Reinforcement Learning with Trajectory Feedback

Yonathan Efroni[*1,2], **Nadav Merlis**[*1] **and Shie Mannor**[1,3]
[1]Technion, Israel Institute of Technology
[2]Microsoft Research, New York
[3]Nvidia Research, Israel

Abstract

The standard feedback model of reinforcement learning requires revealing the reward of every visited state-action pair. However, in practice, it is often the case that such frequent feedback is not available. In this work, we take a first step towards relaxing this assumption and require a weaker form of feedback, which we refer to as trajectory feedback. Instead of observing the reward obtained after every action, we assume we only receive a score that represents the quality of the whole trajectory observed by the agent, namely, the sum of all rewards obtained over this trajectory. We extend reinforcement learning algorithms to this setting, based on least-squares estimation of the unknown reward, for both the known and unknown transition model cases, and study the performance of these algorithms by analyzing their regret. For cases where the transition model is unknown, we offer a hybrid optimistic-Thompson Sampling approach that results in a tractable algorithm.

1 Introduction

The field of Reinforcement Learning (RL) tackles the problem of learning how to act optimally in an unknown dynamical environment. Recently, RL witnessed remarkable empirical success (e.g., Mnih et al. 2015; Levine et al. 2016; Silver et al. 2017). However, there are still some matters that hinder its use in practice. One of them, we claim, is the type of feedback an RL agent is assumed to observe. Specifically, in the standard RL formulation, an agent acts in an unknown environment and receives feedback on its actions in the form of a state-action dependent reward signal. Although such an interaction model seems undemanding at first sight, in many interesting problems, such reward feedback cannot be realized. In practice, and specifically in non-simulated environments, it is hardly ever the case that an agent can query a state-action reward function from every visited state-action pair since such a query can be very costly. For example. consider the following problems:

(i) Consider the challenge of autonomous car driving. Would we want to deploy an RL algorithm for this setting, we would need a reward signal from every visited state-action pair. Obtaining such data is expected to be very costly since it requires scoring each *state-action pair with a real number. For example, if a human is involved in providing the feedback, (a) he or she might refuse to supply with such feedback due to the Sisyphean nature of this task, or (b) supplying with such feedback might take too much time for the needs of the algorithm designer.*

(ii) Consider a multi-stage UX interface that we want to optimize using an RL algorithm. To do so, in the standard RL setting, we would need a score for every visited state-action pair. However, as we ask the users for more information on the quality of different state-action pairs, the users' opinions might change due to the information they need to supply. For example, as we ask for more information, the user might be prone to be more negative about the quality of the interface as the user becomes less patient to provide the requested feedback. Thus, we would like to keep the number of queries from the user to be minimal.

Rather than circumventing this problem by deploying heuristics (e.g., by hand-engineering a reward signal), in this work, we relax the feedback mechanism to a weaker and more practical one. We then study RL algorithms in the presence of this weaker form of feedback mechanism, a setting which we refer to as *RL with trajectory feedback*. In RL with trajectory feedback, the agent does not have access to a per state-action reward function. Instead, it receives the sum of rewards on the performed trajectory as well as the identity of visited state-action pairs in the trajectory. E.g., for autonomous car driving, we only require feedback on the score of a trajectory, instead of the score of each individual state-action pair. Indeed, this form of feedback is much weaker than the standard RL feedback and is expected to be more common in practical scenarios.

We start by defining our setting and specifying the interaction model of RL with trajectory feedback (Section 2). In Section 3, we introduce a natural least-squares estimator with which the true reward function can be learned based on the trajectory feedback. Building on the least-squares estimator, we study algorithms that explicitly trade-off exploration and exploitation. We start by considering the case where the model is known while the reward function needs to be learned. By generalizing the analysis of standard linear bandit algorithms (OFUL (Abbasi-Yadkori, Pál, and Szepesvári 2011) and Thompson-Sampling (TS) for lin-

[*]Equal contribution, alphabetical order

ear bandits (Agrawal and Goyal 2013)), we establish performance guarantees for this setup in sections 4 and 5.1. Although the OFUL-based algorithm gives better performance than the TS-based algorithm, its update rule is computationally intractable, as it requires solving a convex maximization problem. Thus, in Section 5.2 we generalize the TS-based algorithm to the case where both the reward and the transition model are unknown. To this end, we learn the reward by a TS approach and learn the transition model by an optimistic approach. The combination of the two approaches yields a computationally tractable algorithm, which requires solving an empirical Markov Decision Process (MDP) in each round. For all algorithms, we establish regret guarantees that scale as $\sqrt{K}$, where K is the number of episodes. Finally, in Section 5.3, we identify the most computationally demanding stage in the algorithm and suggest a variant to the algorithm that rarely performs this stage. Notably, we show that the effect of this modification on the regret is minor. A summary of our results can be found in Table 1.

