.

Lu, X., Van Roy, B., Dwaracherla, V., Ibrahimi, M., Osband, I., and Wen, Z. Reinforcement learning, bit by bit, 2021. URL https://arxiv.org/abs/2103.04047.

Lu, X., Van Roy, B., Dwaracherla, V., Ibrahimi, M., Osband, I., Wen, Z., et al. Reinforcement learning, bit by bit. *Foundations and Trends® in Machine Learning*, 16 (6):733–865, 2023.

Markou, S. and Rasmussen, C. E. Bayesian methods for efficient reinforcement learning in tabular problems. *NeurIPS Workshop on Biological and Artificial RL*, 2019.

Matsushima, T., Furuta, H., Matsuo, Y., Nachum, O., and Gu, S. Deployment-efficient reinforcement learning via model-based offline optimization. In *International Conference on Learning Representations*, 2020.

Moerland, T. M., Broekens, J., and Jonker, C. M. Learning multimodal transition dynamics for model-based reinforcement learning. In *29th Benelux Conference on Artificial Intelligence November 8–9, 2017, Groningen*, pp. 362.

Moody, J. and Saffell, M. Learning to trade via direct reinforcement. *IEEE transactions on neural Networks*, 12 (4):875–889, 2001.

Nachum, O., Chow, Y., Dai, B., and Li, L. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

Neuneier, R. Enhancing q-learning for optimal asset allocation. *Advances in neural information processing systems*, 10, 1997.

Nguyen-Tang, T. and Arora, R. Viper: Provably efficient algorithm for offline rl with neural function approximation. In *International Conference on Learning Representations*, 2023.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, pp. 2701–2710, International Convention Centre, Sydney, Australia, 2017a. PMLR.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In *International conference on machine learning*, pp. 2701–2710. PMLR, 2017b.

Puterman, M. L. *Markov Decision Processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Rashidinejad, P., Zhu, H., Yang, K., Russell, S., and Jiao, J. Optimal conservative offline rl with general function approximation via augmented lagrangian. In *The Eleventh International Conference on Learning Representations*, 2022.

Rigter, M., Lacerda, B., and Hawes, N. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35: 16082–16097, 2022.

Russo, D. and Van Roy, B. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

Russo, D. and Van Roy, B. Learning to optimize via information-directed sampling. *Operations Research*, 66 (1):230–252, 2018.

Shetty, A., Dwivedi, R., and Mackey, L. Distribution compression in near-linear time. In *Tenth International Conference on Learning Representations (ICLR 2022).*, 2022.

Shi, L., Li, G., Wei, Y., Chen, Y., and Chi, Y. Pessimistic q-learning for offline reinforcement learning: Towards optimal sample complexity. In *International Conference on Machine Learning*, pp. 19967–20025. PMLR, 2022.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

Strehl, A. L. and Littman, M. L. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*, pp. 856–863, 2005.

Sun, Y., Zhang, J., Jia, C., Lin, H., Ye, J., and Yu, Y. Model-Bellman inconsistency for model-based offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 33177–33194, 23–29 Jul 2023.

Suttle, W. A., Koppel, A., and Liu, J. Occupancy information ratio: Infinite-horizon, information-directed, parameterized policy search. *arXiv preprint arXiv:2201.08832*, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. Cambridge: MIT press, 1998.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Sutton, R. S., Barto, A. G., et al. *Reinforcement Learning: An Introduction*. 2 edition, 2017.

Tamar, A., Mannor, S., and Xu, H. Scaling up robust mdps using function approximation. In *International conference on machine learning*, pp. 181–189. PMLR, 2014.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Tsitsiklis, J. N. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16:185–202, 1994.

Uehara, M., Kallus, N., Lee, J. D., and Sun, W. Refined value-based offline rl under realizability and partial coverage. *arXiv preprint arXiv:2302.02392*, 2023.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021a.

Xie, T., Jiang, N., Wang, H., Xiong, C., and Bai, Y. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021b.

Xie, T., Foster, D. J., Bai, Y., Jiang, N., and Kakade, S. M. The role of coverage in online reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Xu, T., Wang, Y., Zou, S., and Liang, Y. Provably efficient offline reinforcement learning with trajectory-wise reward. *arXiv preprint arXiv:2206.06426*, 2022.

Yan, Y., Li, G., Chen, Y., and Fan, J. The efficacy of pessimism in asynchronous q-learning. *IEEE Transactions on Information Theory*, 2023.

Yang, J., Liu, Q., Rao, V., and Neville, J. Goodness-of-fit testing for discrete distributions via stein discrepancy. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5561–5570. PMLR, 10–15 Jul 2018.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Zhang, S. and Jiang, N. Towards hyperparameter-free policy selection for offline reinforcement learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=9RFGrW9z9te`.

Type	Horizon	Policy	Mismatch Concept	Penalty	Experiments	Rate	Computational Complexity	Ref
Model-Based	Finite	Tabular	DD, CC	AH-LCB	No	$\mathcal{O}\left(\sqrt{\frac{H^5SC^\star}{T}}\right)$	$T+HSA^2$	(Xie et al., 2021b)
			DD, CC	B-LCB	Yes	$\mathcal{O}\left(\sqrt{\frac{H^4SC^\star}{T}}\right)$	$T+HSA^2$	(Xie et al., 2021b; Li et al., 2022)
		Parameterized	DD, CC	BS		$\mathcal{O}\left(\frac{H^{3/2}d}{\sqrt{T}}\right)$		(Nguyen-Tang & Arora, 2023)
			DD, CC	DD	No	$\mathcal{O}\left(\frac{H^2d}{\sqrt{T}}\right)$		(Jin et al., 2021; Xu et al., 2022)
	Infinite	Tabular	DD, OO	HT-RM	Yes	$\mathcal{O}\left(\frac{1+\mu_{\pi^*,\mathcal{D}}}{\sqrt{T}(1-\gamma)^2}\right)$	TAS^2	(Kidambi et al., 2020)
			DD	HUE	Yes	None	TAS^2	(Yu et al., 2020)
			DD, CC	AH-LCB	No	$\mathcal{O}\left(\sqrt{\frac{SC^\star}{(1-\gamma)^5T}}\right)$	$(T+S^2A)\log(T)/(1-\gamma)$	(Rashidinejad et al., 2021)
			DD, CC	B-LCB	Yes	$\mathcal{O}\left(\sqrt{\frac{SC^\star\log(T)}{(1-\gamma)^3T}}\right)$	$(T+S^2A)\log(T)/(1-\gamma)$	(Li et al., 2022)
			DD,CC	KSD	Yes	$\mathcal{O}\left(\sqrt{\frac{SC^\star\log(T)}{(1-\gamma)^4T}}\right)$	$(T^3SA+S^2A)\log(T)/(1-\gamma)$	This Work
		Parameterized						✗
Model-Free	Finite	Tabular	DD, CC	AH-LCB	No	$\mathcal{O}\left(\sqrt{\frac{H^4SC^\star}{T}}\right)$	THA	(Shi et al., 2022)
		Parameterized	MCC	L	No	$\mathcal{O}(\sqrt[8]{\frac{B_\omega B_\mathcal{F} A}{T}})$[6]		(Cheng et al., 2022; Uehara et al., 2023)
		Tabular	DD, CC	AH-LCB	No	$\mathcal{O}(\sqrt{\frac{SC^\star}{(1-\gamma)^5T}})$	TA	(Yan et al., 2023)
			DD	KSD	Yes	$\mathcal{O}(\sqrt{\frac{SC^\star\log(T)}{(1-\gamma)^5T}})$	T^4+TA	This Work
	Infinite	Parameterized	DD	Model-based	Yes	No Rate		(Kumar et al., 2019; 2020; Matsushima et al., 2020; Yu et al., 2021; Kostrikov et al., 2021)
			DD,CC	AH-LCB	No	$\mathcal{O}(\sqrt{\frac{d^3H^4\log(HTd)}{T}})$		(Jin et al., 2021)[7]
			BC	BE	No	$\mathcal{O}\left(\frac{1}{1-\gamma}\sqrt{\frac{\log(\lvert\mathcal{F}\rvert\lvert\Pi\rvert)}{T}}\right)$		(Xie et al., 2021a)

Table 3: A comparison of recent offline RL methods according to whether they are model-based or model-free, the horizon, policy mapping, definition of distribution mismatch (Mismatch Concept), and algorithmic mechanism for correcting for it in the form of a pessimistic penalty (Penalty). Distribution Mismatch Concepts: Distributional Distance (dd), Occupancy Overlap (OO) $\mu_{\pi^*,\mathcal{D}}$, Concentrability Coefficient (CC) $C^\star$, Bellman Consistency (BC), Sequential Extrapolation Coefficient (SEC), Model-free concentrability coefficient (MCC). Pessimistic Penalties: Azuma-Hoeffding-based and Bernstein-based Lower Confidence-Bound (AH-LCB) and B-LCB), Hitting-Time Reward Max (HTRM), Heuristic Uncertainty Estimate (HUE), Lagrangian-based (L), Bootstrap (BS), Bellman Error (BE), Model-Based, Kernelized Stein Discrepancy (KSD). Horizon Length: Finite H or Infinite. *All non-highlighted entries that contain rate analyses require unimodal transitions, which is contrast to this work. Our work may be seen in the spirit of model-based augmentation of model-free methods, e.g., (Gu et al., 2016; Yu et al., 2021).*

A. Expanded Literature Review

The major takeaway is that the rates established in this work improve upon previous results for tabular settings in the infinite-horizon case, for both model-based (value iteration) and model-free (Q-learning) approaches in key ways, specifically by: (i) alleviating any implicit requirement of unimodality; and (ii) refining the value iteration regret in (Rashidinejad et al., 2021) by a factor of $1/(1-\gamma)$. To enable these results, we introduce a novel concept of pessimistic penalty, KSD, which is inspired by information-directed sampling (Russo & Van Roy, 2018).

The gap between our rates and the best available (Li et al., 2022)-(Yan et al., 2023) exists as those approaches present algorithms that require multiple time-sales and employ Bernstein concentration bounds/variance reduction, which hinge upon a different error decomposition than that which is studied here. The improved properties of Stein discrepancies relative to Azuma-Hoeffding concentration bounds were relatively simpler to understand and yield simpler algorithms to implement, and thus a better template upon which to introduce DSD. It is possible to employ DSD together with Bernstein/variance reduction to close the gap with (Li et al., 2022)-(Yan et al., 2023) by introducing multiple time-scales, but that is deferred to future work. Moreover, even these sharper rates represent the mismatch as unimodal, whereas we allow it to belong to a mixture family.

On top of these conceptual aspects, this work contributes one of the first analyses that contrasts theoretical with experimental performance in the tabular setting. While there has been extensive numerical evaluation of offline RL methods and possible perception is that it is a mature field, at least experimentally setting, such studies are mostly disconnected from the RL foundations perspective.

B. Technical Preliminaries

B.1. Markov Decision Processes

In this subsection, we expand upon some of the background required for Sec. 2. Value iteration and Q learning operate upon the fact that the value equation 2.1 (respectively, action-value) starting from one state may be decomposed into the

one step reward plus the value (action-value) starting from another, stated as:

$$V^{\pi}(s) = r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} V^{\pi}(s') P(s' \mid s, \pi(s)) \tag{B.1}$$

for all $s \in \mathcal{S}$, or respectively $(s, a) \in \mathcal{S} \times \mathcal{A}$. These expressions are known as Bellman's equations. The right-hand side of equation B.1 defines a Bellman evaluation operator $\mathscr{B}^{\pi} : \mathcal{S} \to \mathcal{S}$ over functions that map state space $\mathcal{S}$ to the reals, i.e., $V : \mathcal{S} \to \mathbb{R}$:

$$(\mathscr{B}^{\pi} V)(s) = r(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} V^{\pi}(s') P_{s', s, \pi(s)}. \tag{B.2}$$

Further define the Bellman optimality operator over the space of Q functions as:

$$(\mathscr{B}^{\star} Q)(s, a) := r(s, a) + \gamma \sum_{s' \in \mathcal{S}} \max_{a'} Q(s', a') P_{s', s, a'} \ . \tag{B.3}$$

Value iteration may be defined by iteratively applying equation B.1 and computing the maximum over $a \in \mathcal{A}$

$$V_t(s) = \max_{a \in \mathcal{A}} \mathbb{E}_{s' \sim \mathbb{P}(\cdot \mid s, a)} \left[r(s, a) + V_{t-1}(s') \right] \tag{B.4}$$

Alternatively, classical asynchronous Q learning (Jaakkola et al., 1993; Tsitsiklis, 1994) operates by sequentially applying stochastic approximation to estimate the Bellman optimality operator equation B.3 based on trajectory data $\{s_{u-1}, a_{u-1}, r_{u-1}, s_u\}_{u=1}^{\infty}$:

$$Q_t(s_{t-1}, a_{t-1}) = (1 - \eta) Q_{t-1}(s_{t-1}, a_{t-1}) + \eta [r(s_{t-1}, a_{t-1}) + \gamma \max_{a'} Q(s_t, a')] \ , \tag{B.5}$$

$$Q_t(s, a) = Q_{t-1}(s, a) \text{ for all } (s, a) \neq (s_{t-1}, a_{t-1}) \ . \tag{B.6}$$

Here η is a step-size (learning rate) possibly diminishing in terms of the number of visits n to state-action pair s_{t-1}, a_{t-1} prior to time t (in which case we which case we write η_t).

B.2. Kernelized Stein Discrepancy

IPM tracks the deviation between a baseline distribution q and an unknown target p: $d_{\mathcal{F}}(q, p) = \sup_{f \in \mathcal{F}} |\mathbb{E}_q[f(X)] - \mathbb{E}_p[f(X)]|$, where the supremum is over a class of real-valued test functions $f \in \mathcal{F}$, i.e., $f : \mathcal{X} \to \mathbb{R}$ for some Euclidean space $\mathcal{X} \subset \mathbb{R}^d$. By adjusting the function class $\mathcal{F}$ (Sriperumbudur et al., 2010), one recovers Total variation, Wasserstein, among others. However, the impediment to evaluating an IPM is its integration under the true distribution p, which is intractable in offline RL.

Stein's method alleviates this issue by restricting the class of distributions $\mathcal{F}$ to functions such that $\mathbb{E}_p[f(X)] = 0$. Building upon this idea, (Liu et al., 2016) develops a tractable way to evaluate the IPM by restricting distributions to the Stein class, associated with a reproducing kernel Hilbert space (RKHS) over Stein kernels (Berlinet & Thomas-Agnan, 2011). In this case, the IPM may be evaluated in expectation with respect to the Stein kernel, which is called the Kernelized Stein Discrepancy (KSD) (Gorham & Mackey, 2015). Stein's method relies on the fact that two smooth densities $p(x)$ and $q(x)$ in function class $\mathcal{F}$ are identical if and only if they satisfy the Stein's identity:

$$\max_{f \in \mathcal{F}} \left(\mathbb{E}_p[\mathbb{S}_q(x) f(x) + \nabla_x f(x)] \right)^2 = 0, \tag{B.7}$$

where $\mathbb{S}_q(x)$ denotes the score function of $q(x)$ given by $\mathbb{S}_q(x) = \nabla_x \log q(x)$. As an example, Stein's identity in equation B.7 holds for smooth functions f lying in the Stein class of p, i.e., it is smooth and satisfies $\int_x \nabla_x (f(x) p(x)) dx = 0$. Hence, for any function f in the Stein class of p, we define a Stein operator $\mathcal{A}_p$ of p, for which $\mathbb{E}_p[\mathcal{A}_p f(x)] = 0$. Based upon this notion, define the Stein discrepancy between p and q (Liu et al., 2016):

$$\text{KSD}^2(p, q) = \max_{f \in \mathcal{F}} \left(\mathbb{E}_p[\mathbb{S}_q(x) f(x) + \nabla_x f(x)] \right)^2, \tag{B.8}$$

However, this definition requires solving a variational optimization. This issue may be alleviated through a tractable modification introduced in (Liu et al., 2016):

$$\text{KSD}^2(p, q) = \mathbb{E}_{x, x' \sim p} \big[u_q(x, x') \big], \tag{B.9}$$

where $u_q(x, x')$ is the Stein kernel defined as

$$u_q(x,x') := \mathbb{S}_q(x)^\top \kappa(x,x') \mathbb{S}_q(x') + \mathbb{S}_q(x)^\top \nabla_{x'} \kappa(x,x') + \nabla_x \kappa(x,x')^\top \mathbb{S}_q(x') + \texttt{trace}(\nabla_{x,x'} \kappa(x,x')), \tag{B.10}$$

In equation B.10, $\kappa(x, x')$ is the base kernel such as a Gaussian or polynomial, meaning that this approximation imposes smoothness on the target distribution q in that it hypothesizes it belongs to a RKHS associated with base kernel κ and we can evaluate its score function $\mathbb{S}_q$, which holds when the target measure belongs to a mixture family. Moreover, `trace` denotes the trace of a square matrix (the sum of the elements on its main diagonal), in the case, the Jacobian of the kernel with respect to its inputs. In continuous space, it would be natural to select $p = \mathbb{P}(\cdot \mid s, \pi_b(s))$ (transition dynamics corresponding to the empirical measure defined by offline sampled data set $\mathcal{D}$ generated by behavioral policy π_b) and $q = \mathbb{P}$ (transition dynamics corresponding to the true MDP). In this case, then, KSD would empower us to evaluate the distance $\text{KSD}(\mathbb{P}(\cdot \mid s, \pi_b(s)), \mathbb{P}(\cdot \mid s, a))$ under the hypothesis we had access to the score function of $\mathbb{P}(\cdot \mid s, a)$, denoted as $\mathbb{S}_{\mathbb{P}(\cdot|s,a)}$, which is abbreviated as $\mathbb{S}_{\mathbb{P}}$.

