Bridging Imitation and Online Reinforcement Learning: An Optimistic Tale

Anonymous authors
Paper under double-blind review

Abstract

In this paper, we address the following problem: Given an offline demonstration dataset from an imperfect expert, what is the best way to leverage it to bootstrap online learning performance in MDPs. We first propose an Informed Posterior Sampling-based RL (iPSRL) algorithm that uses the offline dataset, and information about the expert's behavioral policy used to generate the offline dataset. Its cumulative Bayesian regret goes down to zero exponentially fast in N, the offline dataset size if the expert is competent enough. Since this algorithm is computationally impractical, we then propose the iRLSVI algorithm that can be seen as a combination of the RLSVI algorithm for online RL, and imitation learning. Our empirical results show that the proposed iRLSVI algorithm is able to achieve significant reduction in regret as compared to two baselines: no offline data, and offline dataset but used without information about the generative policy. Our algorithm bridges online RL and imitation learning for the first time.

1 Introduction

An early vision of the Reinforcement Learning (RL) field is to design a learning agent that when let loose in an unknown environment, learns by interacting with it. Such an agent starts with a blank slate (with possibly, arbitrary initialization), takes actions, receives state and reward observations, and thus learns by "reinforcement". This remains a goal but at the same time, it is recognized that in this paradigm learning is too slow, inefficient and often impractical. Such a learning agent takes too long to learn near-optimal policies way beyond practical time horizons of interest. Furthermore, deploying an agent that learns by exploration over long time periods may simply be impractical.

In fact, reinforcement learning is often deployed to solve complicated engineering problems by first collecting offline data using a behavioral policy, and then using off-policy reinforcement learning, or imitation learning methods (if the goal is to imitate the policy that generated the offline dataset) on such datasets to learn a policy. This often suffers from the *sim2real* problem, i.e., the learnt policy upon deployment often performs poorly on out-of-distribution state-action space. Thus, there is a need for adaptation and fine-tuning upon deployment.

In this paper, we propose a systematic way to use offline datasets to bootstrap online RL algorithms. Performance of online learning agents is often measured in terms of cumulative (expected) regret. We show that, as expected, there is a gain in performance (reflected in reduction in cumulative regret) of the learning agent as compared to when it did not use such an offline dataset. We call such an online learning agent as being *partially informed*. However, somewhat surprisingly, if the agent is further informed about the behavioral policy that generated the offline dataset, such an *informed (online learning) agent* can do substantially better, reducing cumulative regret significantly. In fact, we also show that if the behavioral policy is suitably parameterized by a *competence parameter*, wherein the behavioral policy is asymptotically the optimal policy, then the higher the "competence" level, the better the performance in terms of regret reduction over the baseline case of no offline dataset.

We first propose an ideal (informed) `iPSRL` (posterior sampling-based RL) algorithm and show via theoretical analysis that under some mild assumptions, its expected cumulative regret is bounded as $\tilde{O}(\sqrt{T})$ where T

is the number of episodes. In fact, we show that if the competence of the expert is high enough (quantified in terms of a parameter we introduce), the regret goes to zero exponentially fast as N, the offline dataset size grows. This is accomplished through a novel prior-dependent regret analysis of the PSRL algorithm, the first such result to the best of our knowledge. Unfortunately, posterior updates in this algorithm can be computationally impractical. Thus, we introduce a Bayesian-bootstrapped algorithm for approximate posterior sampling, called the (informed) `iRLSVI` algorithm (due to its commonality with the RLSVI algorithm introduced in Osband et al. (2019)). The `iRLSVI` algorithm involves optimizing a loss function that is an *optimistic* upper bound on the loss function for MAP estimates for the unknown parameters. Thus, while inspired by the posterior sampling principle, it also has an optimism flavor to it. Through, numerical experiments, we show that the `iRLSVI` algorithm performs substantially better than both the partially informed-RLSVI (which uses the offline dataset naively) as well as the uninformed-RLSVI algorithm (which doesn't use it at all).

We also show that the `iRLSVI` algorithm can be seen as bridging online reinforcement learning with imitation learning since its loss function can be seen as a combination of an online learning term as well as an imitation learning term. And if there is no offline dataset, it essentially behaves like an online RL algorithm. Of course, in various regimes in the middle it is able to interpolate seamlessly. We note that this is the first algorithm of its kind.

Related Work. Because of the surging use of offline datasets for pre-training (e.g., in Large Language models (LLMs), e.g., see Brown et al. (2020); Thoppilan et al. (2022); Hoffmann et al. (2022)), there has been a lot of interest in Offline RL, i.e., RL using offline datasets (Levine et al., 2020). A fundamental issue this literature addresses is RL algorithm design (Nair et al., 2020; Kostrikov et al., 2021; Kumar et al., 2020; Nguyen-Tang & Arora, 2023; Fujimoto et al., 2019; Fujimoto & Gu, 2021; Ghosh et al., 2022) and analysis to best address the "out-of-distribution" (OOD) problem, i.e., policies learnt from offline datasets may not perform so well upon deployment. The dominant design approach is based on 'pessimism' (Jin et al., 2021; Xie et al., 2021a; Rashidinejad et al., 2021) which often results in conservative performance in practice. Some of the theoretical literature (Xie et al., 2021a; Rashidinejad et al., 2021; Uehara & Sun, 2021; Agarwal & Zhang, 2022) has focused on investigation of sufficient conditions such as "concentrability measures" under which such offline RL algorithms can have guaranteed performance. Unfortunately, such measures of offline dataset quality are hard to compute, and of limited practical relevance (Argenson & Dulac-Arnold, 2020; Nair et al., 2020; Kumar et al., 2020; Levine et al., 2020; Kostrikov et al., 2021; Wagenmaker & Pacchiano, 2022).

There is of course, a large body of literature on online RL (Dann et al., 2021; Tiapkin et al., 2022; Ecoffet et al., 2021; Guo et al., 2022; Ecoffet et al., 2019; Osband et al., 2019) with two dominant design philosophies: Optimism-based algorithms such as UCRL2 in Auer et al. (2008), and Posterior Sampling (PS)-type algorithms such as PSRL (Osband et al., 2013; Ouyang et al., 2017), etc. (Osband et al., 2019; 2016; Russo & Van Roy, 2018; Zanette & Sarkar, 2017; Hao & Lattimore, 2022). However, none of these algorithms consider starting the learning agent with an offline dataset. Of course, imitation learning (Hester et al., 2018; Beliaev et al., 2022; Schaal, 1996) is exactly concerned with learning the expert's behavioral policy (which may not be optimal) from the offline datasets but with no online finetuning of the policy learnt. Several papers have actually studied bridging offline RL and imitation learning (Ernst et al., 2005; Kumar et al., 2022; Rashidinejad et al., 2021; Hansen et al., 2022; Vecerik et al., 2017; Lee et al., 2022). Some have also studied offline RL followed by a small amount of policy fine-tuning (Song et al., 2022; Fang et al., 2022; Xie et al., 2021b; Wan et al., 2022; Schrittwieser et al., 2021; Ball et al., 2023; Uehara & Sun, 2021; Xie et al., 2021b; Agarwal & Zhang, 2022) with the goal of finding policies that optimize simple regret.

But none have studied the problem we introduce and study in this paper: Namely, given an offline demonstration dataset from an imperfect expert, what is the best way to leverage it to bootstrap online learning performance (in terms of cumulative regret) in MDPs. What is the best regret reduction that is achievable by use of offline datasets? How it depends on the quality and quantity of demonstrations, and what algorithms can one devise to achieve them? And does any information about the offline-dataset generation process help in regret reduction? We answer some of these questions in this paper.

2 Preliminaries

Episodic Reinforcement Learning. Consider a scenario where an agent repeatedly interacts with an environment modelled as a finite-horizon MDP, and refer to each interaction as an episode. The finite-horizon MDP is represented by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, H, \nu)$, where $\mathcal{S}$ is a finite state space (of size S), $\mathcal{A}$ is a finite action space (of size A), P encodes the transition probabilities, r is the reward function, H is the time horizon length, and ν is the initial state distribution. The interaction protocol is as follows: at the beginning of each episode t, the initial state s_0^t is independently drawn from ν. Then, at each period $h = 0, 1, \ldots, H-1$ in episode t, if the agent takes action $a_h^t \in \mathcal{A}$ at the current state $s_h^t \in \mathcal{S}$, then it will receive a reward $r_h(s_h^t, a_h^t)$ and transit to the next state $s_{h+1}^t \in P_h(\cdot|s_h^t, a_h^t)$. An episode terminates once the agent arrives at state s_H^t in period H and receives a reward $r_H(s_H^t)$. We abuse notation for the sake of simplicity, and just use $r_H(s_H^t, a_H^t)$ instead of $r_H(s_H^t)$, though no action is taken at period H. The objective is to maximize its expected total reward over T episodes.