2 Notations and Definitions

We consider finite-horizon MDPs with time-independent dynamics. A finite-horizon MDP is defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, P, H)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces with cardinalities S and A, respectively. The immediate reward for taking an action a at state s is a random variable $R(s,a) \in [0,1]$ with expectation $\mathbb{E}[R(s,a)] = r(s,a)$. The transition kernel is $P(s' \mid s,a)$, the probability of transitioning to state s' upon taking action a at state s. $H \in \mathbb{N}$ is the *horizon*, i.e., the number of time-steps in each episode, and $K \in \mathbb{N}$ is the total number of episodes. We define $[N] \stackrel{\text{def}}{=} \{1, \ldots, N\},$ for all $N \in \mathbb{N}$, and use $h \in [H]$ and $k \in [K]$ to denote time-step inside an episode and the index of an episode, respectively. We also denote the initial state in episode $k \in [K]$ by s_1^k, which can be arbitrarily chosen.

A deterministic policy $\pi : \mathcal{S} \times [H] \to \mathcal{A}$ is a mapping from states and time-step indices to actions. We denote by $a_h \stackrel{\text{def}}{=} \pi(s_h, h)$, the action taken at time h at state s_h according to a policy π. The quality of a policy π from state s at time h is measured by its value function, which is defined as

$$V_h^\pi(s) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{h'=h}^{H} r(s_{h'}, \pi(s_{h'}, h')) \mid s_h = s, \pi\right],$$

where the expectation is over the environment randomness. An optimal policy maximizes this value for all states s and time-steps h simultaneously, and the corresponding optimal value is denoted by $V_h^*(s) \stackrel{\text{def}}{=} \max_\pi V_h^\pi(s),$ for all $h \in [H]$. We can also reformulate the optimization problem using the *occupancy measure* (e.g., Puterman 1994; Altman 1999). The occupancy measure of a policy π is defined as the distribution over state-action pairs generated by executing the policy π in the finite-horizon MDP $\mathcal{M}$ with a transition kernel p (e.g., Zimin and Neu 2013):

$$\begin{aligned} q_h^\pi(s,a;p) &\stackrel{\text{def}}{=} \mathbb{E}[1(s_h = s, a_h = a) \mid s_1 = s_1, p, \pi] \\ &= \Pr\{s_h = s, a_h = a \mid s_1 = s_1, p, \pi\} \end{aligned}$$

For brevity, we define the matrix notation $q^\pi(p) \in \mathbb{R}^{HSA}$ where its (s,a,h) element is given by $q_h^\pi(s,a;p)$. Furthermore, let the average occupancy measure be $d_\pi(p) \in \mathbb{R}^{SA}$ such that $d_\pi(s,a;p) \stackrel{\text{def}}{=} \sum_{h=1}^H q_h^\pi(s,a;p)$. For ease of notation, when working with the transition kernel of the true model $p = P$, we write $q^\pi = q^\pi(P)$ and $d_\pi = d_\pi(P)$.

This definition implies the following relation:

$$\begin{aligned} V_1^\pi(s_1;p,r) &= \sum_{h\in[H]} \left(\sum_{s_h,a_h} r(s_h,a_h) q_h^\pi(s_h,a_h;p) \right) \\ &= \sum_{s,a} d_\pi(s,a;p) r(s,a) = d_\pi(p)^T r, \end{aligned} \quad (1)$$

where $V_1^\pi(s_1;p,r)$ is the value of an MDP whose reward function is r and its transition kernel is p.