We developed machinery in discrete space in Sec. 3.1 which is based upon methods that were originally developed for continuous space and are included here for completeness as well as intuition-building. In this case, the score function is defined terms of the probability mass function associated with the (s, a)-th row of the transition matrix $P_{s,a}$ rather than transition kernel $\mathbb{P}$, and kernels must be defined in terms of discrete input sequences, rather than those that appear in classic (continuous-valued) nonparametric statistics (Berlinet & Thomas-Agnan, 2011).

C. Details of Algorithm Execution

In this section, we expand upon the technical motivations, derivations, and execution of information-directed pessimism for value iteration [cf. equation 3.11] and Q-learning [cf. equation 3.12]. In particular, we expand upon the practical evaluation of the pessimistic penalty $b_t(s,a)$ in these expressions, as well as present their pseudo-code.

C.1. Score function implementation using 'estimated' True model

Since DSD is a one-sample test, to evaluate the discrepancy of a dataset with respect to the unknown true transition model, we need to make an implicit assumption that the score function of the true model is known. This may be an overly restrictive departure from the standard offline RL setting, where no assumptions on the model is made. To relax this requirement in our setting, we consider an approach to estimating the true model from data, and using this estimate to compute the score function. Doing so allows us to employ DSD machinery using empirical rather than true score functions. We detail this approach next.

In offline RL, the values at states having low data representation (in the dataset) are typically overestimated. This necessitates adding a penalty that reduces the estimated values. Consider an ordering of the states $\mathcal{S} := \{s'_1, s'_2, \cdots, s'_n\}$ based on values as $V_{t-1}(s'_1) \geq V_{t-1}(s'_2) \geq \cdots \geq V_{t-1}(s'_n)$. We see in Algorithm 1 (minor modification of the algorithm in (Jaksch et al., 2010)) that for a given estimate of the value function at time t, the output is

$$\tilde{P}^t_{s,a}(s' \mid s,a) := \underset{\tilde{p}(\cdot)\in\mathcal{P}(s,a)}{\arg\max} \sum_{s'\in\mathcal{S}} \tilde{p}(s')V_{t-1}(s').$$

This implies that the estimated transition probabilities are such that, more weight is allocated to states with maximum values at the expense of transition probabilities of states with smaller values. Considering that these values already incorporate a penalty, we have a new transition function in the convex polytope $\mathcal{P}(s,a) \subset \Delta(\mathcal{S}\times\mathcal{A})$ that best represents the offline data.

Lemma C.1 ((Jaksch et al., 2010)). *Let $n_t(s,a) := |\{\tau : \tau \leq t,\ s_\tau = s, a_\tau = a\}|$ denote the number of given state-action occurrences for a given trajectory $\{s,a\}_{\tau\leq t}$ of length t. For a given empirical distribution $\hat{P}^t(\cdot \mid s,a)$, let $\mathcal{M}_t$ denote the set of MDPs which are a specified total-variation distance away from the empirical distribution:*

$$\|\tilde{P}^t(\cdot \mid s,a) - \hat{P}^t(\cdot \mid s,a)\|_1 \leq \sqrt{\frac{14S\log(2At/\delta)}{\max\{1, n_t(s,a)\}}}. \tag{C.1}$$

i.e., MDPs in set $\mathcal{M}_t$ have transition functions $\tilde{P}^t(\cdot \mid s,a)$ at most $\sqrt{\frac{14S\log(2At/\delta)}{\max\{1,n_t(s,a)\}}}$ away in total-variation distance. Then the following holds true:

$$\mathbb{P}\{M \notin \mathcal{M}_t\} < \frac{\delta}{15t^6}.$$

C.2. Information-Directed Offline Value Iteration.

The idea of the value iteration algorithm is to begin with the basic procedure of value iteration with a pessimistic penalty. Initially, the value and Q function estimates are null for all states or state-action pairs. We estimate the transition matrix of the dataset. Then, we slice $\mathcal{D}$ along each state-action pair for evaluation of the conditional DSD [cf. equation 3.8]. This penalty is then subtracted from a value-iteration style update [cf. equation B.4] and the action-value function at the next iteration is updated according to the resultant fixed point equation. The value function estimates and policies are then evaluated as the respective maximum and maximizing argument of the Q-function. This procedure is summarized as Algorithm 2.

C.3. Information-Directed Offline Q-Learning

Q-learning proceeds by repeatedly executing stochastic approximations of the Bellman operator equation B.3. For the offline setting, we incorporate the penalty equation 3.9 into the pessimistic update given in equation 3.12. We repeatedly execute this update, with the penalty calibrated to the number of visits to state-action pair (s,a) prior to iteration t. This procedure is summarized in pseudo-code as Algorithm 3.

Algorithm 1 Estimating True Transition Model

Input: Estimates $\hat{P}^t(\cdot \mid s,a)$, distance $d(s,a) := \sqrt{\frac{14S\log(2Ak/\delta)}{\max\{1,N_t(s,a)\}}}$ with $k = \frac{Nt}{T+1}$ and $N_t(s,a) = \sum_{i=1}^{t} m_i(s,a)$, and states $\mathcal{S} := \{s'_1, s'_2, \cdots, s'_n\}$ sorted as $V_{t-1}(s'_1) \geq V_{t-1}(s'_2) \geq \cdots \geq V_{t-1}(s'_n)$.
Output: $\tilde{P}^t(s' \mid s,a) := \arg\max_{\tilde{p}(\cdot)\in\mathcal{P}(s,a)} \sum_{s'\mathcal{S}} \tilde{p}(s')V_{t-1}(s')$, which is over the set of transition probabilities satisfying condition equation C.1.
Set

$$\tilde{P}^t(s'_1) := \min\left\{1, \hat{P}^t(s'_1 \mid s,a) + \frac{d(s,a)}{2}\right\}, \text{and}$$
$$\hat{P}^t(s'_j) := \hat{P}^t(s'_j \mid s,a) \text{ for all states } s'_j \text{ with } j > 1.$$

Set $l := n$
while $\sum_{s'_j\in\mathcal{S}} \tilde{P}^t(s'_j) > 1$ **do**

$$\text{Reset } \tilde{P}^t(s'_l) := \max\left\{0, 1 - \sum_{s'_j \neq s'_l} \tilde{P}^t(s'_j)\right\}$$
$$\text{Set } l := l - 1$$

Algorithm 2 **IDP-VI** Information-Directed Pessimistic Value Iteration

Input: Pessimism coefficient α, offline data set $\mathcal{D}$, discount factor γ
$K := \frac{\log T}{1-\gamma}$.
Initialize $V_0(s) = 0,\ Q_0(s,a) = 0$ and $\pi_0(s) = \arg\max_a m_0(s,a)$
for all $k = 1, \cdots, K$ **do**
$m(s,a) := \sum_{i=1}^{T} \mathbb{1}((s_i,a_i) = (s,a))$ based on dataset $\mathcal{D}$
Estimate the empirical transition $\hat{P}_{s,a}$ and obtain rewards $\hat{r}(s,a)$ elements using the dataset $\mathcal{D}$
Estimate the true model $\tilde{P}_{s,a}$ using Algorithm 1
Compute the penalty using $\mathcal{D}^{(s,a)}$ as

$$b^v(s,a) := \begin{cases} \alpha \cdot \sqrt{\frac{1}{m^2(s,a)} \sum_{s',\check{s}\sim\mathcal{D}^{(s,a)}} \kappa_{\tilde{P}_{s,a}}(s',\check{s})}, & \text{if } (s,a) \in \mathcal{D}^{(s,a)} \\ \max_{(s,a)}\left\{\alpha \cdot \sqrt{\frac{1}{m^2(s,a)} \sum_{s',\check{s}\sim\mathcal{D}^{(s,a)}} \kappa_{\tilde{P}_{s,a}}(s',\check{s})}\right\}, & \text{otherwise.} \end{cases}$$

Update pessimistic Q-function estimate as follows:
for all $(s,a) \in (\mathcal{S}\times\mathcal{A})$ **do**

$$Q_k(s,a) = \hat{r}(s,a) - b^v(s,a) + \gamma\hat{P}_{s,a} \cdot V_{k-1},$$

end for
end for
return Value function estimate $V_K(s)$ and associated policy $\pi_K(s) = \arg\max_a Q_K(s,a)$

Algorithm 3 IDP-Q Information-Directed Pessimistic Asynchronous Q Learning

Input: Pessimism coefficient α, offline data set $\mathcal{D}$, initial state s_0, initial value $V(s_0) = 0$.
Initialize: $Q_0(s,a) = 0, V_0(s) = 0, b_0^q(s,a) = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}, H = \left\lceil \frac{4}{1-\gamma} \log \frac{ST}{\delta} \right\rceil$.
for all $t = 1, \cdots, T$ **do**
Sample $(s_{t-1}, a_{t-1}, r_{t-1}, s_t)$ from offline data set $\mathcal{D}$.
Estimate the true model $\tilde{P}^t_{s_{t-1},a_{t-1}}$ using Algorithm 1
Update the pessimistic offset:

$$b_t^q(s_{t-1}, a_{t-1}) = \max\left\{ \sqrt{\frac{n_t(s_{t-1}, a_{t-1}) - 1}{n_t(s_{t-1}, a_{t-1})}} b_{t-1}^q(s_{t-1}, a_{t-1}), \alpha \sqrt{\mathbb{E}_{s', \check{s} \sim \mathcal{D}_{s_{t-1}, a_{t-1}}} \left[\kappa_{\tilde{P}^t_{s,a}}(s', \check{s}) \right]} \right\},$$

where $n_t(s_{t-1}, a_{t-1})$ is the number of times (s_{t-1}, a_{t-1}) has been visited prior to iteration t. For all $(s,a) \neq (s_{t-1}, a_{t-1})$, set $Q_t(s,a) = Q_{t-1}(s,a)$.
Update pessimistic Q-function estimate according to equation 3.12:

$$Q_t(s_{t-1}, a_{t-1}) = (1 - \eta_t) Q_{t-1}(s_{t-1}, a_{t-1}) + \eta_t [r(s_{t-1}, a_{t-1}) + \gamma \max_{a'} Q(s_t, a') - b_t^q(s_{t-1}, a_{t-1})] \,,$$

where $\eta_t = \frac{H+1}{H + n_t(s,a)}$.
Update value function:

$$V_t(s_{t-1}) = \max\left\{ \max_{a \in \mathcal{A}} Q_t(s_{t-1}, a), V_{t-1}(s_{t-1}) \right\},$$

and $V_t(s) = V_{t-1}(s)$ for all $s \neq s_{t-1}$.
end for
return policy $\hat{\pi}(s) = \arg\max_a Q_T(s,a)$.

D. Technical Lemmas, Propositions, and Their Proofs

D.1. Proof of Proposition 3.1

Recall that for any function $f : \mathcal{S} \to \mathbb{R}$ and distribution $P(\cdot \mid s,a)$, abbreviated as P, the discrete Stein operator is defined as (Yang et al., 2018):

$$\mathbb{A}_{P_{s,a}} f(s') = \mathbb{S}_{P_{s,a}}(s') f(s') - \Delta^* f(s').$$

We have $\mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\mathbb{A}_{P_{s,a}} f(s') \right] = 0$ using difference Stein's Identity. Let $\mathcal{V}$ be the space of real-valued functions on the state space $\mathcal{S}$. For each $V \in \mathcal{V}$, there exists $f_V(s') \in \mathcal{F}_P$ that satisfies the Stein's equation:

$$V(s') - \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[V(s') \right] = \mathbb{A}_{P_{s,a}} f_V(s'), \tag{D.1}$$

where $\mathcal{F}_P$ is the set of real-valued functions that satisfy Stein's identity w.r.t distribution $P(\cdot \mid s,a)$. Consider bounding $|(\mathbb{P}(\cdot \mid s,a) - \mathbb{P}(\cdot \mid s, \pi_b(s)) \cdot V|$, the mean deviation of the value function under the empirical and true model (where we subsequently abbreviate $\mathbb{P}(\cdot \mid s, \pi_b(s))$ by its associated element of the empirical transition matrix as $\hat{P}_{s,a}$, and similarly for $P_{s,a}$ with respect to $\mathbb{P}(\cdot \mid s,a)$) as in Sec. 3. Then we may write:

$$|(P_{s,a} - \hat{P}_{s,a}) \cdot V| \leq \sup_{V \in \mathcal{V}} \left| \mathbb{E}_{s' \sim \hat{P}_{s,a}}[V(s')] - \mathbb{E}_{\check{s} \sim P_{s,a}}[V(\check{s})] \right|. \tag{D.2}$$

Substituting equation D.1, after taking expectation w.r.t to the collection of empirical measures $\hat{P}$ associated with state-action pairs $(s, \pi_b(s))$ in the offline data set $\mathcal{D}$, in the above inequality

$$\sup_{V\in\mathcal{V}}\left|\mathbb{E}_{s'\sim\hat{P}}[V(s')] - \mathbb{E}_{\check{s}\sim P}[V(\check{s})]\right| = \sup_{V\in\mathcal{V}}\left|\mathbb{E}_{s'\sim\hat{P}_{s,a}}[\mathbb{A}_{P_{s,a}}f_V(s')]\right| \tag{D.3}$$

$$= \sup_{f\in\mathcal{H},\ \|f\|_{\mathcal{H}}\leq\frac{1}{1-\gamma}}\left|\mathbb{E}_{s'\sim\hat{P}_{s,a}}[\mathbb{A}_{P_{s,a}}f(x)]\right| \tag{D.4}$$

$$= \frac{1}{1-\gamma}\sqrt{\mathbb{E}_{s',\check{s}\sim\mathcal{D}^{(s,a)}}\left[\kappa_{P_{s,a}}(s',\check{s})\right]}, \tag{D.5}$$

which follows using the definition of kernelized discrete Stein discrepancy. □

D.2. Auxiliary Lemmas for Value Iteration Result

With this result in hand, we introduce a technical lemma required for the analysis of Algorithm 2. To do so, we introduce a transient variant of the discounted state-action occupancy measure defined in Sec. 2. More specifically, let the initial distribution induced by policy π be denoted as $\rho^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, where (s,a) element is equal $\rho(s)\pi(a|s)$. Similarly, let the transition matrix induced by the policy π be given as $P^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$, whose $(s,a)\times(s',a')$ element is equal to $P(s'|s,a)\pi(a'|s')$. In this section, without loss of generality, assume that the penalty at iteration t is given as

$$b_k^v(s,a) = \alpha\cdot\sqrt{\frac{1}{m^2(s,a)}\sum_{s',\check{s}\sim\mathcal{D}^{(s,a)}}\kappa_{\tilde{P}_{s,a}}(s',\check{s})}. \tag{D.6}$$

Note that this still satisfies the bound on DSD in Proposition 3.1.

Lemma D.1. *Suppose Proposition 3.1 holds. For an arbitrary policy π, we have for all $t>0$:*

$$J(\pi) - J(\pi_t) \leq \frac{\gamma^t}{1-\gamma} + 2\sum_{i=1}^{k}\mathbb{E}_{\nu^\pi_{k-i}}[b_i^v(s,a)],$$

where $\nu_k^\pi := \gamma^k\rho^\pi(P^\pi)^k$ for $k\geq 0$ and $J(\pi) := \mathbb{E}_{s\sim\rho}[V_\pi(s)]$.