Let Q_h^* and V_h^* respectively denote the optimal state-action value and state value functions at period h. Then, the Bellman equation for MDP $\mathcal{M}$ is

$$Q_h^*(s,a) = r_h(s,a) + \sum_{s'} P_h(s'|s,a) V_{h+1}^*(s'), \tag{1}$$

where $V_{h+1}^*(s') := \max_b Q_{h+1}^*(s', b)$, if $h < H-1$ and $V_{h+1}^*(s') = 0$, if $h = H-1$. We define a policy π as a mapping from a state-period pair to a probability distribution over the action space A. A policy π^* is optimal if $\pi_h^*(\cdot|s) \in \arg\max_{\pi_h} \sum_a Q_h^*(s,a)\pi_h(a|s)$ for all $s \in \mathcal{S}$ and all h.

Agent's Prior Knowledge about $\mathcal{M}$. We assume that the agent does not fully know the environment $\mathcal{M}$; otherwise, there is no need for learning and this problem reduces to an optimization problem. However, the agent usually has some prior knowledge about the unknown part of $\mathcal{M}$. For instance, the agent might know that $\mathcal{M}$ lies in a low-dimensional subspace, and/or have a prior distribution over $\mathcal{M}$. We use the notation $\mathcal{M}(\theta)$ where θ parameterizes the unknown part of the MDP. When we want to emphasize it as a random quantity, we will denote it by θ^*. Of course, different assumptions about the agent's prior knowledge lead to different problem formulations and algorithm designs. As a first step, we consider two canonical settings:

- **Tabular RL:** The agent knows $\mathcal{S}, \mathcal{A}, r, H$ and ν, but does not know P. That is, $\theta^* = P$ in this setting. We also assume that the agent has a prior over P, and this prior is independent across state-period-action triples.
- **Linear value function generalization:** The agent knows $\mathcal{S}, \mathcal{A}, H$ and ν, but does not know P and r. Moreover, the agent knows that for all h, Q_h^* lies in a low-dimensional subspace span(Φ_h), where $\Phi_h \in \Re^{|S||A| \times d}$ is a known matrix. In other words, $Q_h^* = \Phi_h \theta_h^*$ for some $\theta_h^* \in \Re^d$. Thus, in this setting $\theta^* = \left[\theta_0^{*\top}, \ldots, \theta_{H-1}^{*\top}\right]^\top$. We also assume that the agent has a Gaussian prior over θ^*.

As we will discuss later, the insights developed in this paper could potentially be extended to more general cases.

Offline Datasets. We denote an *offline dataset* with L episodes as $\mathcal{D}_0 = \{(\bar{s}_0^l, \bar{a}_0^l, \cdots, \bar{s}_H^l)_{l=0}^{L-1}\}$, where $N = HL$ denotes the dataset size in terms of number of observed transitions. For the sake of simplicity, we assume we have complete trajectories in the dataset but it can easily be generalized if not. We denote an *online dataset* with t episodes as $\mathcal{H}_t = \{(s_0^l, a_0^l, \cdots, s_H^l)_{l=0}^{t}\}$ and $\mathcal{D}_t = \mathcal{D}_0 \oplus \mathcal{H}_t$.

The Notion of Regret. A online learning algorithm ϕ is a map for each episode t, and time h, $\phi_{t,h} : \mathcal{D}_t \to \Delta_A$, the probability simplex over actions. We define the Bayesian regret of an online learning algorithm ϕ over T episodes as

$$\mathfrak{BR}_T(\phi) := \mathbb{E}\left[\sum_{t=1}^{T} \left(V_0^*(s_0^t; \theta^*) - \sum_{h=0}^{H} r_h(s_h^t, a_h^t)\right)\right],$$

where the (s_h^t, a_h^t)'s are the state-action tuples from using the learning algorithm ϕ, and the expectation is over the sequence induced by the interaction of the learning algorithm and the environment, the prior distributions over the unknown parameters θ^* and the offline dataset $\mathcal{D}_0$.

Expert's behavioral policy and competence. We assume that the expert that generated the offline demonstrations may not be perfect, i.e., the actions it takes are only approximately optimal with respect to the optimal Q-value function. To that end, we model the expert's policy by use of the following generative model,

$$\pi_h^\beta(a|s) = \frac{\exp(\beta(s)Q_h^*(s,a))}{\sum_a \exp(\beta(s)Q_h^*(s,a))}, \tag{2}$$

where $\beta(s) \geq 0$ is called the *state-dependent deliberateness* parameter, e.g., when $\beta(s) = 0$, the expert behaves naively in state s, and takes actions uniformly randomly. When $\beta(s) \to \infty$, the expert uses the optimal policy when in state s. When $\beta(\cdot)$ is unknown, we will assume an independent exponential prior for the sake of analytical simplicity, $f_2(\beta(s)) = \lambda_2 \exp(-\lambda_2 \beta(s))$ over $\beta(s)$ where $\lambda_2 > 0$ is the same for all s. In our experiments, we will regard $\beta(s)$ as being the same for all states, and hence a single parameter.

The above assumes the expert is knowledgeable about Q^*. However, it may know it only approximately. To model that, we introduce a *knowledgeability* parameter $\lambda \geq 0$. The expert then knows $\tilde{Q}$ which is distributed as $\mathcal{N}(Q^*, \mathbb{I}/\lambda^2)$ conditioned on θ, and selects actions according to the softmax policy equation 2, with the Q^* replaced by $\tilde{Q}$. The two parameters (β, λ) together will be referred to as the *competence* of the expert. In this case, we denote the expert's policy as $\pi_h^{\beta,\lambda}$.

Remark 2.1. While the form of the generative policy in eq. equation 2 seems specific, $\pi_h^\beta(\cdot|s)$ is a random vector with support over the entire probability simplex. In particular, if one regards $\beta(s)$ and $\tilde{Q}_h(s,\cdot)$ as parameters that parameterize the policy, the softmax policy structure as in equation 2 is enough to realize any stationary policy.

Furthermore, we note that our main objective here is to yield clear and useful insights when information is available to be able to model the expert's behavioral policy with varying competence levels. Other forms of generative policies can also be used including ϵ-optimal policies introduced in (Beliaev et al., 2022), and the framework extended.

3 The Informed PSRL Algorithm

We now introduce a simple *Informed Posterior Sampling-based Reinforcement Learning* (`iPSRL`) algorithm that naturally uses the offline dataset $\mathcal{D}_0$ and action generation information to construct an informed prior distribution over θ^*. The realization of θ^* is assumed known to the expert (but not the learning agent) with $\tilde{Q}(\cdot,\cdot;\theta^*) = Q(\cdot,\cdot;\theta^*)$, and $\beta(s) := \beta \geq 0$ (i.e., it is state-invariant) is also known to the expert. Thus, the learning agent's posterior distribution over θ^* given the offline dataset is,

$$\begin{aligned} &\mathbb{P}(\theta^* \in \cdot|\mathcal{D}_0) \propto \mathbb{P}(\mathcal{D}_0|\theta^* \in \cdot)\mathbb{P}(\theta^* \in \cdot) \\ =\ & \mathbb{P}(\theta^* \in \cdot) \times \int_{\theta\in\cdot} \prod_{l=0}^{L-1}\prod_{h=0}^{H-1} \theta(\bar{s}_l^{h+1}|\bar{s}_l^h, \bar{a}_l^h)\pi_h^\beta(\bar{a}_l^h|\bar{s}_l^h,\theta)\nu(\bar{s}_l^0)\,d\theta. \end{aligned} \tag{3}$$

A PSRL agent (Osband et al., 2013; Ouyang et al., 2017) takes this as the prior, and then updates the posterior distribution over θ^* as online observation tuples, (s_t, a_t, s_t', r_t) become available. Such an agent is really an ideal agent with assumed posterior distribution updates being exact. In practice, this is computationally intractable and we will need to get samples from an approximate posterior distribution, an issue which we will address in the next section.

3.1 Prior-dependent Regret Bound

It is natural to expect some regret reduction if an offline demonstration dataset is available to warm-start the online learning. However, the degree of improvement must depend on the "quality" of demonstrations, for example through the competence parameter β. Further note that the role of the offline dataset is via the prior distribution the PSRL algorithm uses. Thus, theoretical analysis involves obtaining a prior-dependent regret bound, which we obtain next.

We use $\mathbb{H}$ to denote Shannon entropy (with the natural logarithm). We start by establishing a Bayesian regret bound of PSRL algorithm for MDPs with any prior distribution.