Interaction Model of Reinforcement Learning with Trajectory Feedback. We now define the interaction model of RL agents that receive trajectory feedback, the model that we analyze in this work. We consider an agent that repeatedly interacts with an MDP in a sequence of episodes $[K]$. The performance of the agent is measured by its *regret*, defined as $\mathrm{R}(K) \stackrel{\text{def}}{=} \sum_{k=1}^K \big(V_1^*(s_1^k) - V_1^{\pi_k}(s_1^k)\big)$. We denote by s_h^k and a_h^k for the state and the action taken at the h^{th} time-step of the k^{th} episode. At the end of each episode $k \in [K]$, the agent only observes the cumulative reward experienced while following its policy π_k and the identity of the visited state-action pairs, i.e.,

$$\hat{V}_k(s_1^k) = \sum_{h=1}^H R(s_h^k, a_h^k), \text{ and, } \left\{(s_h^k, a_h^k)\right\}_{h=1}^{H+1}. \quad (2)$$

This comes in contrast to the standard RL setting, in which the agent observes the reward per visited state-action pair, $\left\{R(s_h^k, a_h^k)\right\}_{h=1}^H$. Thus, RL with trajectory feedback receives more obscured feedback from the environment on the quality of its actions. Obviously, standard RL feedback allows calculating $\hat{V}_k(s_1^k)$, but one cannot generally reconstruct $\left\{R(s_h^k, a_h^k)\right\}_{h=1}^H$ by accessing only $\hat{V}_k(s_1^k)$.

Next, we define the filtration F_k that includes all events (states, actions, and rewards) until the end of the k^{th} episode, as well as the initial state of the episode $k+1$. We denote by $T = KH$, the total number of time-steps (samples). Moreover, we denote by $n_k(s,a)$, the number of times that the agent has visited a state-action pair (s,a), and by $\hat{X}_k$, the empirical average of a random variable X. Both quantities are based on experience gathered until the end of the k^{th} episode and are F_k measurable.

Notations. We use $\tilde{\mathcal{O}}(X)$ to refer to a quantity that is upper bounded by X, up to poly-log factors of S, A, T, K, H, and $\frac{1}{\delta}$. Furthermore, the notation $\mathcal{O}(X)$ refers to a quantity that is upper bounded by X up to constant multiplicative factors. We use $X \vee Y \stackrel{\text{def}}{=} \max\{X, Y\}$ and denote I_m as the identity matrix in dimension m. Similarly, we denote by

Result	Exploration	Model Learning	Time Complexity	Regret
Theorem 3	OFUL	X	Computationally-Hard	$\tilde{\mathcal{O}}\left(SAH\sqrt{K}\right)$
Theorem 4	TS	X	$\mathcal{O}\left((SA)^3 + S^2A(H+A)\right)$	$\tilde{\mathcal{O}}\left((SA)^{3/2}H\sqrt{K}\right)$
Theorem 5	TS	V	$\mathcal{O}\left((SA)^3 + S^2A(H+A)\right)$	$\tilde{\mathcal{O}}\left(S^2A^{3/2}H^{3/2}\sqrt{K}\right)$
Theorem 7	TS	V	$\mathcal{O}\left(S^2A(H+A) + \frac{(SA)^4}{\log(1+C)}\frac{\log\frac{KH}{SA}}{K}\right)$	$\tilde{\mathcal{O}}\left((SA)^{3/2}H\sqrt{K}\left(\sqrt{SH}+\sqrt{C}\right)\right)$

Table 1: S and A are the state and action sizes, respectively, and H is the horizon. K is the number of episode and $C > 0$ is a parameter of Algorithm 4. Exploration - whether the reward exploration is optimistic ('OFUL') or uses posterior sampling ('TS'). Model Learning - whether the algorithm knows the model (X) or has to learn it (V). Time complexity - per-episode average time complexity. The hardness of the optimistic algorithm is explained at the end of Section 4 and the time complexity of the TS-based algorithm is explained in Section 5.3. Regret bounds ignore log-factors and constants and assume that $SA \geq H$.

$\mathbf{0}_m \in \mathbb{R}^m$, the vector whose components are zeros. Finally, for any positive definite matrix $M \in \mathbb{R}^{m\times m}$ and any vector $x \in \mathbb{R}^m$, we define $\|x\|_M = \sqrt{x^T M x}$.