Proof. First, note that the penalty holds on an event that occurs with probability 1. By design of the algorithm, $V_{t-1}\leq V_t\leq V^{\pi_t}\leq V^*$ (cf. equation D.6). The result follows using the arguments in (Rashidinejad et al., 2021)(Lemma 2). □

Lemma D.2. *(Jiao et al., 2018) Let $n\sim$ Binomial(N,p). For any $k\geq 0$, there exists a constant c_k depending only on k such that*

$$\mathbb{E}\Big[\frac{1}{(n\vee 1)^k}\Big] \leq \frac{c_k}{(Np)^k},$$

where $c_k = 1 + k2^{k+1} + k^{k+1} + k(\frac{16(k+1)}{r})^{k+1}$.

Let $\text{MMD}_{\kappa_q}(p,q) := \mathbb{E}[\kappa_q(x,x') + \kappa_q(y,y') - 2\kappa_q(x,y')]$ denotes the MMD between p and q evaluated using the Stein kernel κ_q, with x, x' and y, y' drawn i.i.d from p and q.

Lemma D.3. *Let $|\mathcal{D}| = T$ and $\delta\in(0,1)$. For the penalty with probability $1-\delta$*

$$b_t^v(s,\pi(s)) := \alpha\sqrt{\frac{1}{m^2(s,a)}\sum_{s',\check{s}\sim\mathcal{D}^{(s,a)}}\kappa_{\tilde{P}^T_{s,a}}(s',\check{s})} \leq \alpha\sqrt{\frac{\log(1/\delta)}{T}}.$$

Proof. Convergence rate was established for Maximum Mean Discrepancy (MMD) in (Gretton et al., 2012)(Theorem 8). DSD can be treated as a MMD with a special function class associated with the test functions such that Stein's identity holds, i.e., MMD may be evaluated with the Stein kernel (κ) to obtain DSD. Let $\hat{P}_t$ and $\tilde{P}_t$ denote the empirical transition matrix and the estimated true transition matrix. Note the limiting matrices of both $\hat{P}_t$ and $\tilde{P}_t$ are the same, and equal to P

from Lemma C.1. Using similar arguments as in (Gretton et al., 2012)(Theorem 8), we have that with probability $1-\delta$, using the V-statistic of DSD,

$$b_t^v(s,\pi(s)) := \alpha \cdot \text{MMD}_{\kappa_{P_t}}(\hat{P}_t, \tilde{P}_t) = \alpha|\text{MMD}_{\kappa_{\hat{P}_t}}(\hat{P}_t, \tilde{P}_t) - \text{MMD}_{\kappa_{\tilde{P}_{s,a}}}(P,P)| \\ \lesssim \alpha\sqrt{\frac{\log(1/\delta)}{t}} (:= \mathcal{E}_{\mathbf{DSD}}) \tag{D.7}$$

□

E. Proof of Theorem 4.3

Proof. Consider the following decomposition for any expert policy π using equation D.7:

$$\mathbb{E}_{\mathcal{D}}\Big[\sum_s \rho(s)\Big[V_\pi(s) - V_{\pi_K}(s)\Big]\Big] \\ = \mathbb{E}_{\mathcal{D}}\Big[\sum_s \rho(s)\Big[V_\pi(s) - V_{\pi_K}(s)\Big]\mathbb{1}\Big\{m(s,\pi(s)) = 0\Big\}\Big] =: \mathcal{T}_1 \\ + \mathbb{E}_{\mathcal{D}}\Big[\sum_s \rho(s)\Big[V_\pi(s) - V_{\pi_K}(s)\Big]\mathbb{1}\Big\{m(s,\pi(s)) \geq 1\Big\}\mathbb{1}\{\mathcal{E}_{DSD}\}\Big] =: \mathcal{T}_2 \\ + \mathbb{E}_{\mathcal{D}}\Big[\sum_s \rho(s)\Big[V_\pi(s) - V_{\pi_K}(s)\Big]\mathbb{1}\Big\{m(s,\pi(s)) \geq 1\Big\}\mathbb{1}\{\mathcal{E}^c_{DSD}\}\Big] =: \mathcal{T}_3$$

where $\rho(s)$ is any prior distribution over states $s \in \mathcal{S}$. (i) $\mathcal{T}_3$ can be trivially upper bounded by $\frac{\delta}{1-\gamma}$ using the bounded reward assumption and $\mathcal{T}_1$ can be bounded by $\frac{4C^\star S(K+1)^2}{9(1-\gamma)^2T}$ using the same arguments as in (Rashidinejad et al., 2021).
(ii) We now proceed to bound $\mathcal{T}_2$. Using Lemma D.1,

$$\mathcal{T}_2 \leq \frac{\gamma^K}{1-\gamma} + 2\sum_{k=1}^K \mathbb{E}_{\mathcal{D},\nu^\pi_{K-k}}\Big[b_k(s,\pi(s))\mathbb{1}\Big\{m(s,\pi(s)) \geq 1\Big\}\Big].$$

By definition, $\sum_{k=0}^\infty \nu_k^\pi = \frac{\mu_\pi}{1-\gamma}$. Using Lemma D.3 and Lemma 14 of (Rashidinejad et al., 2021),

$$\mathbb{E}_{\mathcal{D},\nu^\pi_{K-k}}\Big[b_k^v(s,\pi(s))\mathbb{1}\Big\{m(s,\pi(s)) \geq 1\Big\}\Big] \leq \mathbb{E}_{\mathcal{D},\nu^\pi_{K-k}}\Big[\frac{\alpha\log(1/\delta)}{\sqrt{m(s,\pi(s))}}\mathbb{1}\Big\{m(s,\pi(s)) \geq 1\Big\}\Big] \\ \leq \mathbb{E}_{\nu^\pi_{K-k}}\left[16\alpha\log(1/\delta)\sqrt{\frac{1}{T\mu(s,\pi(s))}}\right].$$

Using Lemma D.2, we have

$$\sum_{k=1}^K \mathbb{E}_{\nu^\pi_{K-k}}\left[\frac{1}{\sqrt{\mu(s,\pi(s))}}\right] = \sum_{k=1}^K\sum_s \nu^\pi_{K-k}(s,\pi(s))\frac{1}{\sqrt{\mu(s,\pi(s))}} \\ = \sum_s\Big[\sum_{k=1}^K \nu^\pi_{K-k}(s,\pi(s))\Big]\frac{1}{\sqrt{\mu(s,\pi(s))}}$$

By definition, $\sum_{k=0}^\infty \nu_k^\pi = \frac{\mu_\pi}{1-\gamma}$. Therefore, we have

$$\sum_{k=1}^K \mathbb{E}_{\nu^\pi_{K-k}}\left[\frac{1}{\sqrt{\mu(s,\pi(s))}}\right] \leq \sum_s \frac{\mu_\pi(s,\pi(s))}{1-\gamma}\frac{1}{\sqrt{\mu(s,\pi(s))}}.$$

Using the concealability assumption (Assumption 4.1), $\frac{\mu_\pi(s,\pi(s))}{\mu(s,\pi(s))} \leq C^\star$, we have that

$$\sum_s \frac{\mu_\pi(s,\pi(s))}{1-\gamma}\frac{1}{\sqrt{\mu(s,\pi(s))}} \leq \frac{\sqrt{C^\star}}{1-\gamma}\sum_s\sqrt{\mu_\pi(s,\pi(s))}$$

Using Cauchy-Schwarz inequality, we have

$$\frac{\sqrt{C^\star}}{1-\gamma}\sum_s \sqrt{\mu_\pi(s,\pi(s))} \le \frac{\sqrt{C^\star S}}{1-\gamma}\sqrt{\sum_s \mu_\pi(s,\pi(s))} = \frac{\sqrt{SC^\star}}{1-\gamma}.$$

Therefore,

$$2\sum_{k=1}^{K}\mathbb{E}_{\nu^\pi_{K-k}}\left[16\alpha\log(1/\delta)\sqrt{\frac{1}{T\mu(s,\pi(s))}}\right] \le \frac{32\alpha\log(1/\delta)}{1-\gamma}\sqrt{\frac{SC^\star}{T}}.$$

□

F. Proof of Theorem 4.4

The proof of Theorem 4.4 is similar to the proof of Theorem 1 in (Yan et al., 2023). The main difference is we use the DSD as the bonus to replace the Azuma-Hoeffding-type offset.

Now we introduce some notations used in our proof: For all $(s,a)\in\mathcal{S}\times\mathcal{A}$, we define the transition kernel of (s,a) as

For all $t\in[T]$, $(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}$, we define

$$G_t(s'\mid s,a) = \begin{cases} 1 \text{ if } (s,a,s') = (s_{t-1},a_{t-1},s_t) \\ 0 \text{ if otherwise.}\end{cases}$$

For any deterministic policy π, we introduce $G_\pi:\mathcal{S}\to\Delta(\mathcal{S})$ and $G^\pi:\mathcal{S}\times\mathcal{A}\to\Delta(\mathcal{S}\times\mathcal{A})$ as

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

We define the marginal distribution over optimal actions as:

$$\rho^{\pi^\star}(s,a) = \begin{cases}\rho(s) \text{ if } a=\pi^\star(s), \\ 0 \text{ if otherwise.}\end{cases} \tag{F.1}$$

Then we introduce two technical lemmas that are used for the analysis of Theorem 4.4.

Lemma F.1 (Lemma 8, (Li et al., 2021b)). *Consider the Markov chain $\{s_0,s_1,\cdots\}$ and a stationary distribution μ. For any $0<\delta<1$, if $t\ge\frac{443t_{\text{mix}}}{\mu_{\min}}\log\frac{4S}{\delta}$, then*

$$\forall s\in\mathcal{S}: \quad \mathbb{P}\left(\exists s\in\mathcal{S}:\left|\sum_{i=1}^{t}\mathbf{1}\{s_i=s\}-t\mu(s)\right|\ge\frac{1}{2}t\mu(s)\mid s_1=s\right)\le\delta,$$

where $\mu_{\min}$ is defined as $\mu_{\min}=\min_{s\in\mathcal{S},\,a\in\mathcal{A}}\mu_b(s,a)$.

Lemma F.2 ((Yan et al., 2023) Lemma 4). *For any vector with $V\in\mathbb{R}^d_+$, we have*

$$\sum_{j=0}^{\infty}\left[\gamma\left(1+\frac{1}{H}\right)^3\right]^j\left\langle\rho\left(G_{\pi^\star}\right)^j,V\right\rangle \lesssim \frac{1}{1-\gamma}\left\langle\mu_{\pi^\star},V\right\rangle+\frac{\delta}{ST^4(1-\gamma)}\|V\|_\infty.$$

We proceed by defining the following events:

$$\mathcal{I} := \left\{(s,\pi^\star(s))\mid s\in\mathcal{S},\mu_{\text{b}}\left(s,\pi^\star(s)\right)\ge\frac{\delta}{ST}\right\},$$
$$\mathcal{I}^c := \left\{(s,\pi^\star(s))\mid s\in\mathcal{S},\mu_{\text{b}}\left(s,\pi^\star(s)\right)<\frac{\delta}{ST}\right\}.$$

Then we define some quantities related to the learning rates η that are used in our proof:

$$\eta_j = (H+1)/(H+j)\ , \quad \eta_0^t = \prod_{j=1}^{t}(1-\eta_j)\ , \quad \eta_i^t = \begin{cases} \eta_i \prod_{j=i+1}^{t}(1-\eta_j)\ , \text{ if } t > i \\ \eta_i\ , \text{ if } t = i \\ 0\ , \text{ if } t < i. \end{cases} \tag{F.2}$$

Next, we state a key lemma regarding the step-size which is established in prior work ((Jin et al., 2018) Lemma 4.1), ((Yan et al., 2023), Lemma 1) and ((Li et al., 2021a), Lemma 1).

Lemma F.3. *[(Yan et al., 2023) Lemma 1, (Jin et al., 2018) Lemma 4.1 and (Li et al., 2021a), Lemma 1] The learning rates in equation F.2 satisfy the following properties for any $t \geq 1$ and scalar $1/2 \leq a \leq 1$:*

(i) $\sum_{i=1}^{t} \eta_i^t = 1$ *and* $\eta_0^t = 0$

(ii) $\frac{1}{t^a} \leq \sum_{i=1}^{t} \frac{1}{i^a}\eta_i^t \leq \frac{2}{t^a}$

(iii) $\max_{i\in[t]} \eta_i^t \leq \frac{2H}{t}$ *and* $\sum_{i=1}^{t}(\eta_i^t)^2 \leq \frac{2H}{t}$.

Next, we underscore that for each iteration index $t \leq T$, n_t indicates the number of times (s,a) has been visited prior to iteration t. An instantiation of this quantity that appears frequently in the analysis is $n_t(s,\pi^\star(s))$, the number of times a state-action pair is visited by the optimal policy $\pi^\star$. We further define the deterministic policy $\pi_t : \mathcal{S} \to \mathcal{A}$ as:

$$\pi_t(s) := \begin{cases} \arg\max_{a\in\mathcal{A}} Q_t(s_{t-1},a)\ , \text{ if } s = s_{t-1} \text{ and } V_t(s) > V_{t-1}(s), \\ \pi_{t-1}(s)\ , \text{ otherwise}\ . \end{cases} \tag{F.3}$$

If there are multiple maxima of $Q_t(s_{t-1},a)$, then we set $\pi_t(s)$ as an element of the set of maximizers, i.e., $\pi_t(s) \in \arg\max_{a\in\mathcal{A}} Q_t(s_{t-1},a)$. Next we state a key lemma regarding the role pessimism plays in limiting the probability the value function associated with offline data is far from the true optimal value function.

Then we introduce some quantities that we used in (Yan et al., 2023) and our proof.

$$\begin{aligned}
\alpha_j &:= \left[\gamma\left(1+\frac{1}{H}\right)^3\right]^j \sum_{t=1}^{T}\left\langle \rho\left(G_{\pi^\star}\right)^j, V^\star - V_t\right\rangle, \\
\theta_j &:= \left[\gamma\left(1+\frac{1}{H}\right)^3\right]^j \sum_{t=1}^{T}\sum_{s\in\mathcal{S}}\left[\rho\left(G_{\pi^\star}\right)^j\right](s,\pi^\star(s)) \min\left\{b^q_{n_t(s,\pi^\star(s))}(s,\pi^\star(s)), \frac{1}{1-\gamma}\right\}, \\
\xi_j &:= \left[\gamma\left(1+\frac{1}{H}\right)^3\right]^j \sum_{t=1}^{n_{\text{mix}}(\delta)}\left\langle \rho\left(G_{\pi^\star}\right)^j, V^\star - V_t\right\rangle + \left[\gamma\left(1+\frac{1}{H}\right)^3\right]^{j+1}\left\langle \rho\left(G_{\pi^\star}\right)^{j+1}, V^\star - V_0\right\rangle, \\
\psi_j &:= \left[\gamma\left(1+\frac{1}{H}\right)^3\right]^j \sum_{t=n_{\text{mix}}(\delta)}^{T}\left[\sum_{s\in\mathcal{S},a\in\mathcal{A}}\left[\rho^{\pi^\star}\left(G^{\pi^\star}\right)^j\right](s,a)\sum_{i=1}^{n_t(s,a)}\eta_i^{n_t(s,a)}P_{s,a}\left(V^\star - V_{k_i(s,a)}\right)\right. \\
&\quad \left. - \left(1+\frac{1}{H}\right)\frac{\left[\rho^{\pi^\star}\left(G^{\pi^\star}\right)^j\right](s_t,a_t)}{\mu_{\text{b}}(s_t,a_t)}\sum_{i=1}^{n_t(s_t,a_t)}\eta_i^{n_t(s_t,a_t)}P_{s_t,a_t}\left(V^\star - V_{k_i(s_t,a_t))}\right)\right], \\
\phi_j &:= \gamma^{j+1}\left(1+\frac{1}{H}\right)^{3j+2}\sum_{t=0}^{T}\mathbf{1}_{(s_t,a_t)\in\mathcal{I}}\left[\frac{\left[\rho^{\pi^\star}\left(G^{\pi^\star}\right)^j\right](s_t,a_t)}{\mu_{\text{b}}(s_t,a_t)}P_{s_t,a_t}\left(V^\star - V_t\right)\right. \\
&\quad \left. - \left(1+\frac{1}{H}\right)\sum_{s\in\mathcal{S},a\in\mathcal{A}}\left[\rho^{\pi^\star}\left(G^{\pi^\star}\right)^j\right](s,a)P_{s,a}\left(V^\star - V_t\right)\right].
\end{aligned}$$

Now we present an error decomposition lemma used in continuing our proof.