Lemma 3.1. *Let ν be the prior distribution of π^*, then the PSRL algorithm satisfies*

$$\mathfrak{BR}_T(\phi^{PSRL}) \leq \min\Big\{H\sqrt{(SA)^H\mathbb{H}(\nu)T/2}, \sqrt{S^2A^2H^4T\log(STH)}\Big\}.$$

The proof can be found in the Appendix. Note that the first part of the above bound reflects the prior effect though $\mathbb{H}(\nu)$ while the second part is prior-independent. This gives us the following corollary for the `iPSRL` algorithm:

Corollary 3.2. *For the* `iPSRL` *algorithm,*

$$\mathfrak{BR}_T(\phi^{iPSRL}) \leq \min\left\{H\sqrt{(SA)^H\mathbb{H}(\pi^*|\mathcal{D}_0)T/2}, \sqrt{S^2A^2H^4T\log(STH)}\right\}. \tag{4}$$

Proof. Conditioning on $\mathcal{D}_0 = \bar{\mathcal{D}}_0$, by applying Lemma 3.1, we can obtain

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(V_0^*(s_0^t;\theta^*) - \sum_{h=0}^{H} r_h(s_h^t, a_h^t)\right)|\mathcal{D}_0 = \bar{\mathcal{D}}_0\right] \leq H\sqrt{(SA)^H\mathbb{H}(\pi^*|\mathcal{D}_0 = \bar{\mathcal{D}}_0)T/2}.$$

The corollary then follows by taking expectations on both sides over $\mathcal{D}_0$ along with the concavity of the square root function. $\square$

The conditional information $\mathbb{H}(\pi^*|\mathcal{D}_0)$ measures the amount of randomness of π^* given $\mathcal{D}_0$. Therefore, Corollary 3.2 means that the more certain about π^* we are, the less regret the `iPSRL` algorithm will incur. In the rest of the proof, we provide an upper bound for $\mathbb{H}(\pi^*|\mathcal{D}_0)$ by use of Fano's inequality. For positive integer K, let $[K] := \{1, 2, \dots, K\}$.

Lemma 3.3 (Fano's Inequality)**.** *Let $Y, \hat{Y}$ be random variables on $[K]$ such that* $\Pr(Y \neq \hat{Y}) \leq \varepsilon \leq 1/2$. *Then,*

$$\mathbb{H}(Y|\hat{Y}) \leq \varepsilon \log K + \mathbf{h}(\varepsilon), \tag{5}$$

where $\mathbf{h}(\epsilon) = -\epsilon\log\epsilon - (1-\epsilon)\log(1-\epsilon)$ *is the binary entropy function.*

We assume the following about the prior distribution of θ^*.

Assumption 3.4. There exists a $\Delta > 0$ such that for all $\theta \in \Theta$, $h \in [H]$, and $s \in \mathcal{S}$, there exists an $a^* \in \mathcal{A}$ such that $Q_h(s, a^*; \theta) \geq Q_h(s, a'; \theta) + \Delta,\ \forall a' \in \mathcal{A}\backslash\{a^*\}$.

Define $p_h(s;\theta) := \Pr_{\theta,\pi^*(\theta)}(s_h = s)$.

Assumption 3.5. The infimum probability of any reachable state, defined as

$$\underline{p} := \inf\{p_h(s;\theta) : h \in [H], s \in \mathcal{S}, \theta \in \Theta, p_h(s;\theta) > 0\}$$

satisfies $\underline{p} > 0$.

We now describe a procedure to construct an estimator $\hat{\pi}^*$ from $\mathcal{D}_0$ so that $\Pr(\pi^* \neq \hat{\pi}^*)$ is small. Fix an integer N, and choose a $\delta \in (0, 1)$. For each $\theta \in \Theta$, define a deterministic Markov policy $\pi^*(\theta) = (\pi_h^*(\cdot;\theta))_{h=1}^H$ sequentially through

$$\pi_h^*(s;\theta) = \begin{cases} \arg\max_a Q_h(s,a;\theta), & \text{if } \Pr_{\theta,\pi_{1:h-1}^*(\theta)}(s_h = s) > 0 \\ \bar{a}_0, & \text{if } \Pr_{\bar{\theta},\pi_{1:h-1}^*(\bar{\theta})}(s_h = s) > 0, \end{cases} \tag{6}$$

where the tiebreaker for the argmax operation is based on a fixed order on actions, and $\bar{a}_0 \in \mathcal{A}$ is a fixed action in $\mathcal{A}$. It is clear that $\pi^*(\theta)$ is an optimal policy for the MDP θ. Furthermore, for those states that are impossible to be visited, we choose to take a fixed action $\bar{a}_0$. Although the choice of action at those states doesn't matter, our construction will be helpful for the proofs.

Construction of $\hat{\pi}^*$: Let $N_h(s)$ (resp. $N_h(s,a)$) be the number of times state s (resp. state-action pair (s,a)) appears at time h in dataset $\mathcal{D}_0$. Define $\hat{\pi}^*$ to be such that:

- $\hat{\pi}^*_h(s) = \arg\max_{a\in\mathcal{A}} N_h(s,a)$ (ties are broken through some fixed ordering of actions) whenever $N_h(s) \geq \delta N$;
- $\hat{\pi}^*_h(s) = \bar{a}_0$ whenever $N_h(s) < \delta N$. $\bar{a}_0$ is a fixed action in $\mathcal{A}$ that was used in the definition of $\pi^*(\theta)$.

The idea of the proof is that for sufficiently large β and N, we can choose a $\delta \in (0,1)$ such that

- *Claim 1:* If $s \in \mathcal{S}$ is probable at time h under $\pi^*(\theta)$, then $N_h(s) \geq \delta N$ with large probability. Furthermore, $\pi^*_h(s) = \arg\max_{a\in\mathcal{A}} N_h(s,a)$ with large probability as well.
- *Claim 2:* If $s \in \mathcal{S}$ is improbable at time h under $\pi^*(\theta)$, then $N_h(s) < \delta N$ with large probability;

Given the two claims, we can then conclude that $\pi^* = \hat{\pi}^*$ with high probability via a standard union bound argument.

Lemma 3.6. *Let X be the sum of N i.i.d. Bernoulli random variables with mean $p \in (0,1)$. Let $q \in (0,1)$, then*

$$\begin{aligned} \Pr(X \leq qN) &\leq \exp\left(-2N(q-p)^2\right), \qquad \text{if } q < p, \\ \Pr(X \geq qN) &\leq \exp\left(-2N(q-p)^2\right), \qquad \text{if } q > p. \end{aligned}$$

Proof. Both inequalities can be obtained by applying Hoeffding's Inequality. □

Lemma 3.7. *Let Δ and $\underline{p}$ be as in Assumptions 3.4 and 3.5 respectively and let*

$$\underline{\beta} := [\log 3 - \log \underline{p} + \log(H-1) + \log(A-1)]/\Delta.$$

For any $\beta \geq \underline{\beta}$ and $N \in \mathbb{N}$, there exists an estimator $\hat{\pi}^*$ constructed from $\mathcal{D}_0$ that satisfies

$$\Pr(\pi^* \neq \hat{\pi}^*) \leq SH\left[\exp\left(-\frac{N\underline{p}^2}{18}\right) + \exp\left(-\frac{N\underline{p}}{36}\right)\right].$$

The proof is available in the appendix.

Theorem 3.8. *Let $\beta \geq \underline{\beta}$, then for all N such that $\varepsilon_N \leq 1/2$, we have*

$$\mathfrak{BR}_T(\phi^{iPSRL}) \leq \min\left\{\sqrt{S^2A^2H^4T\log(STH)}, H\sqrt{(SA)^H(\varepsilon_N SH\log A + \mathbf{h}(\varepsilon_N))T/2}\right\}. \tag{7}$$

where

$$\varepsilon_N = SH\left[\exp\left(-\frac{N\underline{p}^2}{18}\right) + \exp\left(-\frac{N\underline{p}}{36}\right)\right].$$

Proof. Since $\hat{\pi}^*$ is a function of $\mathcal{D}_0$, we have $\mathbb{H}(\pi^*|\mathcal{D}_0) \leq \mathbb{H}(\pi^*|\hat{\pi}^*)$. The result then follows from Lemma 3.2, Lemma 3.3, and Lemma 3.7. □

Note that using the inequality $\mathbf{h}(\varepsilon) \leq 2\sqrt{\varepsilon} - \varepsilon$ for all $\varepsilon \in (0,1)$, we see that the right-hand side of equation 7 converges to zero exponentially fast as $N \to \infty$.