3 From Trajectory Feedback to Least-Squares Estimation

In this section, we examine an intuitive way for estimating the true reward function r, given only the cumulative rewards on each of the past trajectories and the identities of visited state-actions. Specifically, we estimate r via a Least-Squares (LS) estimation. Consider past data in the form of (2). To make the connection of the trajectory feedback to LS estimation more apparent, let us rewrite (2) as follows,

$$\hat{V}_k(s_1^k) = \hat{q}_k^T R, \tag{3}$$

where $\hat{q}_k \in \mathbb{R}^{SAH}$ is the *empirical state-action visitation* vector given by $\hat{q}_k(s,a,h) = 1\big(s = s_h^k, a = a_h^k\big) \in [0,1]$, and $R \in \mathbb{R}^{SAH}$ is the noisy version of the true reward function, namely $R(s,a,h) = R(s_h, a_h)$. Indeed, since the identity of visited state-action pairs is given to us, we can compute $\hat{q}_k$ using our data. Furthermore, observe that

$$\begin{aligned}\mathbb{E}\left[\hat{q}_k^T R | \hat{q}_k\right] &= \sum_{s,a,h} \hat{q}_k(s,a,h) r(s,a) \\ &= \sum_{s,a}\left(\sum_{h=1}^{H} \hat{q}_k(s,a,h)\right) r(s,a) \overset{\text{def}}{=} \hat{d}_k^T r,\end{aligned}$$

where the first equality holds since we assume the rewards are i.i.d. and drawn at the beginning of each episode. In the last inequality we defined the *empirical state-action frequency* vector $\hat{d}_k \in \mathbb{R}^{SA}$ where $\hat{d}_k(s,a) = \sum_{h=1}^{H} \hat{q}_k(s,a,h)$, or, alternatively,

$$\hat{d}_k(s,a) = \sum_{h=1}^{H} 1\big(s = s_h^k, a = a_h^k\big) \in [0,H].$$

This observation enables us to think of our data as noisy samples of $\hat{d}_k^T r$, from which it is natural to estimate the reward r by a (regularized) LS estimator, i.e., for some $\lambda > 0$,

$$\hat{r}_k \in \arg\min_r \left(\sum_{l=1}^{k} (\langle \hat{d}_l, r\rangle - \hat{V}_l)^2 + \lambda I_{SA}\right),$$

This estimator is also given by the closed form solution

$$\hat{r}_k = (D_k^T D_k + \lambda I_{SA})^{-1} Y_k \overset{\text{def}}{=} A_k^{-1} Y_k, \tag{4}$$

where $D_k \in \mathbb{R}^{k\times SA}$ is a matrix with $\left\{\hat{d}_k^T\right\}$ in its rows, $Y_k = \sum_{s=1}^{k} \hat{d}_s \hat{V}_s \in \mathbb{R}^{SA}$ and $A_k = D_k^T D_k + \lambda I_{SA} \in \mathbb{R}^{SA\times SA}$.

A needed property of the estimator $\hat{r}_k$ is for it to be 'concentrated' around the true reward r. By properly defining the filtration and observing that $\hat{V}_k$ is $\sqrt{H/4}$ sub-Gaussian given $\hat{d}_k$ (as a sum of H independent variables in $[0,1]$), it is easy to establish a uniform concentration bound via Theorem 2 of (Abbasi-Yadkori, Pál, and Szepesvári 2011) (for completeness, we provide the proof in Appendix I).

Proposition 1 (Concentration of Reward). *Let $\lambda > 0$ and $A_k \overset{\text{def}}{=} D_k^T D_k + \lambda I_{SA}$. For any $\delta \in (0,1)$, with probability greater than $1 - \delta/10$ uniformly for all $k \geq 0$, it holds that* $\|r - \hat{r}_k\|_{A_k} \leq \sqrt{\frac{1}{4} SAH \log\left(\frac{1+kH^2/\lambda}{\delta/10}\right)} + \sqrt{\lambda SA} \overset{\text{def}}{=} l_k$.

Relation to Linear Bandits. Assume that the transition kernel P is known while the reward r is unknown. Also, recall that the set of the average occupancy measures, denoted by $\mathcal{K}(P)$, is a convex set (Altman 1999). Then, Equation (1) establishes that RL with trajectory feedback can be understood as an instance of linear-bandits over a convex set. I.e., it is equivalent to the problem of minimizing the regret $\mathrm{R}(K) = \sum_k \max_{d\in\mathcal{K}(P)} d^T r - d_{\pi_k}^T r$, where the feedback is a noisy version of $d_{\pi_k}^T r$ since $\mathbb{E}\left[\hat{V}_k \mid F_{k-1}\right] = d_{\pi_k}^T r$. Under this formulation, choosing a policy π_k is equivalent to choosing an 'action' from the convex set $d_{\pi_k} \in \mathcal{K}(P)$.