Lemma F.4 (Regret Decomposition, (Yan et al., 2023)). *The error can be bounded as*

$$J(\pi^\star) - J(\widehat{\pi}) \leq \frac{1}{T}\alpha_0 \leq \frac{1}{T}(\lim_{j\to+\infty}\alpha_j + \sum_{j=0}^{\infty}\xi_j + \sum_{j=0}^{\infty}\theta_j + \sum_{j=0}^{\infty}\psi_j + \sum_{j=0}^{\infty}\phi_j).$$

Then we can employ the same proof in (Yan et al., 2023) to have

$$\begin{aligned}
&\lim_{j\to+\infty}\alpha_j \leq 0,\\
&\sum_{j=0}^{\infty}\xi_j \lesssim \frac{t_{\text{mix}}}{(1-\gamma)^2}\log\frac{1}{\delta} + \frac{t_{\text{mix}}}{T^4(1-\gamma)^2}\log\frac{1}{\delta}\\
&\sum_{j=0}^{\infty}\psi_j \lesssim \frac{C^\star t_{\text{mix}}\,\iota}{(1-\gamma)^3}\log^2\left(\frac{T}{\delta}\right) + \frac{C^\star S t_{\text{mix}}}{(1-\gamma)^2}\log\left(\frac{T}{\delta}\right),\\
&\sum_{j=0}^{\infty}\phi_j \lesssim \frac{C^\star t_{\text{mix}}\iota}{(1-\gamma)^3}\log^2\left(\frac{T}{\delta}\right) + \frac{C^\star S t_{\text{mix}}}{(1-\gamma)^2}\log\left(\frac{T}{\delta}\right).
\end{aligned}$$

We note that the reason we can utilize the same analysis as in (Yan et al., 2023) to bound $\alpha,\ \xi, \psi$ and ϕ is that the only property utilized in determining these bounds is that V_t is constrained by $1/(1-\gamma)$.

Now we bound $\sum_{j=0}^{\infty}\theta_j$. We have

$$\begin{aligned}
\sum_{j=0}^{\infty}\theta_j &= \sum_{j=0}^{\infty}\left[\gamma\left(1+\frac{1}{H}\right)^3\right]^j \sum_{t=1}^{T}\sum_{s\in\mathcal{S}}\left[\rho\left(G_{\pi^\star}\right)^j\right](s)\min\left\{b^q_{n_t(s,\pi^\star(s))}, \frac{1}{1-\gamma}\right\}\\
&\lesssim \frac{1}{1-\gamma}\sum_{t=1}^{T}\sum_{s\in\mathcal{S}}\mu_{\pi^\star}(s)\min\left\{b^q_{n_t(s,\pi^\star(s))}, \frac{1}{1-\gamma}\right\} + \frac{1}{ST^3(1-\gamma)^2}\\
&\lesssim \sum_{s\in\mathcal{S}}\sum_{t=1}^{\bar{t}(s)}\frac{\mu_{\pi^\star}(s)}{(1-\gamma)^2} + \sum_{s\in\mathcal{S}}\sum_{t=\bar{t}(s)+1}^{T}\mu_{\pi^\star}(s)\sqrt{\frac{H\iota}{n_t\left(s,\pi^\star(s)\right)(1-\gamma)^4}} + \frac{1}{T^3(1-\gamma)^2},
\end{aligned} \tag{F.4}$$

where the first inequality is by Lemma F.2 and in the second inequality $\bar{t}$ is defined as $\mathcal{O}(t_{\text{mix}}/\mu_{\text{b}}\left(s,\pi^\star(s)\right)\log\left(ST/\delta\right))$ together with Lemma D.3.

By Lemma F.1 we know equation F.4 can be bounded as

$$\begin{aligned}
&\sum_{s\in\mathcal{S}}\frac{\mu_{\pi^\star}(s)}{\mu_{\text{b}}\left(s,\pi^\star(s)\right)}\frac{t_{\text{mix}}\,\iota}{(1-\gamma)^2} + \sum_{s\in\mathcal{S}}\sum_{t=\bar{t}(s)+1}^{T}\mu_{\pi^\star}(s)\sqrt{\frac{H\iota}{t\mu_{\text{b}}\left(s,\pi^\star(s)\right)(1-\gamma)^4}} + \frac{1}{T^3(1-\gamma)^2}\\
&\lesssim \frac{C^\star S t_{\text{mix}}\iota}{(1-\gamma)^2} + \sum_{s\in\mathcal{S}}\mu_{\pi^\star}\left(s,\pi^\star(s)\right)\sqrt{\frac{HT\iota}{\mu_{\text{b}}\left(s,\pi^\star(s)\right)(1-\gamma)^4}} + \frac{1}{T^3(1-\gamma)^2}\\
&\lesssim \frac{C^\star S t_{\text{mix}}\,\iota}{(1-\gamma)^2} + \sqrt{\frac{C^\star HT\iota}{(1-\gamma)^4}}\sum_{s\in\mathcal{S}}\sqrt{\mu_{\pi^\star}\left(s,\pi^\star(s)\right)}\\
&\lesssim \frac{C^\star S t_{\text{mix}}}{(1-\gamma)^2} + \sqrt{\frac{C^\star ST\iota^2}{(1-\gamma)^5}},
\end{aligned}$$

where the first inequality follows from the fact that $\sum_{t=1}^{T}1/\sqrt{t} \leq 2\sqrt{T}$, the second inequality uses Assumption 4.1 and the last inequality is by the Cauchy-Schwarz inequality.

Combine the bound for $\alpha,\ \theta,\ \psi$ and ϕ, we have

$$J(\pi^\star) - J(\widehat{\pi}) \lesssim \sqrt{\frac{C^\star S\iota^2}{T(1-\gamma)^5}} + \frac{C^\star S t_{\text{mix}}\iota}{T(1-\gamma)^2} + \frac{C^\star t_{\text{mix}}\iota^2}{T(1-\gamma)^3},$$

which concludes the proof of Theorem 4.4.

Now we present the important lemma to show the monotonicity and pessimism of the estimated q-function.

Lemma F.5. *For all $s \in \mathcal{S}$, we have $V_t(s) \leq V^{\pi_t}(s) \leq V^*(s)$ for all $s \in \mathcal{S}$ and*

$$Q^\star\left(s, \pi^\star(s)\right) - Q_t\left(s, \pi^\star(s)\right) \leq \gamma \sum_{i=1}^{n} \eta_i^n P_{s,\pi^\star(s)}\left(V^\star - V_{k_i}\right) + b_n^q\left(s, \pi^\star(s)\right),$$

where n here denote $n_t(s, \pi^\star(s))$.

Proof of F.5. For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, we denote $k = n_t(s, a)$. We set $k_0 = 1$, and for each $1 \leq i \leq N$ we define

$$k_i := \min\left\{\left\{0 \leq k < T : k > k_{i-1}, (s_k, a_k) = (s, a)\right\}, T\right\}.$$

Then we turn to the update rule of the Q-function, we have

$$\begin{aligned}
\left(Q^\star - Q_t\right)(s, a) &= r(s, a) + \gamma P_{s,a} V^\star - \sum_{i=1}^{n} \eta_i^n \left\{r(s, a) + \gamma V_{k_i}\left(s_{k_i+1}\right) - b_i^q(s, a)\right\} \\
&= \sum_{i=1}^{n} \eta_i^n \gamma P_{s,a}\left(V^\star - V_{k_i}\right) + \sum_{i=1}^{n} \eta_i^n \gamma\left(\left(P - G_{k_i}\right) V_{k_i}\right)(s, a) + \sum_{i=1}^{n} \eta_i^n b_i^q(s, a), \qquad \text{(F.5)}
\end{aligned}$$

where the second equation is by Lemma F.3 and we use $P(s, a)$ to denote $P_{s,a}$.

Now we focus on the case when $a = \pi^*(s)$. By using Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\left|\sum_{i=1}^{n} \eta_i^n \gamma\left(\left(P - G_{k_i}\right) V_{k_i}\right)(s, a)\right| &\leq \frac{1}{1-\gamma} \sqrt{\left(\sum_{i=1}^{n} (\eta_i^n)^2\right) \times \left(\sum_{i=1}^{n} \left(\left(P - G_{k_i}\right) V_{k_i}\right)^2 (s, \pi^*(s))\right)} \\
&\leq \frac{1}{1-\gamma} \sqrt{\frac{H}{n}} \sqrt{\sum_{i=1}^{n} \left(\left(P - G_{k_i}\right) V_{k_i}\right)^2 (s, \pi^*(s))} \\
&\leq \frac{1}{1-\gamma} \sqrt{\frac{H \log n}{n}} b_n^q(s, \pi^*(s)), \qquad \text{(F.6)}
\end{aligned}$$

where the second inequality is by Lemma F.3 and the last inequality is by Theorem 3.1 combined with the definition of the bonus in the Algorithm 3 ($\sqrt{i} b_i^q \leq \sqrt{n} b_n^q$).

Combining Lemma F.3, equation F.5 and equation F.6, we have

$$\left(Q^\star - Q_t\right)\left(s, \pi^\star(s)\right) \leq \gamma \sum_{i=1}^{n} \eta_i^n P_{s,\pi^\star(s)}\left(V^\star - V_{k_i}\right) + b_n^q\left(s, \pi^\star(s)\right)$$

for all $s \in \mathcal{S}$ and $t \in [T]$.

Now we turn to prove that $V^{\pi_t} \geq V_t$, first we prove

$$\left(Q^{\pi_t} - Q_j\right)\left(s, \pi_t(s)\right) \geq \gamma P_{s,\pi_t(s)}\left(V^{\pi_t} - V_j\right) \mathbf{1}\left\{n_t\left(s, \pi_t(s)\right) \geq 1\right\} \qquad \text{(F.7)}$$

holds for all $s \in \mathcal{S}$, and $t \in [T]$.

Similar to the decomposition in equation F.5, we have

$$
\begin{aligned}
&\left(Q^{\pi_t} - Q_j\right)(s, \pi_t(s)) \\
&= \left(r + \gamma P V^{\pi_t}\right)(s, \pi_t(s)) - \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} \left\{r\left(s, \pi_t(s)\right) + \gamma V_{k_i}\left(s_{k_i+1}\right) - b_i^q\left(s, \pi_t(s)\right)\right\} \\
&= \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} \gamma \left\{P_{s,\pi_t(s)}\left(V^{\pi_t} - V_{k_i}\right) + \left(\left(P - G_{k_i}\right) V_{k_i}\right)(s, \pi_t(s))\right\} + \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} b_i^q\left(s, \pi_t(s)\right) \\
&\geq \left(\sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))}\right) \gamma \min_{1 \leq i \leq n} P_{s,\pi_t(s)}\left(V^{\pi_t} - V_{k_i}\right) + \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} \gamma \left(\left(P - G_{k_i}\right) V_{k_i}\right)(s, \pi_t(s)) \\
&\quad + \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} b_i^q\left(s, \pi_t(s)\right) \\
&\geq \gamma P_{s,\pi_t(s)}\left(V^{\pi_t} - V_t\right) + \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} \gamma \left(\left(P - G_{k_i}\right) V_{k_i}\right)(s, \pi_t(s)) + \sum_{i=1}^{n_j(s,\pi_t(s))} \eta_i^{n_j(s,\pi_t(s))} b_i^q\left(s, \pi_t(s)\right),
\end{aligned}
\tag{F.8}
$$

where the last inequality is by the fact that V_t is non-decreasing in t.

By Theorem 3.1 and the definition of the bonus in the Algorithm 3, we have

$$
\gamma \left(\left(P - G_{k_i}\right) V_{k_i}\right)(s, \pi_t(s)) \lesssim b_i^q(s, \pi_t(s)), \tag{F.9}
$$

holds for all $s \in \mathcal{S}$ and $i \in [T]$, then we could combine equation F.9 and equation F.8 to conclude equation F.7.

Now we continue with equation F.7. By the update rule of Algorithm 3, there exists $j(t) \leq t$ such that $V_t(s) = V_{j(t)}(s) = Q_{j(t)}\left(s, \pi_{j(t)}(s)\right)$ and $\pi_t(s) = \pi_{j(t)}(s)$. Hence we have

$$
\begin{aligned}
\left(V^{\pi_t} - V_t\right)(s) &= Q^{\pi_t}\left(s, \pi_t(s)\right) - Q_{j(t)}\left(s, \pi_{j(t)}(s)\right) \\
&\geq \min\left\{\gamma P_{s,\pi_t(s)}\left(V^{\pi_t} - V_{j(t)}\right), 0\right\} \\
&\geq \min\left\{\gamma P_{s,\pi_{j(t)}(s)}\left(V^{\pi_t} - V_t\right), 0\right\},
\end{aligned}
$$

where the second inequality is by equation F.7 and the last inequality is by the monotonicity of V^t. Now we set $s_{\min} = \arg\min_{s \in \mathcal{S}}\left(V^{\pi_t} - V_t\right)(s)$, then for all $s \in \mathcal{S}$ we have

$$
\begin{aligned}
\left(V^{\pi_t} - V_t\right)(s) \geq \left(V^{\pi_t} - V_t\right)(s_{\min}) &\geq \min\left\{\gamma P_{s_{\min},\pi_{j(t)}}(s_{\min})\left(V^{\pi_t} - V_t\right), 0\right\} \\
&\geq \min\left\{\gamma\left(V^{\pi_t} - V_t\right)(s_{\min}), 0\right\},
\end{aligned}
$$

where the second inequality uses that $s_{\min} = \arg\min_{s \in \mathcal{S}}\left(V^{\pi_t} - V_t\right)(s)$. Then we have $\left(V^{\pi_t} - V_t\right)(s_{\min}) \geq 0$. Hence, we conclude the proof. □

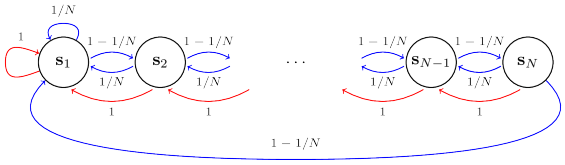

Figure 7: The DeepSea MDP

G. Additional Experiments and Details

In this section, we expand upon the details required for the experiments presented in Sec. 5 as well as provide an expanded set of experiments for the purpose of rounding out the validation. All datasets included one sampled optimal trajectory to satisfy our assumption of Single-Policy Concentrability (Assumption 4.1). We further clipped the minimum transition probability in all environments except Frozen Lake to a small number (10^{-6} for Prior, DeepSea and 10^{-2} for Random, Portfolio) to ensure calculated DSD values were more stable. All experiments were run on an AWS c5.2xlarge instance except for the Random MDP experiments which used a cluster with an Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz and 252 GB of RAM. First, we discuss the different environments.

Portfolio Optimization. We consider portfolio optimization (Neuneier, 1997; Moody & Saffell, 2001), where the goal is to select a series of asset allocations, given a series of prices and instantaneous returns. For details refer Appendix G.2.1. The ordering of states is by the asset index $(1, .., k)$ from lowest to highest asset value. For example, if we consider the states of 3 assets priced between $[50, 55]$ with five discretized values $s_a = (50, 51, 50)$, $s_b = (50, 51, 52)$ and $s_c = (51, 50, 50)$, then $s_a < s_b < s_c$ in our ordering. In this example, a negative cyclic permutation would be as follows $\vee s_a = (50, 50, 55)$, $\vee s_b = (50, 51, 51)$ and $\vee s_c = (50, 55, 55)$. This ordering permits calculating DSD.

Experimentally, we consider 3 assets, and each is constrained to lie in the interval $[50, 100]$ which is discretized into five equally-spaced distinct values yielding a total of $5^3 = 125$ states. The action space is discretized into five possible values along each asset allocation in $[0, 1]$ with valid actions being vectors whose values sum to one yielding a total of 15 distinct actions. The transition probabilities were chosen from a uniform distribution on $[0, 1]$ and normalized. Returns are calculated over 200 steps.

Frozen Lake. Frozen Lake is a discrete state-action environment with 16 states and 4 actions (up/down/left/right). An agent starts on the top left of a 4x4 grid and must reach the bottom right while avoiding frozen states. An additional randomness is present due to a "slip" factor that causes perpendicular movement to a chosen direction with uniform probability. We order the states based on the row and column, with default indexing provided by Gym (Brockman et al., 2016). Episodes were terminated based on reaching the goal or a hazard.

DeepSea MDP. The DeepSea MDP (Osband & Van Roy, 2017a) is a sparse reward environment that follows a chain-like structure as shown in Fig. 7. The agent starts from the left (s_1) and is allowed to swim either left or right. All states have a zero reward except the final transition from state s_{N-1} to s_N. Upon taking the 'swim right' action, the agent is allowed to transition to the adjacent states given a certain probability as shown. The agent is allowed to move left deterministically upon taking the 'swim left' action. Thus without a sampled optimal trajectory, it is difficult for the agent to explore and reach the final state. States were ordered from left to right in the chain. To make the offline data sufficiently challenging, episode lengths were 4 steps longer than the total number of states, i.e. 68 steps long for the 64 state MDP and 132 steps long for the 128 state MDP.