Remark 3.9. (a) For fixed N, and large S and A, the regret bound is $\tilde{O}(SAH^2\sqrt{T})$, which possibly could be improved in H. (b) For a suitably large β, the regret bound obtained goes to zero exponentially fast as N, the offline dataset size, goes to infinity thus indicating the online learning algorithm's ability to learn via imitation of the expert. (c) Corollary 3.2 can be improved to remove the exponential dependency on H by using the Cauchy–Schwarz inequality in the space of state-action occupancy measures. Such technique has been successfully used in (Hao & Lattimore, 2022) in a purely online setting. We leave this refinement as a part of future work.

4 Approximating iPSRL

4.1 The Informed RLSVI Algorithm

The `iPSRL` algorithm introduced in the previous section assumes that posterior updates can be done exactly. In practice, the posterior update in Eq. equation 3 is challenging due to the loss of conjugacy while using the Bayes rule. Thus, we must find a computationally efficient way to do approximate posterior updates (and obtain samples from it) to enable practical implementation. Hence, we propose a novel approach based on Bayesian bootstrapping to obtain approximate posterior samples. The key idea is to perturb the loss function for the maximum a posterior (MAP) estimate and use the point estimate as a surrogate for the exact posterior sample.

Note that in the ensuing, we regard β as also unknown to the learning agent (and $\lambda = \infty$ for simplicity). Thus, the learning agent must form a belief over both θ and β via a joint posterior distribution conditioned on the offline dataset $\mathcal{D}_0$ and the online data at time t, $\mathcal{H}_t$. We denote the prior pdf over θ by $f(\cdot)$ and prior pdf over β by $f_2(\cdot)$.

For the sake of compact notation, we denote $Q_h^*(s, a; \theta)$ as $Q_h^\theta(s, a)$ in this section. Now, consider the offline dataset,

$$\mathcal{D}_0 = \{((s_h^l, a_h^l, \check{s}_h^l, r_h^l)_{h=0}^{H-1})_{l=1}^{L}\}$$

and denote $\theta = (\theta_h)_{h=0}^{H-1}$. We introduce the *temporal difference error* $\mathcal{E}_h^l$ (parameterized by a given Q^θ),

$$\mathcal{E}_h^l(Q^\theta) := \left(r_h^l + \max_b Q_{h+1}^\theta(\check{s}_h^l, b) - Q_h^\theta(s_h^l, a_h^l) \right).$$

We will regard Q_h^θ to only be parameterized by θ_h, i.e., $Q_h^{\theta_h}$ but abuse notation for the sake of simplicity. We use this to construct a *parameterized offline dataset*,

$$\mathcal{D}_0(Q^\theta) = \{((s_h^l, a_h^l, \check{s}_h^l, \mathcal{E}_h^l(Q^\theta))_{h=0:H-1})_{l=1:L}\}.$$

A parametrized online dataset $\mathcal{H}_t(Q^\theta)$ after episode t can be similarly defined. To ease notation, we will regard the jth episode during the online phase as the $(L + j)$th observed episode. Thus,

$$\mathcal{H}_t(Q^\theta) = \{((s_h^k, a_h^k, \check{s}_h^k, \mathcal{E}_h^k(Q^\theta))_{h=0:H-1})_{k=L+1:L+t}\},$$

the dataset observed during the online phase by episode t.

Note that Q^θ is to be regarded as a parameter. Now, at time t, we would like to obtain a **MAP estimate** for (θ, β) by solving the following:

$$\textbf{MAP:} \quad \arg\max_{\theta,\beta} \log P(\mathcal{H}_t(Q^\theta)|\mathcal{D}_0(Q^\theta), \theta, \beta) + \log P(\mathcal{D}_0(Q^\theta)|\theta, \beta) + \log f(\theta) + \log f_2(\beta). \tag{8}$$

Denote a perturbed version of the Q^θ-parameterized offline dataset by

$$\tilde{\mathcal{D}}_0(Q^\theta) = \{((s_h^l, \tilde{a}_h^l, \check{s}_h^l, \tilde{\mathcal{E}}_h^l)_{h=0:H-1})_{l=1:L}\}$$

where random perturbations are added: (i) actions have perturbation $w_l^h \sim \exp(1)$, (ii) rewards have perturbations $z_h^l \sim \mathcal{N}(0, \sigma^2)$, and (iii) the prior $\tilde{\theta} \sim \mathcal{N}(0, \Sigma_0)$.

Note that the first and second terms involving $\mathcal{H}_t$ and $\mathcal{D}_0$ in equation 8 are independent of β when conditioned on the actions. Thus, we have a sum of *log-likelihood of TD error, transition and action* as follows:

$$\begin{aligned}\log P(\tilde{\mathcal{D}}_0(Q^\theta)|Q^\theta_{0:H}) &= \sum_{l=1}^{L}\sum_{h}\Big(\log P(\tilde{\mathcal{E}}^l_h|\check{s}^l_h, a^l_h, s^l_h, Q^\theta_{0:H}) + \log P(\check{s}^l_h|a^l_h, s^l_h, Q^\theta_{0:H}) + \log P(a^l_h|s^l_h, Q^\theta_{0:H})\Big)\\ &\leq \sum_{l=1}^{L}\sum_{h}\Big(\log P(\tilde{\mathcal{E}}^l_h|\check{s}^l_h, a^l_h, s^l_h, Q^\theta_{h:h+1}) + \log \pi^\beta_h(a^l_h|s^l_h, Q^\theta_h)\Big).\end{aligned}$$

By ignoring the log-likelihood of the transition term (akin to optimizing an upper bound on the negative loss function), we are actually being *optimistic*.

For the terms in the upper bound above, under the random perturbations assumed above, we have

$$\log P(\tilde{\mathcal{E}}^l_h|\check{s}^l_h, a^l_h, s^l_h, Q^\theta_{h:h+1}) = -\frac{1}{2}\left(r^l_h + z^l_h + \max_b Q^\theta_{h+1}(\check{s}^l_h, b) - Q^\theta_h(s^l_h, a^l_h)\right)^2 + \text{ constant}$$

and

$$\log \pi^\beta_h(a^l_h|s^l_h, Q^\theta_h) = w^l_h\left(\beta Q^\theta_h(s^l_h, a^l_h) - \log\sum_b \exp\left(\beta Q^\theta_h(s^l_h, b)\right)\right).$$

Now, denote a perturbed version of the Q^θ-parametrized online dataset,

$$\tilde{\mathcal{H}}_t(Q^\theta) = \{((s^k_h, a^k_h, \check{s}^k_h, \tilde{\mathcal{E}}^k_h)_{h=0:H-1})_{k=L+1:L+t}\},$$

and thus similar to before, we have

$$\begin{aligned}\log P(\tilde{\mathcal{H}}_t(Q^\theta)|\tilde{\mathcal{D}}_0(Q^\theta), Q^\theta_{0:H}) &= \sum_{k=L+1}^{L+t}\sum_{h}\Big(\log P(\tilde{\mathcal{E}}^k_h(Q^\theta)|\check{s}^k_h, a^k_h, s^k_h, Q^\theta_{0:H}) + \log P(\check{s}^k_h|a^k_h, s^k_h, Q^\theta)\Big),\\ &\leq \sum_{k=L+1}^{L+t}\sum_{h}\left(\log P(\tilde{\mathcal{E}}^k_h|\check{s}^k_h, a^k_h, s^k_h, Q^\theta_{h:h+1})\right),\end{aligned}$$

where we again ignored the transition term to obtain an *optimistic* upper bound.

Given the random perturbations above, we have

$$\log P(\tilde{\mathcal{E}}^k_h(Q^\theta)|\check{s}^k_h, a^k_h, s^k_h, Q^\theta_{h:h+1}) = -\frac{1}{2}\left(r^k_h + z^k_h + \max_b Q^\theta_{h+1}(\check{s}^k_h, b) - Q^\theta_h(s^k_h, a^k_h)\right)^2 + \text{ constant}.$$

The prior over β, $f_2(\beta)$ is assumed to be an exponential pdf $\lambda_2 \exp(-\lambda_2\beta), \beta \geq 0$, while that over θ is assumed Gaussian. Thus, putting it all together, we get the following ***optimistic loss function*** (to minimize over θ and β),

$$\begin{aligned}\tilde{\mathcal{L}}(\theta, \beta) &= \frac{1}{2\sigma^2}\sum_{k=1}^{L+t}\sum_{h=0}^{H-1}\left(r^k_h + z^k_h + \max_b Q^\theta_{h+1}(\check{s}^k_h, b) - Q^\theta_h(s^k_h, a^k_h)\right)^2\\ &- \sum_{l=1}^{L}\sum_{h=0}^{H-1} w^l_h\left(\beta Q^\theta_h(s^l_h, a^l_h) - \log\sum_b \exp\left(\beta Q^\theta_h(s^l_h, b)\right)\right) + \frac{1}{2}(\theta - \tilde{\theta})^\top \Sigma_0(\theta - \tilde{\theta}) + \lambda_2\beta.\end{aligned} \tag{9}$$

The above loss function is difficult to optimize in general due to the max operation, and the Q-value function in general having a nonlinear form.