However, we make use of $\hat{d}_k$, and not the actual 'action' that was taken, d_{π_k}. Importantly, this view allows us to generalize algorithms to the case where the transition model P is unknown (as we do in Section 5). When the transition model is not known and estimated via $\bar{P}$, there is an *error in identifying the action*, in the view of linear bandits, since $d_\pi \neq d_\pi(\bar{P})$. This error, had we used the 'naïve' linear bandit approach of choosing contexts from $\mathcal{K}(\bar{P})$, would result in errors in the matrix A_k. Since our estimator uses the empirical average state-action frequency, $\hat{d}_k$, the fact the model is unknown does not distort the reward estimation.

Algorithm 1 OFUL for RL with Trajectory Feedback and Known Model

Require: $\delta \in (0,1)$, $\lambda = H$,
$$l_k = \sqrt{\tfrac{1}{4} SAH \log\Big(\tfrac{1+kH^2/\lambda}{\delta/10}\Big)} + \sqrt{\lambda SA}$$
Initialize: $A_0 = \lambda I_{SA}$, $Y_0 = \mathbf{0}_{SA}$
for $k = 1, ..., K$ **do**
  Calculate $\hat{r}_{k-1}$ via LS estimation (4)
  Solve $\pi_k \in \arg\max_\pi \Big(d_\pi^T \hat{r}_{k-1} + l_{k-1} \|d_\pi\|_{A_{k-1}^{-1}} \Big)$
  Play π_k, observe $\hat{V}_k$ and $\big\{(s_h^k, a_h^k)\big\}_{h=1}^H$
  Update $A_k = A_{k-1} + \hat{d}_k \hat{d}_k^T$ and $Y_k = Y_{k-1} + \hat{d}_k \hat{V}_k$.
end for

Policy Gradient Approach for RL with Trajectory Feedback. Another natural algorithmic approach to the setting of RL with trajectory feedback is to use policy search. That is, instead of estimating the reward function via least-square and follow a model-based approach, one can directly optimize over the policy class. By the log-derivative trick (as in the REINFORCE algorithm (Williams 1992)):

$$\nabla_\pi V^\pi(s) = \mathbb{E}\left[\left(\sum_{h=1}^H \nabla_\pi \log \pi(a_h|s_h) \right) \left(\sum_{h=1}^H r(s_h, a_h) \right) | s_1 = s, \pi \right].$$

Thus, if we are supplied with the cumulative reward of a trajectory we can estimate the derivative $\nabla_\pi V^\pi(s)$ and use stochastic gradient ascent algorithm. However, this approach fails in cases the exploration is challenging (Agarwal et al. 2020), i.e., the sample complexity can increase exponentially with H. We conjecture that by combining exploration bonus the REINFORCE algorithm can provably perform well, with polynomial sample complexity. We leave such an extension as an interesting future research direction.

4 OFUL for RL with Trajectory Feedback and Known Model

Given the concentration of the estimated reward in Proposition 1, it is natural to follow the optimism in the face of uncertainty approach, as used in the OFUL algorithm (Abbasi-Yadkori, Pál, and Szepesvári 2011) for linear bandits. We adapt this approach to RL with trajectory feedback, as depicted in Algorithm 1; on each episode, we find a policy that maximizes the estimated value $V^\pi(s_1; P, \hat{r}_{k-1}) = d_\pi^T \hat{r}_{k-1}$, and an additional 'confidence' term $l_{k-1}\|d_\pi\|_{A_{k-1}^{-1}}$ that properly encourages the policy π_k to be exploratory.

The analysis of OFUL is based upon two key ingredients, (i) a concentration result, and (ii) an elliptical potential lemma. For the setting of RL with trajectory feedback, the concentration result can be established with similar tools to Abbasi-Yadkori, Pál, and Szepesvári (see Proposition 1). However, (ii), the elliptical potential lemma, should be re-derived. The usual elliptical potential lemma (Abbasi-Yadkori, Pál, and Szepesvári 2011) states that for $x_k \in \mathbb{R}^m$,

$$\sum_{k=1}^K \|x_k\|_{A_{k-1}} \le \tilde{\mathcal{O}}\Big(\|x\| \sqrt{mK/\lambda} \Big),$$