Prior MDP. Unlike the structured environments discussed, we also consider a more general MDP from (Markou & Rasmussen, 2019) inspired by (Osband & Van Roy, 2017b). Here the state transitions from s to s' given action a are sampled from a Categorical variable where the probabilities are from a Dirichlet prior with $\kappa = 1$. The reward values for a given (s, a, s') is set by sampling a Normal distribution whose mean and precision are set from a Normal-Gamma prior with parameters $(\mu_0, \lambda, \alpha, \beta) = (0.00, 1.00, 4.00, 4.00)$. The final rewards are clipped to lie between $[-1, 1]$ since our penalties assume $|r| < 1$. DSD values were calculated using the given state indexing. Each episode was sampled for 100 time steps.

Random MDP. We experimented on another sparser Random MDP variant (pym, 2015) with 64 states and 64 actions. Here the transition probabilities are sampled from a uniform distribution between $[0, 1]$. Next a mask is calculated by sampling the same uniform distribution $|S|^2|A|$ times, fixing a threshold sampled from the uniform distribution, and setting the mask values below the threshold to zero. Finally the transition probabilities are normalized for each (s, a) pair. The rewards for each transition are sampled uniformly between $[-1, 1]$. We order the states using the given state indexing to calculate the

DSD values. Episodes were sampled with a 100 step horizon.

G.1. Hyperparameter Selection and Implementation Details

For hyperparameter selection we permit all algorithms one final 'learned policy' evaluation on the environment. Much prior work (Agarwal et al., 2020) has worked with this same assumption and we provide this advantage to all tested algorithms fairly. An alternative would be to choose an Offline Policy Evaluation (OPE) approach (Zhang & Jiang, 2021) to select hyperparameters.

In this section, we detail the hyperparameter selection of the pessimistic penalty coefficient α in equation 3.8-equation 3.9, the coefficients used in the penalty in (Rashidinejad et al., 2021) which are V_{max}, an upper-bound on the value function, and L, which is a sample size and state-cardinality, dependent constant. In the paper, $L = L_c[\log(2(T+1)SA/\delta]$ where $L_c = 2000$ but we find experimentally to hand-tune this quantity according to a grid-search to be more effective. In Table 4 we present the range of values used, as well as the actually selected value for each experimental instance over all seeds. Further note that value iteration is a learning rate-free method, so there is no valid concept of learning rate η here.

Similarly, Table 5 requires specifying a learning rate, a penalty coefficient α for DSD, or otherwise a multiplicative constant C_b that determines the scale of the penalty in (Yan et al., 2023). In this reference, minimal guidance is given on how to select C_b for suitable performance in practice, so we performed a grid-search. We report the used value for each environment.

Task	Hyperparameter	Definition	Value Range	Selection
Random	α	equation 3.8	$[0.01, 0.1, 1, 10]$	0.1
	L_c (Rashidinejad et al., 2021)	LCB Coef.	$[10^{-3}, ..., 10^{3}]$	0.03
	V_{max} (Rashidinejad et al., 2021)	Value Upper-Bound	$[1, 50, 100]$	1
Portfolio	α	equation 3.8	$[0.01, 0.1, 1, 10]$	0.1
	L_c (Rashidinejad et al., 2021)	LCB Coef.	$[10^{-3}, ..., 10^{3}]$	1000
	V_{max} (Rashidinejad et al., 2021)	Value Upper-Bound	$[1, 50, 100]$	50
Prior	α	equation 3.8	$[0.01, 0.1, 1, 10]$	0.1
	L_c (Rashidinejad et al., 2021)	LCB Coef.	$[10^{-3}, ..., 10^{3}]$	0.3
	V_{max} (Rashidinejad et al., 2021)	Value Upper-Bound	$[1, 50, 100]$	2
DeepSea	α	equation 3.8	$[0.01, 0.1, 1, 10]$	0.1
	L_c (Rashidinejad et al., 2021)	LCB Coef.	$[10^{-3}, ..., 10^{3}]$	10^{-3}
	V_{max} (Rashidinejad et al., 2021)	Value Upper-Bound	$[1, 50, 100]$	1
Frozen Lake	α	equation 3.8	$[0.01, 0.1, 1, 10]$	0.01
	L_c (Rashidinejad et al., 2021)	LCB Coef.	$[10^{-3}, ..., 10^{3}]$	316
	V_{max} (Rashidinejad et al., 2021)	Value Upper-Bound	$[1, 50, 100]$	1

Table 4: Hyperparameters for the Value Iteration experiments. A grid search was carried over these parameter ranges for each seed of a random number generator, which determines the initialization. Then, the reported value (by a majority selection over the seeds) is under the "Selection" column. Notes: 13 log. spaced values, $L_c = 2000$ in (Rashidinejad et al., 2021) in each row.

G.2. Additional Experiments

In this section, we provide more details of the tasks specified in Sec. 5. Recall that these contain a portfolio optimization task, Frozen Lake within OpenAI gym, and Random MDP with 64 States and Actions for a general setting.

G.2.1. Portfolio Optimization

To bring out the value addition using this new *information-directed* approach captured by IPM, we consider a simple financial setting of offline portfolio optimization where the historical datasets lack information in all regions of the asset-action space. When one wants to learn a safe policy from historical data that performs well under previously unobserved asset values, the new pessimism introduces an information-theoretic regularization using information available only in the dataset. While this is best one could hope to achieve with no active data collection, the Stein information-directed pessimism comes closer to the true average return values than just using probabilistic bounds based ones as in (Rashidinejad

Task	Hyperparameter	Definition	Value Range	Selection
Random	η	Learning Rate	$[10^{-1}, 10^{-0.5}, 0.5, 1]$	0.1
	α	equation 3.9	$[0.01, 0.1, 1, 10]$	0.1
	C_b(Yan et al., 2023)	LCB Coef.	$[10^{-2}, ..., 10^{4}]$	100.0
Portfolio	η	Learning Rate	$[10^{-1}, 10^{-0.5}, 0.5, 1]$	0.1
	α	equation 3.9	$[0.01, 0.1, 1, 10]$	0.1
	C_b(Yan et al., 2023)	LCB Coef.	$[10^{-2}, ..., 10^{4}]$	0.01
Prior	η	Learning Rate	$[10^{-1}, 10^{-0.5}, 0.5, 1]$	0.1
	α	equation 3.9	$[0.01, 0.1, 1, 10]$	0.1
	C_b(Yan et al., 2023)	LCB Coef.	$[10^{-2}, ..., 10^{4}]$	0.01
DeepSea	η	Learning Rate	$[10^{-1}, 10^{-0.5}, 0.5, 1]$	0.5
	α	equation 3.9	$[0.01, 0.1, 1, 10]$	0.1
	C_b(Yan et al., 2023)	LCB Coef.	$[10^{-2}, ..., 10^{4}]$	10.0
Frozen Lake	η	Learning Rate	$[10^{-2}, 10^{-1}, 10^{-0.5}, 0.5, 1]$	0.01
	α	equation 3.9	$[0.01, 0.1, 1, 10]$	0.01
	C_b (Yan et al., 2023)	LCB Coef.	$[10^{-2}, ..., 10^{4}]$	0.01

Table 5: Hyperparameters for the Q-learning experiments. A grid search was carried over these parameter ranges for each seed of a random number generator, which determines the initialization. Then, the reported value (by a majority selection over the seeds) is under the "Selection" column.

et al., 2021; Uehara et al., 2023).

Here we consider the formulation of the portfolio optimization problem in the framework of MDP. Portfolio optimization is one of the most fundamental and important applications in finance, with the goal of finding an allocation of the investments in line with the preferences of clients. We consider a simple version that focuses on just the first moment, unlike the traditional setup that uses higher moments like variance, skewness etc, to illustrate the performance of the proposed algorithm. Let $\nu_1, \nu_2, \cdots, \nu_k$ denote the k available assets in a portfolio. Assume that each asset ν^i can take positive discrete values inside some set $S_i \subset \mathbb{R}^+$. Let $\mathcal{S} = S_1 \times S_2 \times \cdots S_k$ denotes the set of all possible combinations of values that the assets can take. Let a finite set $\mathcal{A} \subset \{a \in \mathbb{R}^k | \sum_{i=1}^k a_i = 1, a_i \geq 0\}$, which is the $k-$dimensional probability simplex. Here each $a = (a_1, a_2, \cdots, a_k) \in \mathcal{A}$ is an allocation of the principal to each of the assets $\nu_1, \nu_2, \cdots, \nu_k$: where the proportion a_i is allocated to asset ν_i for each $i \in [k]$. Let the transition kernel $\mathbb{P}(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ represent the market dynamics. The expected return of choosing an allocation a in state s is given by

$$(\hat{r}_t :=)\hat{R}(s_t, a_t) = \sum_j a_j(\nu_j^{s_{t+1}} - \nu_j^{s_t})/\nu_j^{s_t}.$$

where ν_j^s denotes the value of asset ν_j in state s. Our ordering of states is by our asset index $(1, .., k)$ from lowest to highest asset value. For example, if we consider the states of 3 assets priced between $[50, 55]$ with five discretized values $s_a = (50, 51, 50)$, $s_b = (50, 51, 52)$ and $s_c = (51, 50, 50)$, then $s_a < s_b < s_c$ in our ordering. Using this defined ordering we calculate the discrete Stein discrepancy.

Table 6 expands upon the results given in Sec. 5 to include all data sampling scenarios, and studies the role of the size of the offline data set (which may be also found in Fig. 8, and 15). We further include results with a smaller asset price range ($[50, 55]$) to showcase the effects of the reward range on the algorithm performance. This causes the absolute reward $|r_i|$ at time step i to be $|r_i| < 5/55$ vs $|r_i| < 50/55$ for the asset price range $[50, 100]$, i.e. effectively reducing reward variance given the same transition matrix. Of note is that prior pessimism-based offline RL algorithms (VI-LCB) require further modifications to work well in practice. A penalty free variant (VI-Vanilla) performs significantly better albeit with *no guarantees* on optimal behaviour, so any good performance is just an artifact of the specific dataset. Our approach with a DSD-based penalty is shown to improve upon this, highlighting the benefits of the approach to offline RL problems in the financial domain.

We note how the reward range affects the gains observed by adding a pessimism term into our update. We posit this gap in performance is due to the higher reward variance causing a larger gap from the true expected return for a given policy.

AR	Data	$\mathbf{N_{ep}}$	T ($\times 10^3$)	VI	VI-LCB (P)	VI-LCB (Td.)	IDP-VI (Ours)	VI Vanilla	CQL	QL	IDP-Q (Ours)	Q LCB	IDP-Q-KP (Ours)
$[50, 100]$	**'Hard'**	31	7.0	62.13	13.05	16.71	48.31	44.65	21.76	52.05	53.02	24.39	51.93
		100	22.0	62.19	12.37	16.42	**54.20**	51.96	32.24	55.41	**57.54**	33.42	55.67
		316	70.0	62.16	14.22	15.69	**59.74**	58.44	35.93	56.13	54.42	34.54	**56.29**
	'Random'	31	6.6	62.11	12.79	17.07	**47.16**	45.10	12.81	50.45	**52.59**	21.85	51.10
		100	20.0	62.16	11.27	15.97	**56.57**	54.81	27.16	50.27	52.64	22.58	**52.83**
		316	64.0	62.11	11.04	16.28	**60.00**	59.84	31.62	51.48	53.20	21.21	**54.09**
$[50, 55]$	**'Hard'**	31	7.0	6.38	0.25	0.83	4.45	4.43	2.53	**5.34**	5.33	1.87	5.19
		100	22.0	6.39	0.36	0.74	**5.13**	5.10	2.20	**5.70**	5.54	3.27	5.52
		316	70.0	6.38	0.76	0.60	**5.89**	5.87	2.06	**6.02**	5.44	3.90	5.41
		1000	220.0	6.39	0.24	0.84	**6.35**	6.31	2.50	**6.04**	5.54	4.64	5.43

Table 6: Mean Return in the Portfolio Optimization environment for varying dataset sizes and asset price ranges over 5 differently seeded training runs. **AR**: Asset Price Range ($[50, 55]$ or $[50, 100]$), **Data**: Dataset used ('**Easy**', '**Hard**', '**Random**'), $\mathbf{N_{ep}}$: Number of episodes used to create the dataset, **T**: Total dataset size i.e. number of (s, a, r, s) tuples, **VI** : Value Iteration (Sutton & Barto, 1998) with oracle access to true MDP, **VI-LCB (P)**: VI-LCB (Rashidinejad et al., 2021) w/ paper reported constants, **VI-LCB (Td.)**: VI-LCB (Rashidinejad et al., 2021) w/ tuned constants, **IDP-VI (Ours)**: Our IDP-VI algorithm, **VI Vanilla**: VI-Vanilla with no pessimism penalty term, **CQL**: Conservative Q-learning after 30 epochs (Kumar et al., 2020), **QL**: Q-learning (Watkins & Dayan, 1992), **IDP-Q (Ours)**: Our IDP-Q algorithm, **Q-LCB**: Q-learning with LCB (Shi et al., 2022), **IDP-Q-KP (Ours)**: Our IDP-Q algorithm with Known Transition Probabilities to calculate DSD Penalty.

G.2.2. FROZEN LAKE

Here we report alternate data sampling and offline data set size ablation studies for Frozen Lake. The results of these comparisons are given in Table 7. While model-free techniques (QL, IDP-Q) exhibit greater performance than the model-based (or value iteration based) variants, we note our pessimistic update (IDP-Q) is useful to get closer to optimal performance. This gap between the two sets of techniques may be due to the sparse transition matrix causing DSD values to be harder to compute in a stable manner. See also Fig. 16, 18, and 17 for visualizations.

Data	$\mathbf{N_{ep}}$	T ($\times 10^3$)	VI	VI-LCB (P)	VI-LCB (Td.)	IDP-VI (Ours)	VI Vanilla	CQL	QL	IDP-Q (Ours)	Q LCB	IDP-Q-KP (Ours)
'Easy'	100	6.0	0.73	0.01	0.08	0.48	0.49	0.35	0.51	0.54	**0.60**	0.46
	316	18.0	0.73	0.05	0.12	0.62	0.64	0.43	0.71	0.65	**0.76**	0.54
	1000	59.0	0.73	0.00	0.15	0.66	0.62	0.66	0.70	0.66	**0.75**	0.64
	3162	190.0	0.74	0.00	0.14	0.64	0.62	0.75	0.74	0.61	0.75	0.63
'Hard'	100	0.9	0.72	0.04	0.10	**0.31**	0.17	0.12	**0.64**	0.52	0.15	0.62
	316	2.7	0.73	0.03	0.16	**0.51**	0.35	0.09	0.50	0.52	0.33	**0.67**
	1000	8.8	0.73	0.01	0.17	**0.52**	0.33	0.09	0.69	0.68	0.52	**0.73**
	3162	28.0	0.73	0.00	0.21	**0.44**	0.28	0.12	0.69	0.73	**0.74**	0.68
'Random'	100	0.9	0.72	0.00	0.09	**0.50**	0.34	0.05	0.31	0.43	0.20	**0.46**
	316	2.5	0.72	0.00	0.11	**0.64**	0.51	0.06	0.32	0.37	0.28	**0.54**
	1000	7.8	0.74	0.04	0.17	**0.70**	0.63	0.08	0.72	**0.73**	0.05	0.71
	3162	24.0	0.73	0.02	0.10	**0.63**	0.62	0.08	0.74	0.70	0.13	**0.76**

Table 7: Mean test performance on Frozen-Lake-v1 for varying dataset sizes over 5 differently seeded training runs. **Data**: Dataset used ('**Easy**', '**Hard**', '**Random**'), $\mathbf{N_{ep}}$: Number of episodes used to create the dataset, **T**: Total dataset size i.e. number of (s, a, r, s) tuples, **VI** : Value Iteration (Sutton & Barto, 1998) with oracle access to true MDP, **VI-LCB (P)**: VI-LCB (Rashidinejad et al., 2021) w/ paper reported constants, **VI-LCB (Td.)**: VI-LCB (Rashidinejad et al., 2021) w/ tuned constants, **IDP-VI (Ours)**: Our IDP-VI algorithm, **VI Vanilla**: VI-Vanilla with no pessimism penalty term, **CQL**: Conservative Q-learning after 30 epochs (Kumar et al., 2020), **QL**: Q-learning (Watkins & Dayan, 1992), **IDP-Q (Ours)**: Our IDP-Q algorithm, **Q-LCB**: Q-learning with LCB (Shi et al., 2022), **IDP-Q-KP (Ours)**: Our IDP-Q algorithm with Known Transition Probabilities to calculate DSD Penalty.