Remark 4.1. Note that the loss function in equation 9 can be hard to jointly optimize over θ and β. In particular, estimates of β can be quite noisy when β is large, and the near-optimal expert policy only covers the state-action space partially. Thus, we consider other methods of estimating β that are more robust, which can then be plugged into the loss function in equation 9. Specifically, we could simply look at the entropy of the empirical distribution of the action in the offline dataset. Suppose the empirical distribution of $\{\bar{a}_0^l, \dots \bar{a}_H^l\}_{l=1}^L$ is μ_A. Then we use $c_0/\mathcal{H}(\mu_A)$ as an estimation for β, where $c_0 > 0$ is a hyperparameter. The intuition is that for smaller β, the offline actions tend to be more uniform and thus the entropy will be large. This is an unsupervised approach and agnostic to specific offline data generation process.

Remark 4.2. In the loss function in equation 9, the parameter θ appears inside the max operation. Thus, it can be quite difficult to optimize over β. Since the loss function is typically optimized via an iterative algorithm such as a gradient descent method, a simple and scalable solution that works well in practice is to use the parameter estimate θ from the previous iteration inside the max operation, and thus optimize over θ only in the other terms.

4.2 iRLSVI bridges Online RL and Imitation Learning

In the previous subsection, we derived `iRLSVI`, a Bayesian-bootstrapped algorithm. We now present interpretation of the algorithm as bridging online RL (via commonality with the RLSVI algorithm (Osband et al., 2016) and imitation learning, and hence a way for its generalization.

Consider the RLSVI algorithm for online reinforcement learning as introduced in (Osband et al., 2019). It draws its inspiration from the posterior sampling principle for online learning, and has excellent cumulative regret performance. RLSVI, that uses all of the data available at the end of episode t, including any offline dataset involves minimizing the corresponding loss function at each time step:

$$\tilde{\mathcal{L}}_{\text{RLSVI}}(\theta) = \frac{1}{2\sigma^2}\sum_{k=1}^{L+t}\sum_{h=0}^{H-1}\left(r_h^k + \max_b Q_{h+1}^\theta(\check{s}_h^k, b) - Q_h^\theta(s_h^k, a_h^k)\right)^2 + \frac{1}{2}(\theta_{0:H} - \tilde{\theta}_{0:H})^\top \Sigma_0 (\theta_{0:H} - \tilde{\theta}_{0:H}).$$

Now, let us consider an imitation learning setting. Let $\tau_l = (s_h^l, a_h^l, \check{s}_h^l)_{h=0}^{H-1}$ be the trajectory of the lth episode. Let $\hat{\pi}_h(a|s)$ denote the empirical estimate of probability of taking action a in state s at time h, i.e., an empirical estimate of the expert's randomized policy. Let $p(\tau)$ denote the probability of observing the trajectory under the policy $\hat{\pi}$.

Let $\pi_h^{\beta,\theta}(\cdot|s)$ denote the parametric representation of the policy used by the expert. And let $p^{\beta,\theta}(\tau)$ denote the probability of observing the trajectory τ under the policy $\pi^{\beta,\theta}$. Then, the loss function corresponding to the KL divergence between $\Pi_{l=1}^L p(\tau_l)$ and $\Pi_{l=1}^L p^{\beta,\theta}(\tau_l)$ is given by

$$\begin{aligned}
\tilde{\mathcal{L}}_{\text{IL}}(\beta,\theta) = D_{KL}\left(\Pi_{l=1}^L p(\tau_l) || \Pi_{l=1}^L p^{\beta,\theta}(\tau_l)\right) &= \int \Pi_{l=1}^L p(\tau_l) \log \frac{\Pi_{l=1}^L p(\tau_l)}{\Pi_{l=1}^L p^{\beta}(\tau_l)} = \sum_{l=1}^L \int p(\tau_l) \log \frac{p(\tau_l)}{p^{\beta,\theta}(\tau_l)}, \\
&= \sum_{l=1}^L \sum_{h=0}^{H-1} \log \frac{\hat{\pi}_h(a_h^l|s_h^l)}{\pi_h^{\beta,\theta}(a_h^l|s_h^l)} \\
&= \sum_{l=1}^L \sum_{h=0}^{H-1} [\log \hat{\pi}_h(a_h^l|s_h^l) - \log \pi_h^{\beta,\theta}(a_h^l|s_h^l)] \\
&= -\sum_{l=1}^L \sum_{h=0}^{H-1} \left(\beta Q_h^\theta(s_h^l, a_h^l) - \log \sum_b \exp\left(\beta Q_h^\theta(s_h^l, b)\right)\right) \quad + \text{constant}.
\end{aligned}$$

Remark 4.3. (i) The loss function $\tilde{\mathcal{L}}_{\text{IL}}(\beta,\theta)$ is the same as the second (action-likelihood) term in equation 9 while the loss function $\tilde{\mathcal{L}}_{\text{RLSVI}}(\theta)$ is the same as the first and third terms there (except for perturbation) and minus the $\lambda_2\beta$ term that corresponds to the prior over β. (ii) Note that while we used the more common KL divergence for the imitation learning loss function, use of log loss would yield the same outcome.

Thus, the `iRLSVI` loss function can be viewed as

$$\tilde{\mathcal{L}}(\beta,\theta) = \tilde{\mathcal{L}}_{\mathrm{RLSVI}}(\theta) + \tilde{\mathcal{L}}_{\mathrm{IL}}(\beta,\theta) + \lambda_2\beta, \tag{10}$$

thus establishing that the proposed algorithm may be viewed as bridging Online RL with Imitation Learning. Note that the last term corresponds to the prior over β. If β is known (or uniform), it will not show up in the loss function above.

The above also suggests a possible way to generalize and obtain other online learning algorithms that can bootstrap by use of offline datasets. Namely, at each step, they can optimize a general loss function of the following kind:

$$\tilde{\mathcal{L}}_{\alpha}(\beta,\theta) = \alpha\tilde{\mathcal{L}}_{\mathrm{ORL}}(\theta) + (1-\alpha)\tilde{\mathcal{L}}_{\mathrm{IL}}(\beta,\theta) + \lambda_2\beta, \tag{11}$$

where $\tilde{\mathcal{L}}_{\mathrm{ORL}}$ is a loss function for an Online RL algorithm, $\tilde{\mathcal{L}}_{\mathrm{IL}}$ is a loss function for some Imitation Learning algorithm, and factor $\alpha \in [0,1]$ provides a way to tune between emphasizing the offline imitation learning and the online reinforcement learning.

5 Empirical Results

Performance on the Deep Sea Environment. We now present some empirical results on "deep sea", a prototypical environment for online reinforcement learning (Osband et al., 2019). We compare three variants of the `iRLSVI` agents, which are respectively referred to as *informed* RLSVI (`iRLSVI`), *partially informed* RLSVI (`piRLSVI`), and *uninformed* RLSVI (`uRLSVI`). All three agents are tabular RLSVI agents with similar posterior sampling-type exploration schemes. However, they differ in whether or not and how to exploit the offline dataset. In particular, `uRLSVI` ignores the offline dataset; `piRLSVI` exploits the offline dataset but does not utilize the information about the generative policy; while `iRLSVI` fully exploits the information in the offline dataset, about both the generative policy and the reward feedback. We note no other algorithms are known for the problem as posed.

Deep sea is an episodic reinforcement learning problem with state space $\mathcal{S} = \{0,1,\ldots,M\}^2$ and , where M is its size. The state at period h in episode t is $s_h^t = (x_h^t, d_h^t) \in \mathcal{S}$, where $x_h^t = 0,1,\ldots,M$ is the horizontal position while $d_h^t = 0,1,\ldots,M$ is the depth (vertical position). Its action space is $\mathcal{A} = \{\mathtt{left}, \mathrm{a}\}$ and time horizon length is $H = M$. Its reward function is as follows: If the agent chooses an action a in period $h < H$, then it will receive a reward $-0.1/M$, which corresponds to a "small cost"; If the agent successfully arrives at state (M,M) in period $H = M$, then it will receive a reward 1, which corresponds to a "big bonus"; otherwise, the agent will receive reward 0. The system dynamics are as follows: for period $h < H$, the agent's depth in the next period is always increased by 1, i.e., $d_{h+1}^t = d_h^t + 1$. For the agent's horizontal position, if $a_h^t = \mathtt{left}$, then $x_{h+1}^t = \max\{x_h^t - 1, 0\}$, i.e., the agent will move left if possible. On the other hand, if $a_h^t = \mathrm{a}$, then we have $x_{h+1}^t = \min\{x_h^t + 1, M\}$ with prob. $1 - 1/M$ and $x_{h+1}^t = x_h^t$ with prob. $1/M$. The initial state of this environment is fixed at state $(0,0)$.