where $A_k = A_{k-1} + x_k x_k^T$, $A_0 = \lambda I_m$ and $\|x_k\| \le \|x\|$. However, for RL with trajectory feedback, the term we wish to bound is $\sum_{k=1}^K \|d_{\pi_k}\|_{A_{k-1}}$, where $A_k = A_{k-1} + \hat{d}_k \hat{d}_k^T$, $A_0 = \lambda I_{SA}$. Thus, since $d_{\pi_k} \neq \hat{d}_k$, we cannot apply the usual elliptical potential lemma. Luckily, it is possible to derive a variation of the lemma, suited for our needs, by recognizing that $d_{\pi_k} = \mathbb{E}\Big[\hat{d}_k | F_{k-1}\Big]$ where the equality holds component-wise. Based on this observation, the next lemma – central to the analysis of all algorithms in this work – can be established (in Appendix B we prove a slightly more general statement that will be useful in next sections as well).

Lemma 2 (Expected Elliptical Potential Lemma)**.** *Let $\lambda > 0$. Then, uniformly for all $K > 0$, with probability greater than $1 - \delta$, it holds that*

$$\sum_{k=0}^K \|d_{\pi_k}\|_{A_{k-1}^{-1}} \le \mathcal{O}\left(\sqrt{\frac{H^2}{\lambda} KSA \log\Big(\lambda + \frac{KH^2}{SA} \Big)} \right).$$

Proof Sketch. Applying Jensen's inequality (using the convexity of norms), we get

$$\|d_{\pi_k}\|_{A_{k-1}^{-1}} = \Big\| \mathbb{E}\Big[\hat{d}_k | F_{k-1}\Big] \Big\|_{A_{k-1}^{-1}} \le \mathbb{E}\Big[\|\hat{d}_k\|_{A_{k-1}^{-1}} | F_{k-1} \Big]$$

Therefore, we can write

$$\sum_{k=0}^K \|d_{\pi_k}\|_{A_{k-1}^{-1}} \le \sum_{k=1}^K \mathbb{E}\Big[\|\hat{d}_k\|_{A_{k-1}^{-1}} | | F_{k-1} \Big]$$
$$= \underbrace{\sum_{k=1}^K \Big(\mathbb{E}\Big[\|\hat{d}_k\|_{A_{k-1}^{-1}} | | F_{k-1} \Big] - \|\hat{d}_k\|_{A_{k-1}^{-1}} \Big)}_{(a)} + \underbrace{\sum_{k=1}^K \|\hat{d}_k\|_{A_{k-1}^{-1}}}_{(b)},$$

where in the last relation, we added and subtracted the random variable $\|\hat{d}_k\|_{A_{k-1}^{-1}}$. It is evident that (a) is a bounded martingale difference sequence and, thus, can be bounded with probability $1 - \delta/2$ by

$$(a) \le 4 \sqrt{\frac{H^2}{\lambda} K \log\Big(\frac{2K}{\delta} \Big)}.$$

Term (b) can be bounded by applying the usual elliptical potential (Abbasi-Yadkori, Pál, and Szepesvári 2011)) by

$$(b) \le \sqrt{\frac{H^2}{\lambda}} \sqrt{2KSA \log\Big(\lambda + \frac{KH^2}{SA} \Big)}.$$

Combining the bounds on (a), (b) concludes the proof. □

Based on the concentration of the estimated reward $\hat{r}_k$ around the true reward r (Proposition 1) and the expected elliptical potential lemma (Lemma 2), the following performance guarantee of Algorithm 1 is established (see Appendix C for the full proof).

Theorem 3 (OFUL for RL with Trajectory Feedback and Known Model)**.** *For any $\delta \in (0,1)$, it holds with probability greater than $1 - \delta$ that for all $K > 0$,*

$$\mathrm{R}(K) \le \mathcal{O}\left(SAH\sqrt{K} \log\Big(\frac{KH}{\delta} \Big) \right).$$

Algorithm 2 TS for RL with Trajectory Feedback and Known Model

Require: $\delta \in (0,1)$, $\lambda = H$, $v_k = \sqrt{9SAH \log \frac{kH^2}{\delta/10}}$

$$l_k = \sqrt{\tfrac{1}{4} SAH \log\Big(\tfrac{1+kH^2/\lambda}{\delta/10}\Big)} + \sqrt{\lambda SA}$$

Initialize: $A_0 = \lambda I_{SA}$, $Y_0 = \mathbf{0}_{SA}$

for $k = 1, ..., K$ **do**

Calculate $\hat{r}_{k-1}$ via LS estimation (4)