G.2.3. PRIOR MDP

Observing the Prior MDP results in Table. 8 and Fig. 9, 10, and 11 we note the strong performance of Q-LCB on the '**Easy**'and '**Hard**'datasets due to the relatively large amount of optimal transition data provided. In the '**Random**'dataset however we see the strengths of our DSD-based approach able to capture the multi-modal nature of the problem with insufficient optimal data. We also note how the Q-learning variant (IDP-Q) may have some instabilities in calculating the DSD value when dealing with low probability state transitions (clipped at 10^{-6} for the PriorMDP environment) leading to a perfomance impact which can occur with the larger sampled datasets (Fig. 10). This is less likely to occur in the Value Iteration variant of our algorithm due to a larger batch size used during each update allowing a more stable DSD calculation. A carefully constructed sampling function or a smoothing function on the penalty could be used to alleviate this effect.

Data	N_{ep}	T ($\times 10^4$)	VI	VI-LCB (P)	VI-LCB (Td.)	IDP-VI (Ours)	VI Vanilla	CQL	QL	IDP-Q (Ours)	Q LCB	IDP-Q-KP (Ours)
'Easy'	100	3.0	29.54	-1.13	4.19	12.35	2.18	8.50	29.33	29.44	9.93	26.07
	316	9.5	29.36	2.48	4.98	**28.64**	4.23	10.39	29.70	29.32	30.27	28.49
	1000	30.0	29.48	0.30	10.33	**30.31**	8.39	11.91	29.61	25.46	**30.31**	27.20
	3162	95.0	30.02	1.02	24.44	**30.16**	14.01	15.56	29.82	27.77	**30.10**	28.51
'Hard'	100	1.1	29.88	3.42	4.46	6.10	3.44	2.70	16.27	17.23	1.14	15.57
	316	3.5	29.54	0.87	5.67	12.80	4.88	3.96	21.75	21.06	29.26	17.34
	1000	11.0	29.37	-0.82	7.11	16.84	4.52	5.53	24.08	23.88	**30.31**	20.30
	3162	35.0	29.75	1.00	11.62	**21.68**	11.65	6.54	26.13	11.69	**30.47**	24.12
'Random'	100	1.0	29.91	3.16	4.27	7.20	3.70	1.01	9.05	10.96	0.49	9.49
	316	3.2	29.68	-0.03	7.73	10.94	7.36	1.86	11.68	**13.30**	7.71	12.40
	1000	10.0	29.31	-1.37	7.94	**18.79**	8.23	3.59	14.25	15.99	7.30	15.88
	3162	32.0	29.61	-3.27	12.93	**23.14**	12.90	3.41	15.81	12.05	7.41	15.50

Table 8: Mean Return on PriorMDP 64x64 for varying dataset sizes over 5 differently seeded training runs. **Data**: Dataset used ('**Easy**', '**Hard**', '**Random**'), N_{ep}: Number of episodes used to create the dataset, **T**: Total dataset size i.e. number of (s, a, r, s) tuples, **VI** : Value Iteration (Sutton & Barto, 1998) with oracle access to true MDP, **VI-LCB (P)**: VI-LCB (Rashidinejad et al., 2021) w/ paper reported constants, **VI-LCB (Td.)**: VI-LCB (Rashidinejad et al., 2021) w/ tuned constants, **IDP-VI (Ours)**: Our IDP-VI algorithm, **VI Vanilla**: VI-Vanilla with no pessimism penalty term, **CQL**: Conservative Q-learning after 30 epochs (Kumar et al., 2020), **QL**: Q-learning (Watkins & Dayan, 1992), **IDP-Q (Ours)**: Our IDP-Q algorithm, **Q-LCB**: Q-learning with LCB (Shi et al., 2022), **IDP-Q-KP (Ours)**: Our IDP-Q algorithm with Known Transition Probabilities to calculate DSD Penalty.

G.2.4. DEEPSEA

As expected, the unimodal nature of the DeepSea MDP allows for a reasonable performance by existing model-based approaches (VI-LCB). From the Table 9 and Fig. 19 we can see that our method (IDP-VI) is performant in these sparse reward, unimodal settings as well.

G.2.5. RANDOM MDP

In Table 2, Fig. 12 we note how existing pessimism-based approaches require significantly larger data sizes to achieve comparable results to our DSD-based algorithm. Returns are calculated over 100 steps and results are averaged over 5 differently seeded runs. We further point out for the '**Easy**' case (Fig. 14) how the VI Vanilla approach (without pessimism) keeps a consistent gap with the upper bound in performance (indicated by the Value Iteration curve) highlighting the need for pessimism in Offline RL approaches in all data size regimes. For the '**Random**'case, we see how model-free approaches (QL, IDP-Q) are insufficient and model-based approaches (IDP-VI, VI-Vanilla) use the data efficiently to learn a model of the environment that can yield the optimal policy.

G.3. Value function plots

To visualize the value functions of our environments, we include the plots over different states in Fig. 20. We further consider environments with an explicit bimodal distribution from which we sample the transition probabilities. The plots are indicative of the multi-modal nature of our problem.

$\|\mathbf{S}\|$	Data	$\mathbf{N_{ep}}$	T ($\times 10^4$)	VI	VI-LCB (P)	VI-LCB (Td.)	IDP-VI (Ours)	VI Vanilla	CQL	QL	IDP-Q (Ours)	Q LCB	IDP-Q-KP (Ours)
64	'Random'	100	0.69	2.01	0.00	0.07	0.04	0.02	0.62	0.03	0.05	0.00	0.05
		316	2.2	2.02	0.00	0.03	0.05	0.04	0.62	0.01	0.04	0.00	0.04
		1000	6.8	2.03	0.01	0.72	0.73	0.72	0.62	0.01	0.03	0.00	0.02
		3162	22.0	2.03	0.00	1.38	1.42	1.39	0.31	0.00	0.03	0.01	0.02
128	'Random'	100	1.3	2.15	0.00	0.36	0.37	0.31	0.93	0.13	0.10	0.00	0.13
		316	4.2	2.14	0.02	0.20	0.20	0.12	0.92	0.04	0.04	0.00	0.04
		1000	13.0	2.16	0.00	0.98	0.96	0.94	0.92	0.02	0.04	0.00	0.10
		3162	420.0	2.14	0.00	2.27	**2.40**	2.13	0.32	0.03	0.09	0.08	0.09

Table 9: Mean Return in the DeepSea environment for different state space sizes over 5 differently seeded training runs. $|\mathbf{S}|$: Size of State space (64 or 128), **Data**: Dataset used ('**Random**'), $\mathbf{N_{ep}}$: Number of episodes used to create the dataset, **T**: Total dataset size i.e. number of (s, a, r, s) tuples, **VI** : Value Iteration (Sutton & Barto, 1998) with oracle access to true MDP, **VI-LCB (P)**: VI-LCB (Rashidinejad et al., 2021) w/ paper reported constants, **VI-LCB (Td.)**: VI-LCB (Rashidinejad et al., 2021) w/ tuned constants, **IDP-VI (Ours)**: Our IDP-VI algorithm, **VI Vanilla**: VI-Vanilla with no pessimism penalty term, **CQL**: Conservative Q-learning after 30 epochs (Kumar et al., 2020), **QL**: Q-learning (Watkins & Dayan, 1992), **IDP-Q (Ours)**: Our IDP-Q algorithm, **Q-LCB**: Q-learning with LCB (Shi et al., 2022), **IDP-Q-KP (Ours)**: Our IDP-Q algorithm with Known Transition Probabilities to calculate DSD Penalty.

Data	$\mathbf{N_{ep}}$	T ($\times 10^4$)	VI	VI-LCB (P)	VI-LCB (Td.)	IDP-VI (Ours)	VI Vanilla	CQL	QL	IDP-Q (Ours)	Q LCB	IDP-Q-KP (Ours)
'Easy'	100	3.0	38.19	-0.17	7.30	25.25	4.66	10.21	37.36	35.07	9.68	37.13
	316	9.5	38.19	-0.18	16.29	36.77	6.44	14.85	37.40	26.29	35.57	36.91
	1000	30.0	38.11	0.04	35.32	**38.26**	10.06	19.53	37.41	34.74	37.25	37.25
	3162	95.0	38.22	-0.21	38.25	**38.37**	27.27	23.18	37.43	34.75	37.37	36.93
'Hard'	100	1.1	37.31	-0.55	4.18	10.51	3.92	3.01	27.64	29.54	0.24	27.08
	316	3.5	37.35	0.16	5.16	**20.83**	5.31	7.52	**33.60**	30.74	32.44	32.21
	1000	11.0	37.34	-0.85	9.89	**32.61**	8.77	8.75	36.10	27.59	**36.94**	32.92
	3162	35.0	37.18	0.68	19.60	**36.53**	19.63	8.77	36.43	33.27	**37.46**	34.35
'Random'	100	1.0	37.31	0.51	4.93	5.69	4.69	2.06	11.89	15.15	-0.67	**18.27**
	316	3.2	37.34	-2.03	13.08	**15.16**	13.31	4.80	13.09	16.18	2.56	**21.42**
	1000	10.0	37.07	0.33	23.22	**30.44**	23.06	4.61	21.03	23.83	3.67	**25.32**
	3162	32.0	37.03	1.44	31.04	**35.80**	31.83	4.04	23.03	25.54	2.41	**26.65**

Table 10: Mean Return on Random 64x64 for varying dataset sizes over 5 differently seeded training runs. **Data**: Dataset used ('**Easy**', '**Hard**', '**Random**'), $\mathbf{N_{ep}}$: Number of episodes used to create the dataset, **T**: Total dataset size i.e. number of (s, a, r, s) tuples, **VI** : Value Iteration (Sutton & Barto, 1998) with oracle access to true MDP, **VI-LCB (P)**: VI-LCB (Rashidinejad et al., 2021) w/ paper reported constants, **VI-LCB (Td.)**: VI-LCB (Rashidinejad et al., 2021) w/ tuned constants, **IDP-VI (Ours)**: Our IDP-VI algorithm, **VI Vanilla**: VI-Vanilla with no pessimism penalty term, **CQL**: Conservative Q-learning after 30 epochs (Kumar et al., 2020), **QL**: Q-learning (Watkins & Dayan, 1992), **IDP-Q (Ours)**: Our IDP-Q algorithm, **Q-LCB**: Q-learning with LCB (Shi et al., 2022), **IDP-Q-KP (Ours)**: Our IDP-Q algorithm with Known Transition Probabilities to calculate DSD Penalty.

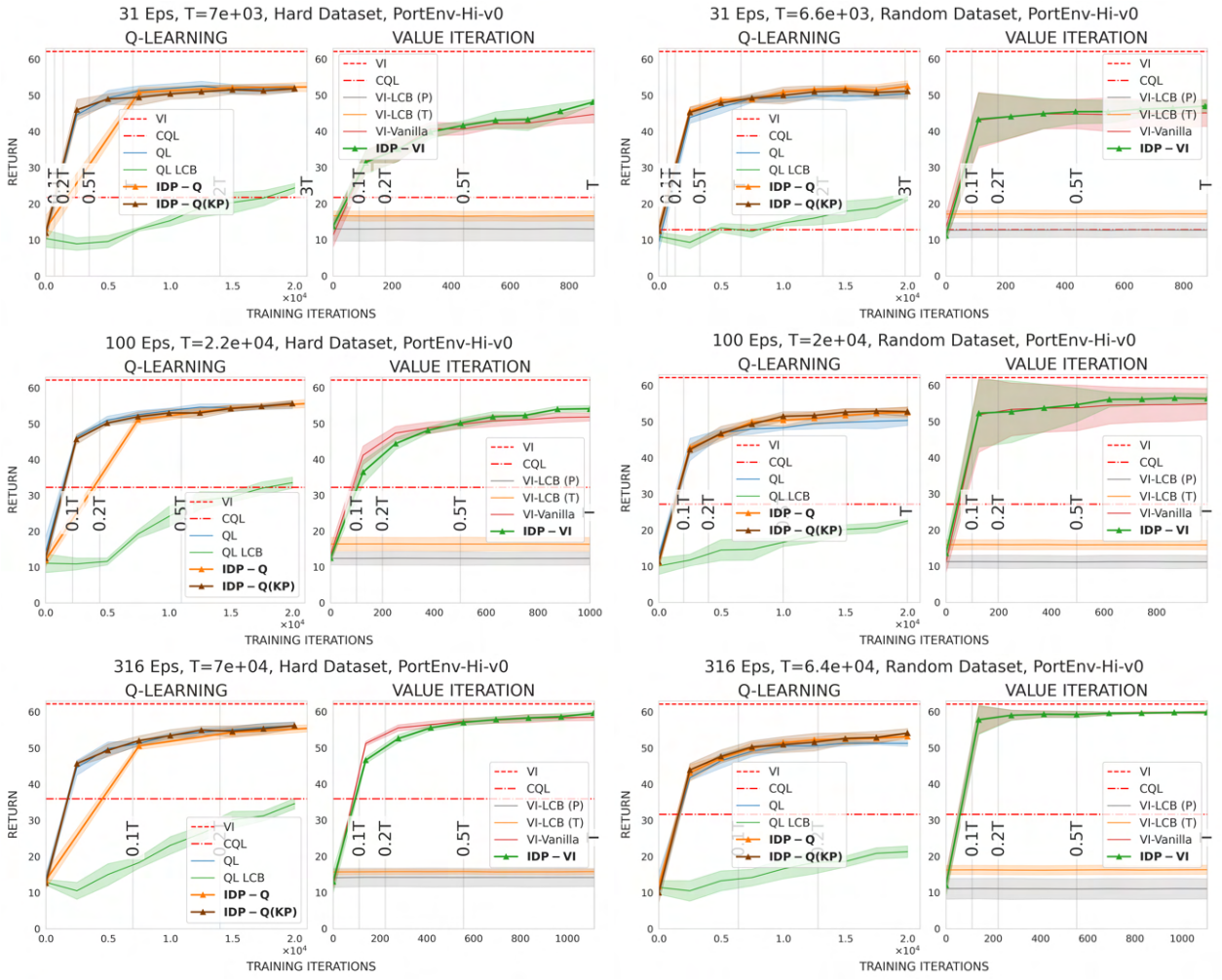

Figure 8: Performance of the policies obtained during training in the Portfolio Environment with asset range $[50, 100]$ (Hi) for the '**Hard**' Dataset and '**Random**' Dataset. The figures display the average portfolio return obtained from the policies trained using different algorithms on the various data set sizes used. The figures indicate more stable performance with our DSD penalty.

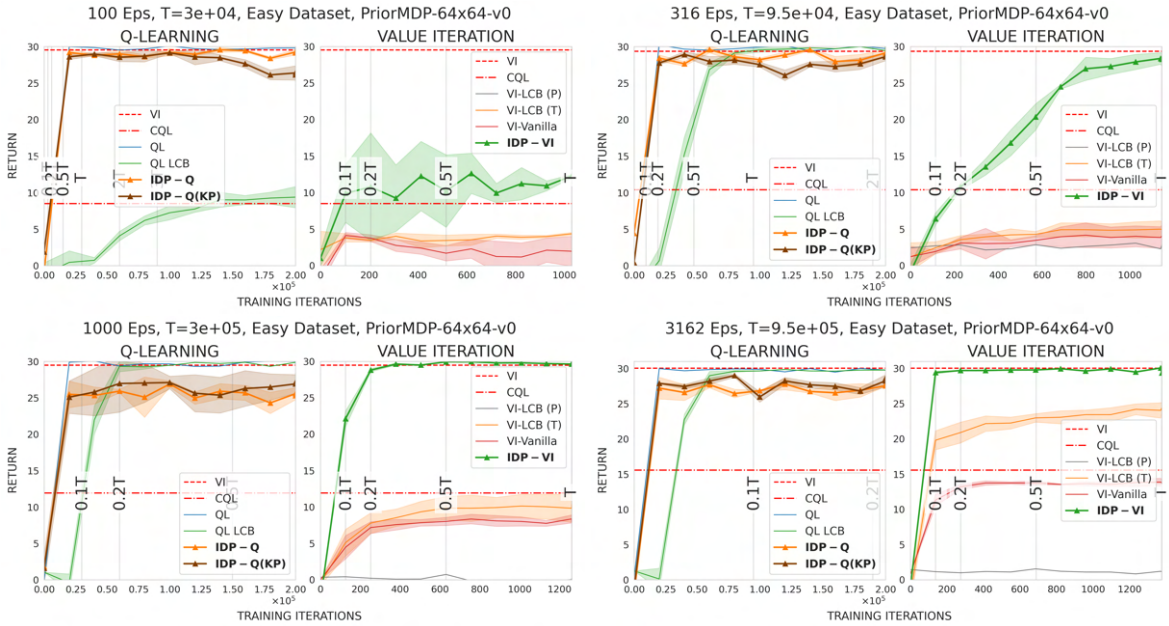

Figure 9: Performance of the policies obtained during training in the Prior MDP Environment with 64 states and 64 actions for the '**Easy**' Dataset over 5 differently seeded training runs. Among model-based methods, the VI-LCB approach is the most sample efficient. Yet given the large amount of optimal data provided, model-free methods perform remarkably well.