The offline dataset is generated based on the expert's policy specified in Eq. equation 2, and we assume $\beta(s) = \beta$ (a constant) across all states. We set the size of the offline dataset $\mathcal{D}_0$ as $|\mathcal{D}_0| = \kappa|\mathcal{A}||\mathcal{S}|$, where $\kappa \geq 0$ is referred to as *data ratio*. We fix the size of deep sea as $M = 10$. We run the experiment for $T = 300$ episodes, and the empirical cumulative regrets are averaged over 50 simulations. The experimental results are illustrated in Figure 1, as well as Figure 3 in Appendix C.

Specifically, Figure 1 plots the cumulative regret in the first $T = 300$ episodes as a function of the expert's deliberateness β, for two different data ratio $\kappa = 1, 5$. There are several interesting observations based on Figure 1: (i) Figure 1 shows that `iRLSVI` and `piRLSVI` tend to perform much better than `uRLSVI`, which demonstrates the advantages of exploiting the offline dataset, and this improvement tends to be more dramatic with a larger offline dataset. (ii) When we compare `iRLSVI` and `piRLSVI`, we note that their performance is similar when β is small, but `iRLSVI` performs much better than `piRLSVI` when β is large. This is because when β is small, the expert's generative policy does not contain much information; and as β gets larger, it contains more information and eventually it behaves like imitation learning and learns the optimal policy as $\beta \to \infty$. Note that the error bars denote the standard errors of the empirical cumulative regrets, hence the improvements are statistically significant.

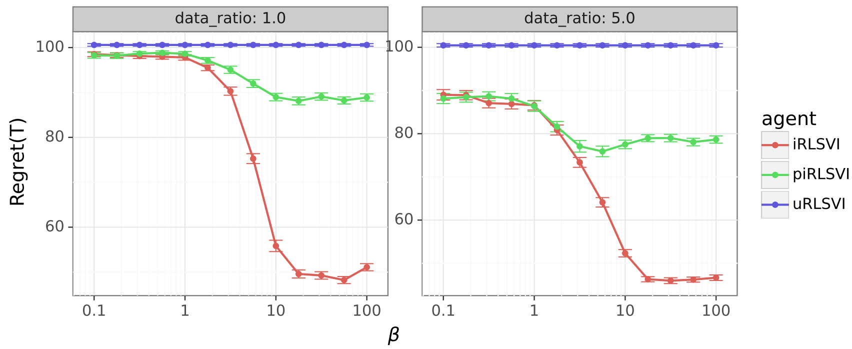

Figure 1: Cumulative regret vs. β in deep sea.

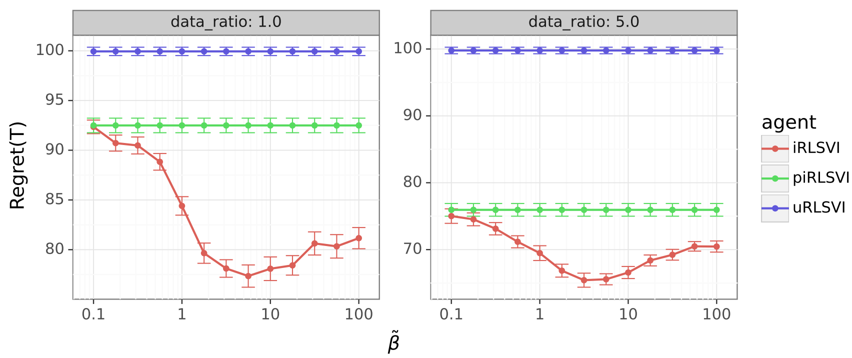

Figure 2: Robustness of `iRLSVI` to misspecification.

Robustness to misspecification of β. We also investigate the robustness of various RLSVI agents with respect to the possible misspecification of β. In particular, we demonstrate empirically that in the deep sea environment with $M = 10$, with offline dataset is generated by an expert with deliberateness $\beta = 5$, the `iRLSVI` agent is quite robust to moderate misspecification. Here, the misspecified deliberateness parameter is denoted $\tilde{\beta}$. The empirical results are illustrated in Figure 2, where the experiment is run for $T = 300$ episodes and the empirical cumulative regrets are averaged over 50 simulations.

Since `uRLSVI` and `piRLSVI` do not use parameter $\tilde{\beta}$, thus, as expected, their performance is constant over $\tilde{\beta}$. On the other hand, `iRLSVI` explicitly uses parameter $\tilde{\beta}$. As Figure 2 shows, the performance of `iRLSVI` does not vary much as long as $\tilde{\beta}$ has the same order of magnitude as β. However, there will be significant performance loss when $\tilde{\beta}$ is too small, especially when the data ratio is also small. This makes sense since when $\tilde{\beta}$ is too small, `iRLSVI` will choose to ignore all the information about the generative policy and eventually reduces to `piRLSVI`.

6 Conclusions

In this paper, we have introduced and studied a new problem: Given an offline demonstration dataset from an imperfect expert, what is the best way to leverage it to bootstrap online learning performance in MDPs. We have followed a principled approach and introduced two algorithms: the ideal `iPSRL` algorithm, and the `iRLSVI` algorithm that is computationally practical and seamlessly bridges online RL and imitation learning in a very natural way. We have shown significant reduction in regret both empirically, and theoretically as compared to two natural baselines. The dependence of the regret bound on some of the parameters (e.g., H) could be improved upon, and is a good direction for future work. In future work, we will also combine the `iRLSVI` algorithm with deep learning to leverage offline datasets effectively for continuous state and action spaces as well.

References

Alekh Agarwal and Tong Zhang. Model-based rl with optimistic posterior sampling: Structural conditions and sample complexity. *arXiv preprint arXiv:2206.07659*, 2022.

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. *arXiv preprint arXiv:2008.05556*, 2020.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. *arXiv preprint arXiv:2302.02948*, 2023.

Mark Beliaev, Andy Shih, Stefano Ermon, Dorsa Sadigh, and Ramtin Pedarsani. Imitation learning by estimating expertise of demonstrators. *Proceedings of the 39th International Conference on Machine Learning*, 162:1732–1748, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Christoph Dann, Mehryar Mohri, Tong Zhang, and Julian Zimmert. A provably efficient model-free posterior sampling method for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12040–12051, 2021.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *arXiv preprint arXiv:1901.10995*, 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.

Kuan Fang, Patrick Yin, Ashvin Nair, and Sergey Levine. Planning to practice: Efficient online fine-tuning by composing goals in latent space. *arXiv preprint arXiv:2205.08129*, 2022.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Dibya Ghosh, Anurag Ajay, Pulkit Agrawal, and Sergey Levine. Offline rl policies should be trained to be adaptive. In *International Conference on Machine Learning*, pp. 7513–7530. PMLR, 2022.

Zhaohan Daniel Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *arXiv preprint arXiv:2206.08332*, 2022.

Nicklas Hansen, Yixin Lin, Hao Su, Xiaolong Wang, Vikash Kumar, and Aravind Rajeswaran. Modem: Accelerating visual model-based reinforcement learning with demonstrations. *arXiv preprint arXiv:2212.05698*, 2022.

Botao Hao and Tor Lattimore. Regret bounds for information-directed reinforcement learning. *arXiv preprint arXiv:2206.04640*, 2022.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

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

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. When should we prefer offline reinforcement learning over behavioral cloning? *arXiv preprint arXiv:2204.05618*, 2022.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Thanh Nguyen-Tang and Raman Arora. Provably efficient neural offline reinforce-ment learning via perturbed rewards. *submitted*, 2023.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Ian Osband, Benjamin Van Roy, Daniel J Russo, Zheng Wen, et al. Deep exploration via randomized value functions. *J. Mach. Learn. Res.*, 20(124):1–62, 2019.

Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. Learning unknown markov decision processes: A thompson sampling approach. *Advances in neural information processing systems*, 30, 2017.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

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

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

Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

Julian Schrittwieser, Thomas Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, 34:27580–27591, 2021.