Draw noise $\xi_k \sim \mathcal{N}(0, v_k^2 A_{k-1}^{-1})$ and define $\tilde{r}_k = \hat{r}_{k-1} + \xi_k$

Solve an MDP with perturbed empirical reward $\pi_k \in \arg\max_\pi d(P)_\pi^T \tilde{r}_k$

Play π_k, observe $\hat{V}_k$ and $\big\{(s_h^k, a_h^k)\big\}_{h=1}^H$

Update $A_k = A_{k-1} + \hat{d}_k \hat{d}_k^T$ and $Y_k = Y_{k-1} + \hat{d}_k \hat{V}_k$.

end for

To exemplify how the expected elliptical potential lemma is applied in the analysis of Algorithm 1 we supply a sketch of the proof.

Proof Sketch. By the optimism of the update rule, following (Abbasi-Yadkori, Pál, and Szepesvári 2011), it is possible to show that with high probability,

$$V_1^*(s_1^k) = d_{\pi^*}^T r \le d_{\pi_k}^T \hat{r}_{k-1} + l_{k-1} \|d_{\pi_k}\|_{A_{k-1}^{-1}},$$

for any $k > 0$. Thus, we only need to bound the *on-policy* prediction error given as follows,

$$\begin{aligned} \mathrm{R}(K) &= \sum_{k=1}^K (V_1^* - V_1^{\pi_k}) \\ &\le \sum_{k=1}^K (d_{\pi_k}^T \hat{r}_{k-1} + l_{k-1} \|d_{\pi_k}\|_{A_{k-1}^{-1}} - d_{\pi_k}^T r) \\ &\le 2 l_K \sum_{k=1}^K \|d_{\pi_k}\|_{A_{k-1}^{-1}} . \end{aligned} \tag{5}$$

where the last inequality can be derived using Proposition 1 and the Cauchy Schwartz inequality. Applying the expected elliptical potential lemma (Lemma 2) and setting $\lambda = H$ concludes the proof. □

Although Algorithm 1 provides a natural solution to the problem, it results in a major computational disadvantage. The optimization problem needed to be solved in each iteration is a convex maximization problem (known to generally be NP-hard (Atamtürk and Gómez 2017)). Furthermore, since $\|d_\pi\|_{A_{k-1}^{-1}}$ is non-linear in d_π, it restricts us from solving this problem by means of Dynamic Programming. In the next section, we follow a different route and formulate a *Thompson Sampling* based algorithm, with computational complexity that amounts to sampling a Gaussian noise for the reward and solving an MDP at each episode.

5 Thompson Sampling for RL with Trajectory Feedback

The OFUL-based algorithm for RL with trajectory feedback, analyzed in the previous section, was shown to give good performance in terms of regret. However, implementing the algorithm requires solving a convex maximization problem before each episode, which is, in general, computationally hard. Instead of following the OFUL-based approach, in this section, we analyze a Thompson Sampling (TS) approach for RL with trajectory feedback.

We start by studying the performance of Algorithm 2, which assumes access to the transition model (as in Section 4). Then, we study Algorithm 3 which generalizes the latter method to the case where the transition model is unknown. In this generalization, we use an optimistic-based approach to learn the *transition model*, and a TS-based approach to learn the *reward*. The combination of optimism and TS results in a tractable algorithm in which every iteration amounts to solving an empirical MDP (which can be done by Dynamic Programming). The reward estimator in both Algorithm 2 and Algorithm 3 is the same LS estimator (4) used for the OFUL-like algorithm. Finally, we focus on improving the most computationally-demanding stage of Algorithm 3, which is the reward sampling, and suggest a more efficient method in Algorithm 4.

5.1 TS for RL with Trajectory Feedback and Known Model

For general action sets, it is known that OFUL (Abbasi-Yadkori, Pál, and Szepesvári 2011) results in a computationally intractable update rule. One popular approach to mitigate the computational burden is to resort to TS for linear bandits (Agrawal and Goyal 2013). Then, the update rule amounts to solving a linear optimization problem over the action set. Yet, the reduced computational complexity of TS comes at the cost of an increase in the regret. Specifically, for linear bandit problems in dimension m, OFUL achieves $\tilde{\mathcal{O}}(m\sqrt{T})$, whereas TS achieves $\tilde{\mathcal{O}}(m^{3/2}\sqrt{T})$ (Agrawal and Goyal 2013; Abeille, Lazaric et al. 2017).