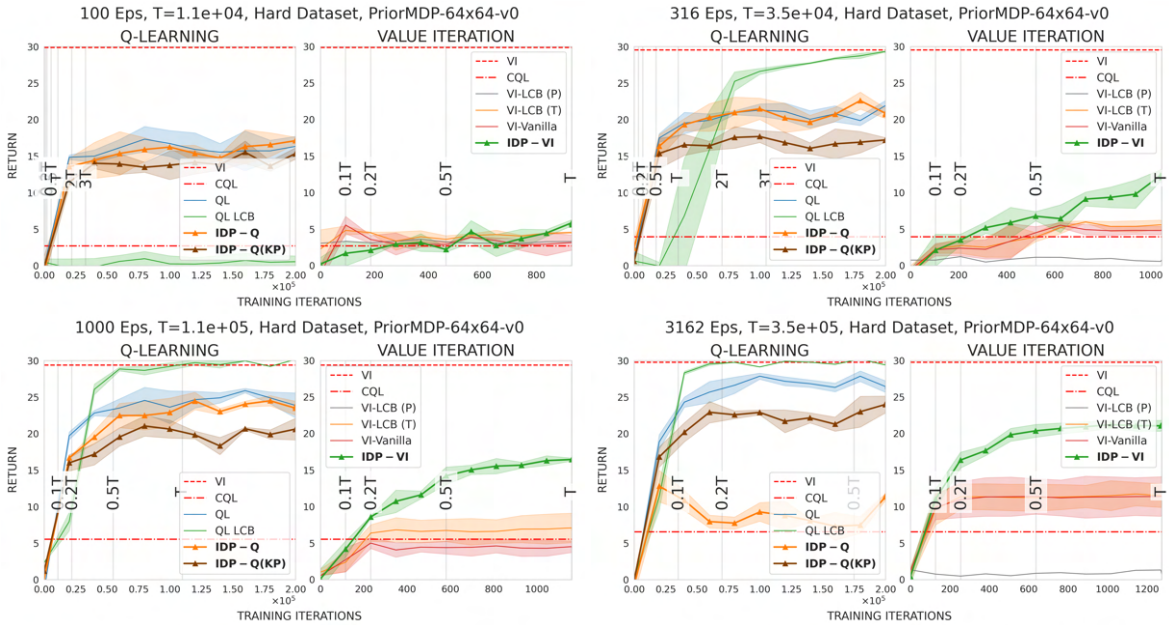

Figure 10: Performance of the policies obtained during training in the Prior MDP Environment with 64 states and 64 actions for the '**Hard**' Dataset over 5 differently seeded training runs. All model-free methods perform comparably with Q-LCB being able to perform well given enough (noisy) optimal data.

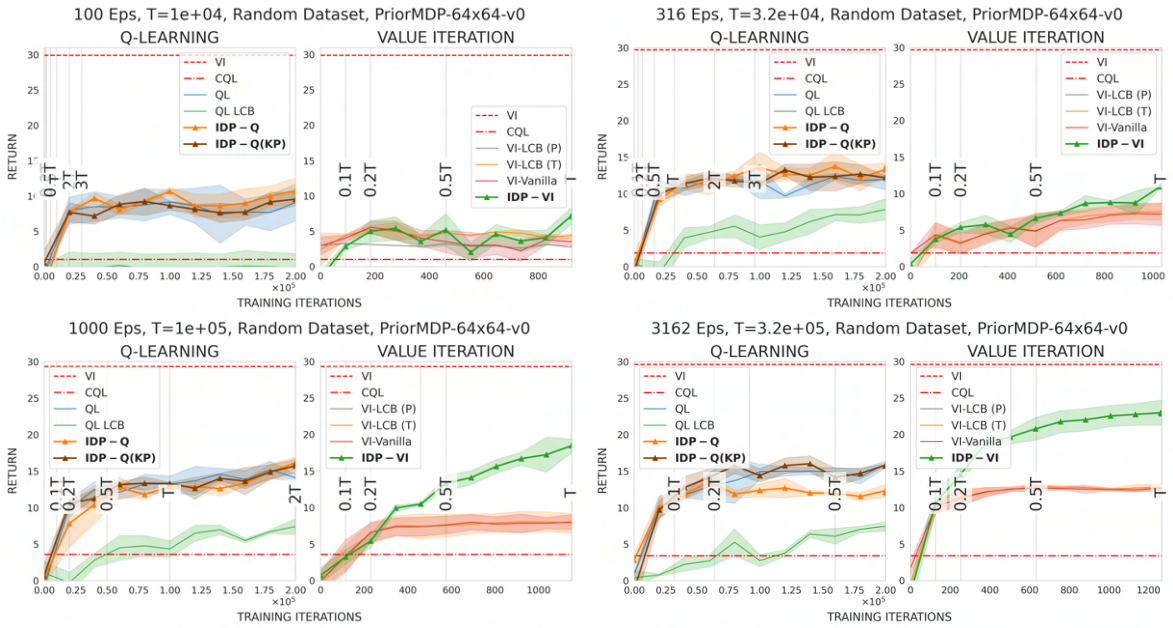

Figure 11: Performance of the policies obtained during training in the Prior MDP Environment with 64 states and 64 actions for the '**Random**' Dataset over 5 differently seeded training runs. It is clear from the figures that VI-LCB is the most sample efficient method in the multimodal setting given minimal optimal data. We posit this is due to its ability to characterize the mismatch more accurately than Azuma-Hoeffding based methods.

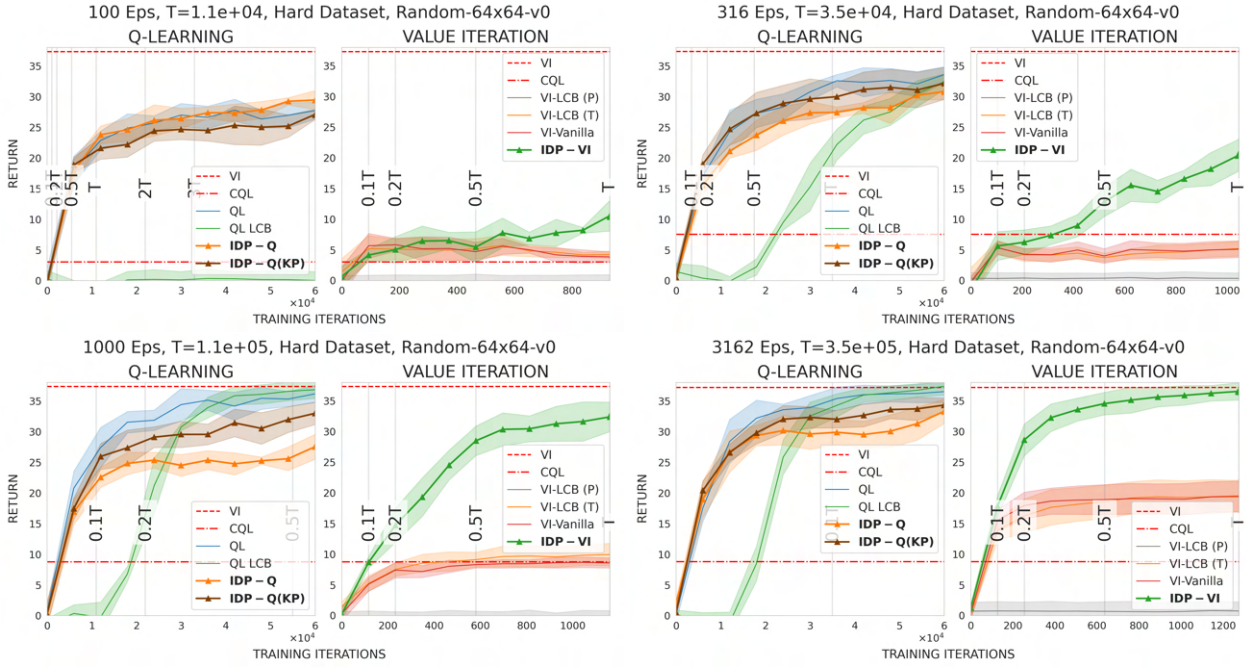

Figure 12: Performance of the policies obtained during training on a random MDP with 64 States and 64 Actions for the '**Hard**' Dataset. We note a consistent gap between competing model-based methods and the optimal score. Model-free algorithms are more sample efficient (e.g. performance at the $0.1T$ interval) given enough optimal data.

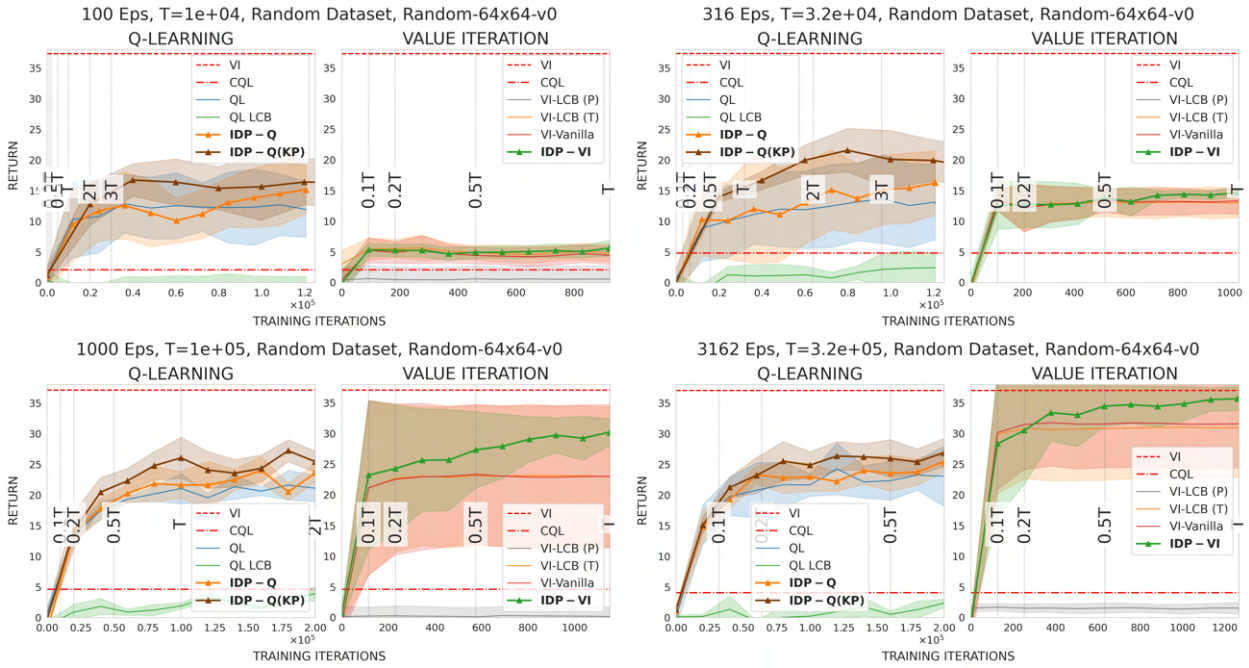

Figure 13: Performance of the policies obtained during training on a random MDP with 64 States and 64 Actions for the '**Random**' Dataset. We note that our form of pessimism helps to reach the optimal performance attained by value iteration (with oracle knowledge) given enough data.

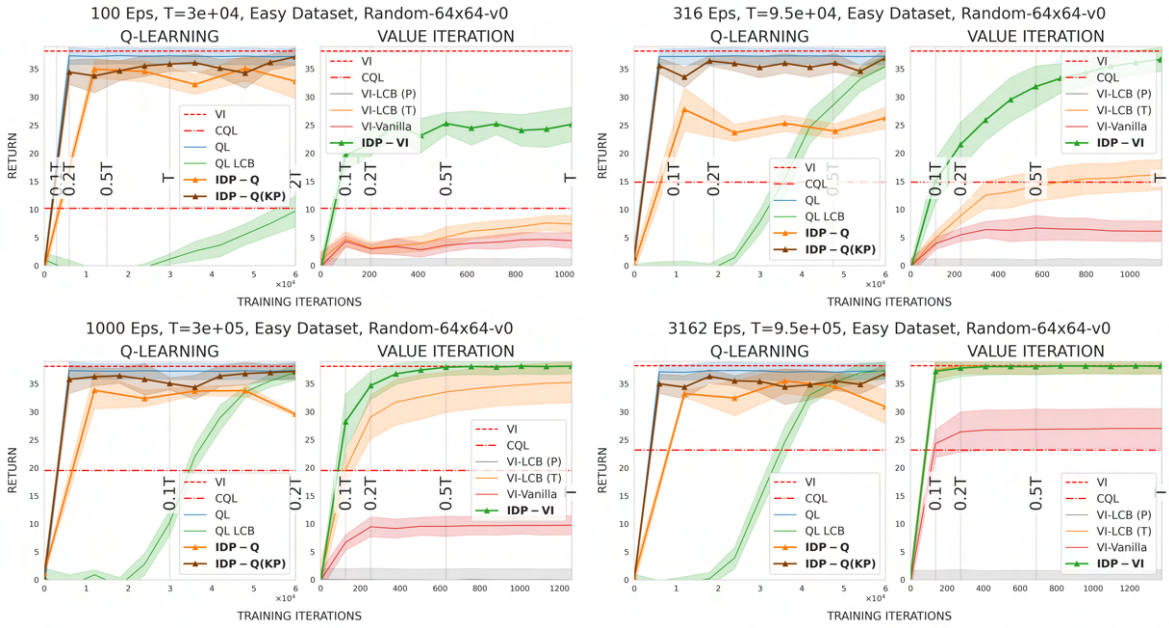

Figure 14: Performance of the policies obtained during training on a random MDP with 64 States and 64 Actions for the '**Easy**' Dataset. We note that for the value iteration experiments, concentration bound based pessimism metrics require significantly larger data to achieve similar performance.

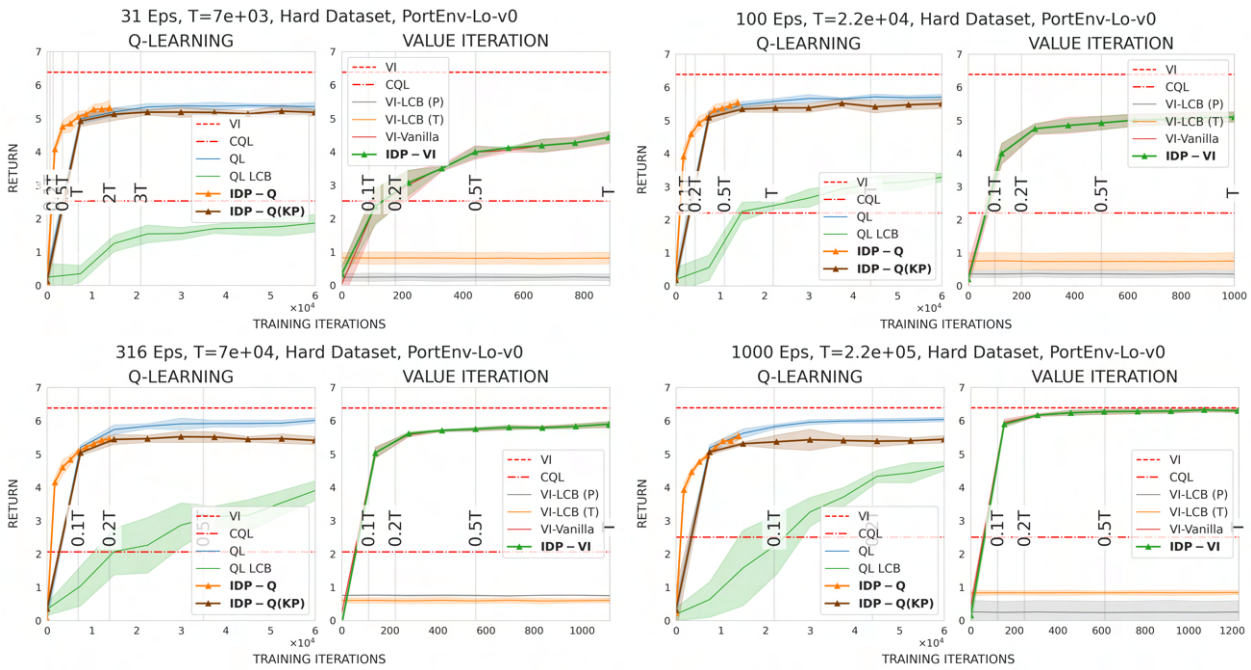

Figure 15: Performance of the policies obtained during training in the Portfolio Environment with asset range $[50, 55]$ (Lo) for the '**Hard**' Dataset. The figures display the average portfolio return obtained from the policies trained using different algorithms on the various data set sizes used. The figures indicate more stable performance with our DSD penalty.