Yuda Song, Yifei Zhou, Ayush Sekhari, J Andrew Bagnell, Akshay Krishnamurthy, and Wen Sun. Hybrid rl: Using both offline and online data can make rl efficient. *arXiv preprint arXiv:2210.06718*, 2022.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Éric Moulines, Remi Munos, Alexey Naumov, Mark Rowland, Michal Valko, and Pierre Ménard. Optimistic posterior sampling for reinforcement learning with few samples and tight guarantees. *arXiv preprint arXiv:2209.14414*, 2022.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. *arXiv preprint arXiv:2107.06226*, 2021.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

Andrew Wagenmaker and Aldo Pacchiano. Leveraging offline data in online reinforcement learning. *arXiv preprint arXiv:2211.04974*, 2022.

Runzhe Wan, Branislav Kveton, and Rui Song. Safe exploration for efficient policy evaluation and comparison. *arXiv preprint arXiv:2202.13234*, 2022.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021a.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34: 27395–27407, 2021b.

Andrea Zanette and Rahul Sarkar. Information directed reinforcement learning. *Tech. Rep., Technical report, Technical report*, 2017.

A Proof of Lemma 3.1

Proof. To simplify notations, for a deterministic Markov policy $\pi = (\pi_h)_{h=1}^H$, $\pi_h : \mathcal{S} \mapsto \mathcal{A}$, set $\pi_h(a|s) = \mathbf{1}_{\{\pi_h(s)=a\}}$.

For each deterministic Markov policy π, define a $(SA)^H$-dimensional indicator vector $x_\pi \in \{0,1\}^{(\mathcal{S}\times\mathcal{A})^H}$ by

$$x_\pi(s_{1:H}, a_{1:H}) = \prod_{h=1}^H \pi_h(a_h|s_h). \tag{12}$$

We claim that $V_0^\pi(\theta)$, the value of policy π in an MDP θ, is a linear function of x_π. This can be seen since the probability of each trajectory $(s_{1:H}, a_{1:H})$ is given by

$$\begin{aligned}
&\Pr_{\theta,\pi}(s_{1:H}, a_{1:H}) && (13)\\
&= P_0^\theta(s_1)\pi_1(a_1|s_1)P_1^\theta(s_2|s_1,a_1)\pi_2(a_2|s_2)P_2^\theta(s_3|s_2,a_2)\times\cdots && (14)\\
&\quad\times P_{H-1}^\theta(s_H|s_{H-1},a_{H-1})\pi_H(a_H|s_H) && (15)\\
&:= P^\theta(s_{1:H},a_{1:H})x_\pi(s_{1:H},a_{1:H}), && (16)
\end{aligned}$$

and $V_0^\pi(\theta)$ is a linear function of the probability measure on the trajectory.

Let $\{\pi^1, \pi^2, \cdots, \pi^M\}$ where $M = A^{SH}$ be the set of all deterministic Markov policies. Let $\{x^1, x^2, \cdots, x^M\}$ be their corresponding $(SA)^H$-dimensional indicator vectors. If we treat x^i's as "actions", and the state-action trajectories as "observations", then the MDP learning process can be treated as a linear bandit process with bandit feedback. Applying Proposition 1 and Proposition 5 in (Russo & Van Roy, 2016) we obtain the result.

For notational simplicity define $\nu_i = \Pr(\pi^* = \pi^i)$. Let R_i denote the random total reward divided by H in MDP θ if one applies policy π^i. We have $R_i \in [0, 1]$ holds with probability 1. We have

$$\mathbb{E}[R_i] = \mathbb{E}[(v^\theta)^T x^i] = \left(\mathbb{E}[v^\theta]\right)^T x^i \tag{17}$$

$$\mathbb{E}[R_i|\pi^* = \pi^j] = \mathbb{E}[(v^\theta)^T x^i|\pi^* = \pi^j] = \left(\mathbb{E}[v^\theta|\pi^* = \pi^j]\right)^T x^i. \tag{18}$$

Set $\mu = \mathbb{E}[v^\theta]$ and $\mu^j = \mathbb{E}[v^\theta|\pi^* = \pi^j]$. Define a matrix $\Phi \in \mathbb{R}^{M\times M}$ by

$$\Phi_{ij} = \sqrt{\alpha_i\alpha_j}(\mathbb{E}[R_i|\pi^* = \pi^j] - \mathbb{E}[R_i]) \tag{19}$$

$$= \sqrt{\alpha_i\alpha_j}(\mu^j - \mu)^T x^i. \tag{20}$$

Then following the proof of (Russo & Van Roy, 2016) we have

$$\mathbb{E}[\mathrm{Reg}(T)/H] \leq \sqrt{\mathrm{rank}(\Phi)\mathbb{H}(\nu)T/2}. \tag{21}$$

It remains to bound rank(Φ). Notice that

$$\Phi = \begin{bmatrix} \sqrt{\alpha_1}(\mu^1 - \mu)^T \\ \sqrt{\alpha_2}(\mu^2 - \mu)^T \\ \vdots \\ \sqrt{\alpha_M}(\mu^M - \mu)^T \end{bmatrix} \begin{bmatrix} \sqrt{\alpha_1}x^1 & \sqrt{\alpha_2}x^1 & \cdots & \sqrt{\alpha_M}x^M. \end{bmatrix} \tag{22}$$

which means that Φ is the product of a $M \times (SA)^H$ matrix and a $(SA)^H \times M$ matrix. This means that rank$(M) \leq (SA)^H$.

On the other hand, we could follow the proof of Theorem 4.11 in (Hao & Lattimore, 2022) to obtain the second part of the regret upper bound.

□

B Proof of Lemma 3.7

Proof. For convenience, write $\Pr_\theta(\cdot) = \Pr(\cdot|\theta)$.

Define the event $\mathcal{E}_{n,h} = \{\bar{a}_{n,h} \neq a_h^*(\bar{s}_{n,h}; \theta)\}$, i.e. in the n-th round of demonstration, the expert did not take the optimal action at time h. Given the expert's randomized policy ϕ, we have

$$\Pr_\theta(\mathcal{E}_{n,h}|\bar{s}_{n,1:h}, \bar{a}_{n,1:h-1}) \tag{23}$$

$$= 1 - \frac{1}{1 + \sum_{a\neq a_h^*(\bar{s}_{n,h};\theta)} \exp(-\beta\Delta_h(\bar{s}_{n,h}, a; \theta))} \tag{24}$$

$$\leq \sum_{a\neq a_h^*(\bar{s}_{n,h};\theta)} \exp(-\beta\Delta_h(\bar{s}_{n,h}, a; \theta)) \tag{25}$$

$$\leq (A - 1)\exp(-\beta\Delta) =: \tilde{\kappa}_\beta. \tag{26}$$

Define $\kappa_\beta = (H - 1)\tilde{\kappa}_\beta$. Then $\beta \geq \underline{\beta}$ means that $\kappa_\beta \leq \underline{p}/3$.

Consider each $(h, s) \in [H] \times \mathcal{S}$, conditioning on θ there are two cases:

- If $p_h(s; \theta) > 0$ then

$$\Pr_\theta(\bar{s}_{n,h} = s) \geq (1 - \tilde{\kappa}_\beta)^{h-1} p_h(s; \theta) \geq (1 - \kappa_\beta)\underline{p} \geq \frac{2}{3}\underline{p}. \tag{27}$$

The first inequality in equation 27 can be established via induction on h: First observe that $\Pr_\theta(\bar{s}_{n,1} = s) = p_1(s;\theta)$ for all $s \in \mathcal{S}$ by definition. Suppose that we have proved the statement for time h, i.e. $\Pr_\theta(\bar{s}_{n,h} = s) \geq (1-\tilde{\kappa}_\beta)^{h-1} p_h(s;\theta)$ for all $s \in \mathcal{S}$. Then we have

$$p_{h+1}(s';\theta) = \sum_{s\in\mathcal{S}} \Pr_\theta(s'|s, a_h^*(s;\theta))p_h(s;\theta) \tag{28}$$

$$\Pr_\theta(\bar{s}_{n,h+1} = s') \geq \sum_{s\in\mathcal{S}} \phi_h^\beta(a_h^*(s;\theta)|s;\theta) \Pr_\theta(s'|s, a_h^*(s;\theta)) \Pr_\theta(\bar{s}_{n,h} = s) \tag{29}$$

$$\geq \sum_{s\in\mathcal{S}} (1-\tilde{\kappa}_\beta) \Pr_\theta(s'|s, a_h^*(s;\theta)) \Pr_\theta(\bar{s}_{n,h} = s) \tag{30}$$

$$\geq \sum_{s\in\mathcal{S}} (1-\tilde{\kappa}_\beta)^h \Pr_\theta(s'|s, a_h^*(s;\theta))p_h(s;\theta). \tag{31}$$

The statement for $h+1$ then follows by comparing equation 28 and equation 31, establishing the induction step.