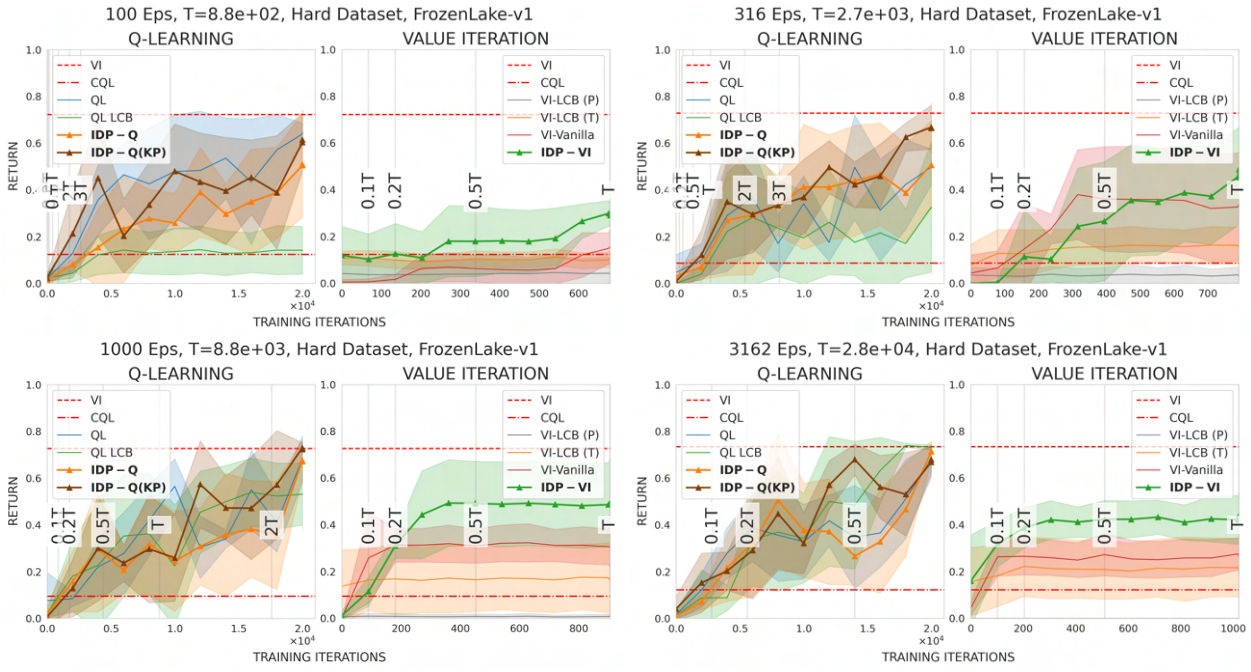

Figure 16: Performance of the policies obtained during training in the Frozen Lake Environment for the '**Hard**' Dataset over 5 differently seeded training runs. It is clear from the figures that over a range of episodes and samples, the return using DSD is higher, with the intuition that the penalty is just enough at every state-action to improve learning.

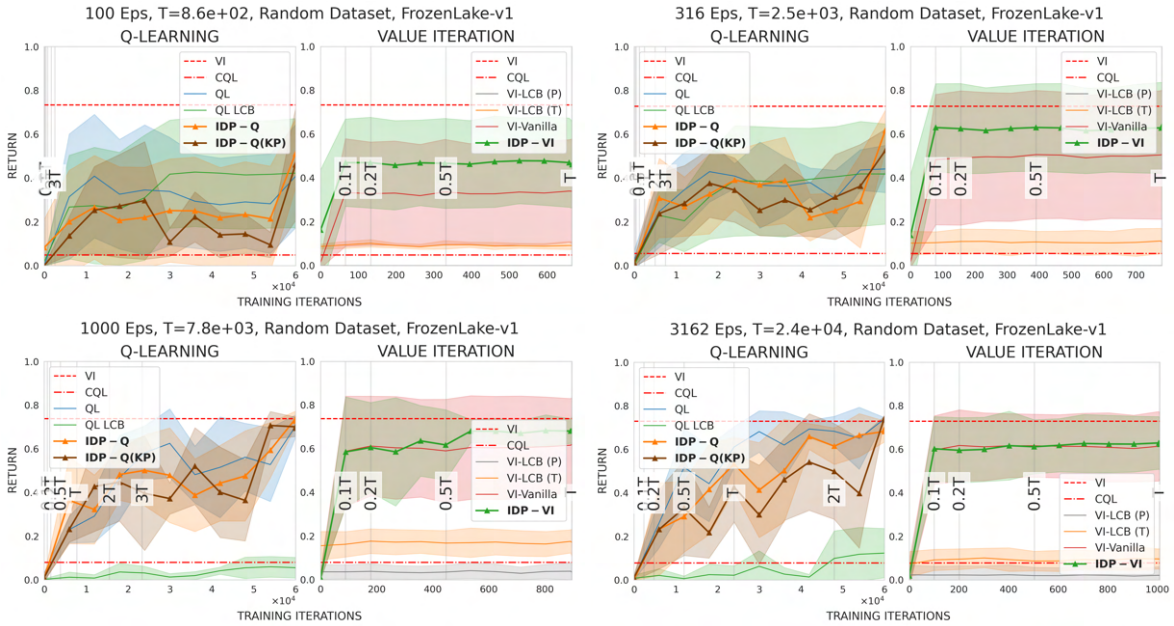

Figure 17: Performance of the policies obtained during training in the Frozen Lake Environment for the '**Random**' Dataset over 5 differently seeded training runs. The figures show comparable performance albeit with slightly increased stability when using a DSD-based penalty.

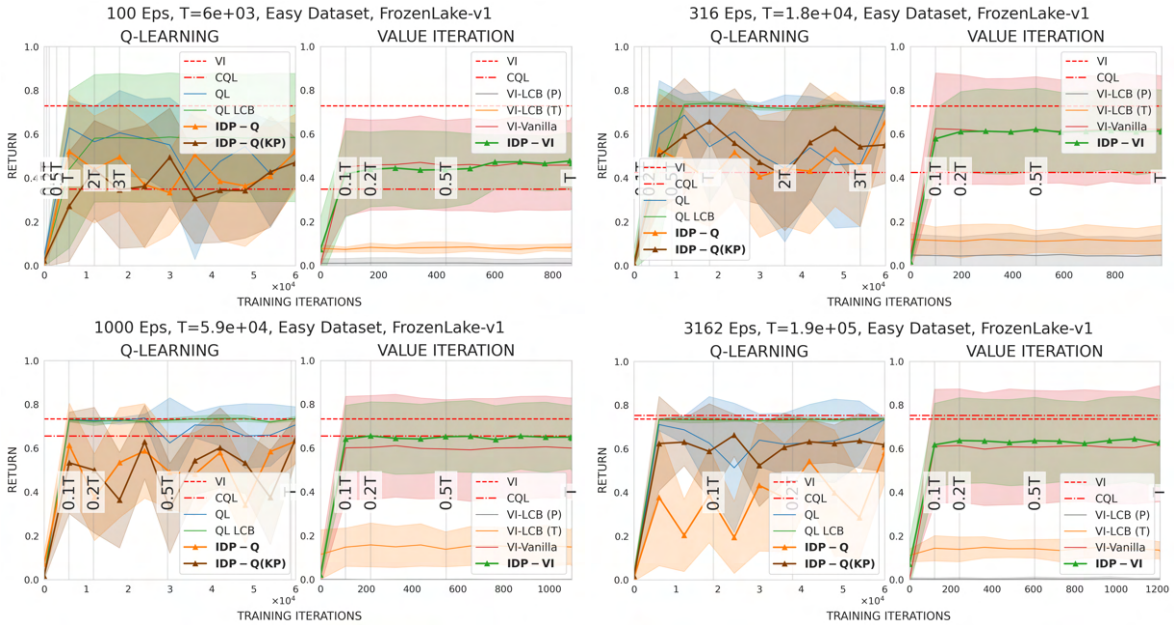

Figure 18: Performance of the policies obtained during training in the Frozen Lake Environment for the '**Easy**' Dataset over 5 differently seeded training runs. We see that all methods (except VI-LCB) perform comparably given the large amount of optimal data provided and the simple unimodally structured environment.

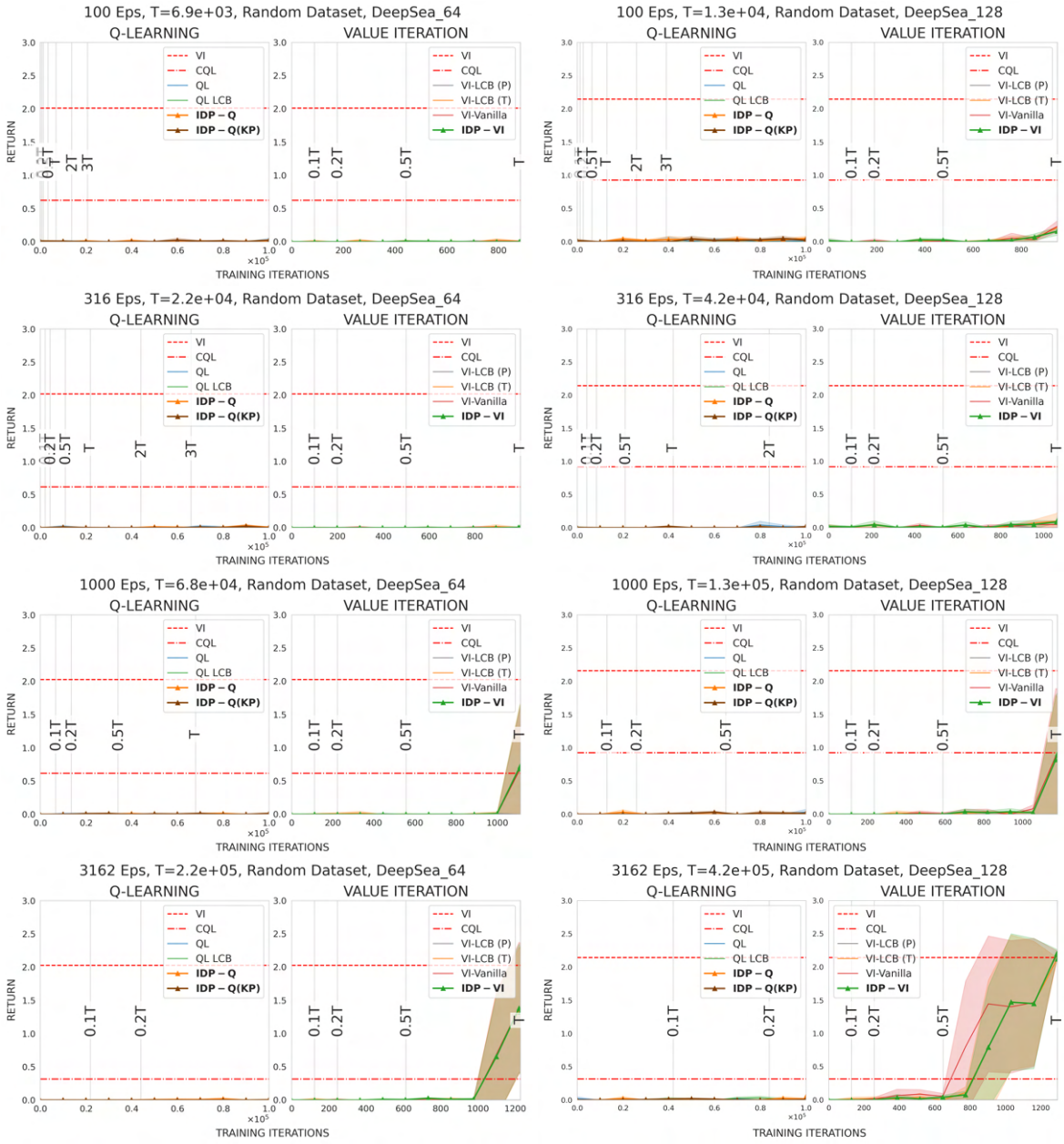

Figure 19: Performance of the policies obtained during training in the DeepSea Environment with 64 states and 128 states for the '**Random**' Dataset over 5 differently seeded training runs. Since this is a sparse reward problem a large number of samples are required for optimal performance. The model-based methods have a comparable advantage since the single sampled trajectory from the optimal policy in the dataset can be used to construct an accurate environment model yielding improved performance as shown in the bottom-right.

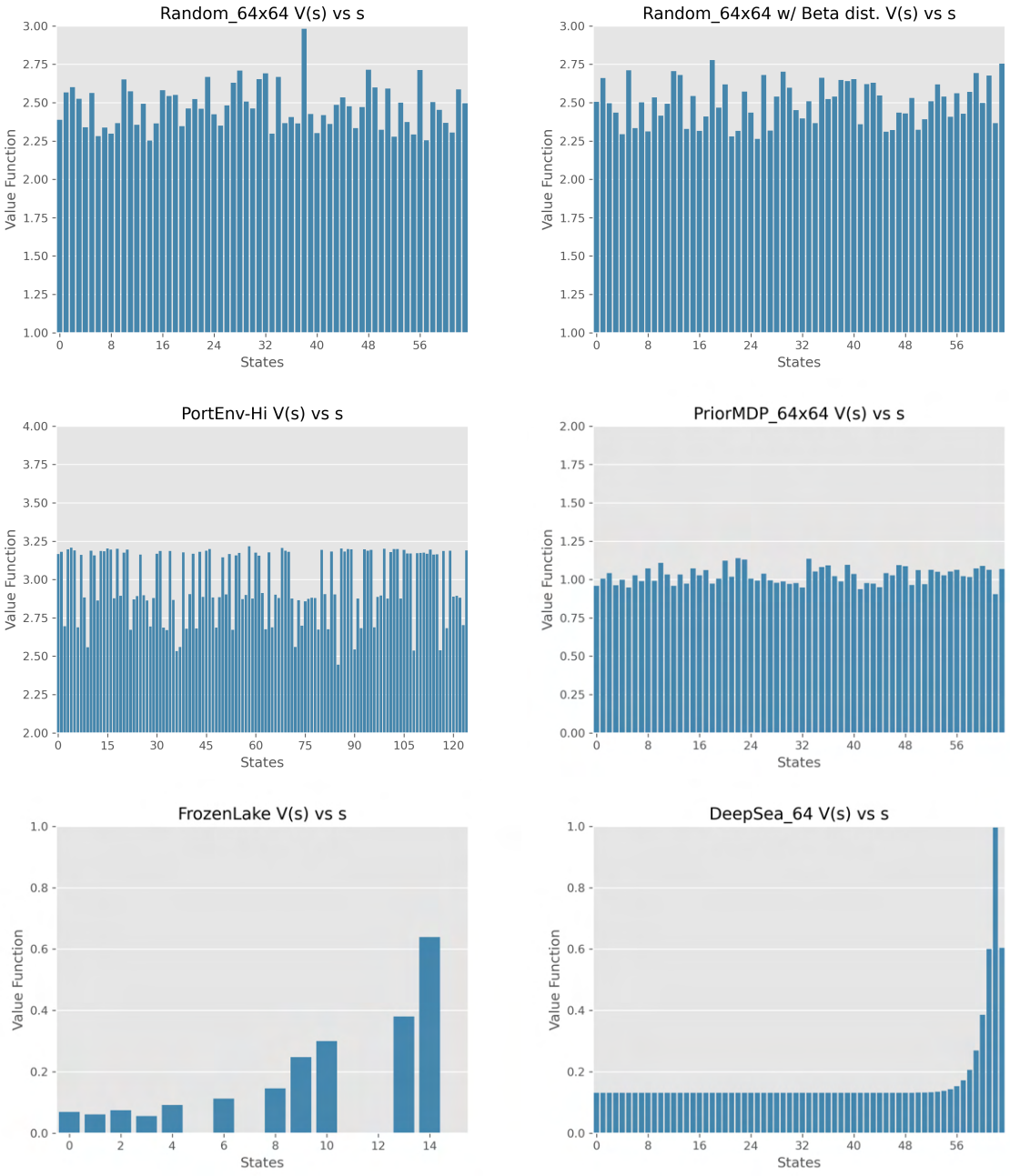

Figure 20: Value function plots (estimating discounted returns) over different states for the environments considered showing the multi-modal nature of the problem.