- If $p_h(s;\theta) = 0$, then if $\bar{s}_{n,h} = s$, the expert must have chosen some action that was not optimal before time h in the n-th round of demonstration. We conclude that

$$\Pr_\theta(\bar{s}_{n,h} = s) \leq \Pr_\theta\left(\bigcup_{\tilde{h}=1}^{h-1} \mathcal{E}_{n,\tilde{h}}\right) \leq \sum_{\tilde{h}=1}^{h-1} \Pr_\theta(\mathcal{E}_{n,\tilde{h}}) \leq \kappa_\beta \leq \frac{1}{3}\underline{p}. \tag{32}$$

The above argument shows that there's a separation of probability between two types of state and time index pairs under the expert's policy $\phi^\beta(\theta)$: the ones that are probable under the optimal policy $\pi^*(\theta)$ and the ones that are not. Using this separation, we will proceed to show that when N is large, we can distinguish the two types of state and time index pairs through their appearance counts in $\mathcal{D}_0$. This will allow us to construct a good estimator of π^*.

Define $\hat{\pi}^*$ to be the estimator of π^* constructed with $\delta = \underline{p}/2$. If $\hat{\pi}^* \neq \pi$, then either one of the following cases happens

- There exists an $(s,h) \in \mathcal{S}\times[H]$ pair such that $p_h(s;\theta) > 0$ but $N_h(s) < \delta N$;
- There exists an $(s,h) \in \mathcal{S}\times[H]$ pair such that $p_h(s;\theta) = 0$ but $N_h(s) \geq \delta N$;
- There exists an $(s,h) \in \mathcal{S}\times[H]$ pair such that $p_h(s;\theta) > 0$ and $N_h(s) \geq \delta N$, but $\pi_h^*(s;\theta) = a_h^*(s;\theta) \neq \arg\max_a N_h(s,a) = \hat{\pi}_h^*(s)$;

Using union bound, we have

$$\Pr_\theta(\pi^* \neq \hat{\pi}^*) \tag{33}$$

$$\leq \sum_{(s,h):p_h(s;\theta)>0} \Pr_\theta(N_h(s) < \delta N) + \sum_{(s,h):p_h(s;\theta)=0} \Pr_\theta(N_h(s) \geq \delta N) \tag{34}$$

$$+ \sum_{(s,h):p_h(s;\theta)>0} \Pr_\theta(N_h(s) \geq \delta N, a_h^*(s;\theta) \neq \arg\max_a N_h(s,a)). \tag{35}$$

Let $\mathrm{Bin}(M,q)$ denote a binomial random variable with parameters $M \in \mathbb{N}$ and $q \in [0,1]$. Notice that conditioning on θ, each $N_h(s)$ is a binomial random variable with parameters N and $\tilde{p}_{\theta,h}(s) := \Pr_\theta(\bar{s}_{1,h} = s)$.

Using equation 27 and Lemma 3.6, we conclude that each term in the first summation of equation 35 satisfies

$$\Pr_\theta(N_h(s) < \delta N) \leq \Pr(\mathrm{Bin}(N, 2\underline{p}/3) < (\underline{p}/2)N) \tag{36}$$

$$\leq \exp(-2N(\underline{p}/6)^2) = \exp\left(-\frac{N\underline{p}^2}{18}\right). \tag{37}$$

Using equation 32 and Lemma 3.6, we conclude that each term in the second summation of equation 35 satisfies

$$\Pr_{\theta}(N_h(s) \geq \delta N) \leq \Pr(\text{Bin}(N, \kappa_\beta) \leq (\underline{p}/2)N) \tag{38}$$

$$\leq \exp\left(-2N\left(\frac{\underline{p}}{2} - \kappa_\beta\right)^2\right) \leq \exp\left(-\frac{N\underline{p}^2}{18}\right). \tag{39}$$

Again, using Lemma 3.6, each term in the third summation of equation 35 satisfies

$$\Pr_{\theta}(N_h(s) \geq \delta N, a_h^*(s;\theta) \neq \arg\max_a N_h(s,a)) \tag{40}$$

$$\leq \Pr_{\theta}(a_h^*(s;\theta) \neq \arg\max_a N_h(s,a) \mid N_h(s) \geq \delta N) \tag{41}$$

$$\leq \Pr_{\theta}(N_h(s) - N_h(s, a_h^*(s;\theta)) \geq N_h(s)/2 \mid N_h(s) \geq \delta N) \tag{42}$$

$$\leq \Pr_{\theta}(\text{Bin}(N_h(s), \tilde{\kappa}_\beta) \geq N_h(s)/2 \mid N_h(s) \geq \delta N) \tag{43}$$

$$\leq \Pr_{\theta}(\text{Bin}(N_h(s), 1/3) \geq N_h(s)/2 \mid N_h(s) \geq \delta N) \tag{44}$$

$$\leq \mathbb{E}_\theta\left[\exp\left(-\frac{N_h(s)}{18}\right) \;\middle|\; N_h(s) \geq \delta N\right] \tag{45}$$

$$\leq \exp\left(-\frac{\delta N}{18}\right) = \exp\left(-\frac{N\underline{p}}{36}\right). \tag{46}$$

Combining the above we obtain

$$\Pr_{\theta}(\pi^* \notin \hat{\pi}^*) \leq SH\left[\exp\left(-\frac{N\underline{p}^2}{18}\right) + \exp\left(-\frac{N\underline{p}}{36}\right)\right]. \tag{47}$$

□

C More empirical results on deep sea

In this appendix, we provide more empirical results for the deep sea experiment described in Section 5. Specifically, for deep sea with size $M = 10$, data ratio $\kappa = 1, 5$, and expert's deliberateness $\beta = 1, 10$, we plot the cumulative regret of `iRLSVI`, `piRLSVI`, and `uRLSVI` as a function of the number of episodes t for the first $T = 300$ episodes. The experiment results are averaged over 50 simulations and are illustrated in Figure 3.

As we have discussed in the main body of the paper, when both data ratio κ and the expert's deliberateness β are small, then there are not many offline data and the expert's generative policy is also not very informative. In this case, `iRLSVI`, `piRLSVI`, and `uRLSVI` perform similarly. On the other hand, when the data ratio κ is large, `iRLSVI` and `piRLSVI` tend to perform much better than `uRLSVI`, which does not use the offline dataset. Similarly, when the expert's deliberateness β is large, then the expert's generative policy is informative. In this case, `iRLSVI` performs much better than `piRLSVI` and `uRLSVI`.

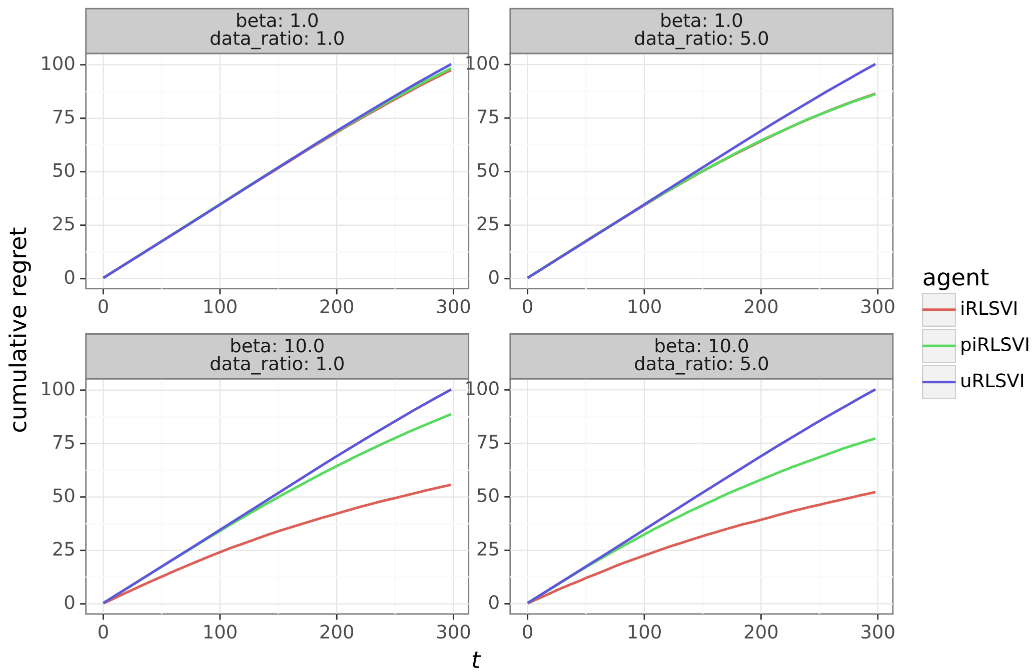

Figure 3: Cumulative regret vs. number of episodes in deep sea.