Algorithm 2 can be understood as a TS variant of Algorithm 1, much like TS for linear bandits (Agrawal and Goyal 2013) is a TS variant of OFUL. Unlike the common TS algorithm for linear bandits, Algorithm 2 uses the LS estimator in Section 3, i.e., the one which uses the empirical state-action distributions $\hat{d}_k$, instead of the 'true action' d_{π_k}. In terms of analysis, we deal with this discrepancy by applying – as in Section 4 – the expected elliptical potential lemma,[1] instead of the standard elliptical potential lemma. Then, by extending techniques from (Agrawal and Goyal 2013; Russo 2019) we obtain the following performance guarantee for Algorithm 2 (see Appendix D for the full proof).

Theorem 4 (TS for RL with Trajectory Feedback and Known Model)**.** *For any $\delta \in (0,1)$, it holds with probability greater than $1-\delta$ that for all $K > 0$,*

$$\begin{aligned} \mathrm{R}(K) \le \mathcal{O}\Big((SA)^{3/2} H \sqrt{K \log(K)} \log\Big(\frac{KH}{\delta}\Big)\Big) \\ + \mathcal{O}\Big(SAH \sqrt{\log\Big(\frac{KH^2}{\delta}\Big)}\Big). \end{aligned}$$

[1] We use a slightly more general version of the expected elliptical potential lemma, presented in Appendix B

Observe that Theorem 4 establishes a regret guarantee of $m^{3/2}\sqrt{K}$ since the dimension of the specific linear bandit problem is $m = SA$ (see (1)). This is the type of regret is expected due to TS type of analysis (Agrawal and Goyal 2013). It is an interesting question whether this bound can be improved due to the structure of the problem.

5.2 UCBVI-TS for RL with Trajectory Feedback

In previous sections, we devised algorithms for RL with trajectory feedback, assuming access to the true transition model and that only the reward function is needed to be learned. In this section, we relax this assumption and study the setting in which the transition model is also unknown.

This setting highlights the importance of the LS estimator (4), which uses the empirical state-action frequency $\hat{d}_k$, instead of d_{π_k}. I.e., when the transition model is not known, we do not have access to d_{π_k}. Nevertheless, it is reasonable to assume access to $\hat{d}_k$ since it only depends on the *observed* sequence of state-action pairs in the k^{th} episode $\{s_h^k, a_h^k\}_{h=1}^H$ and does not require any access to the true model. For this reason, the LS estimator (4) is much more amenable to use in RL with trajectory feedback when the transition model *is not given* and needed to be estimated.

Algorithm 3, which we refer to as UCBVI-TS (Upper Confidence Bound Value Iteration and Thompson Sampling), uses a combined TS and optimistic approach for RL with trajectory feedback. At each episode, the algorithm perturbs the LS estimation of the reward $\hat{r}_{k-1}$ by a random Gaussian noise ξ_k, similarly to Algorithm 2. Furthermore, to encourage the agent to learn the unknown transition model, UCBVI-TS adds to the reward estimation the bonus $b_{k-1}^{pv} \in \mathbb{R}^{SA}$ where

$$b_{k-1}^{pv}(s,a) \simeq \frac{H}{\sqrt{n_{k-1}(s,a) \vee 1}}, \tag{6}$$

up to logarithmic factors (similarly to Azar, Osband, and Munos 2017). Then, it simply solves the empirical MDP defined by the plug-in transition model $\bar{P}_{k-1}$ and the reward function $\hat{r}_{k-1} + \xi_k + b_{k-1}^{pv}$. Specifically, the transition kernel $\bar{P}_{k-1}$ is estimated by

$$\bar{P}_k(s' \,|\, s,a) = \frac{\sum_{l=1}^{k} \sum_{h=1}^{H} 1\big(s_h^l = s, a_h^l = a, s_{h+1}^l = s'\big)}{n_k(s,a) \vee 1}. \tag{7}$$

The next result establishes a performance guarantee for UCBVI-TS with trajectory feedback (see proof in Appendix E.3). The key idea for the proof is showing that the additional bonus term (6) induces sufficient amount of optimism with fixed probability. Then, by generalizing the analysis of Theorem 4 while using some structural properties of MDPs we derive the final result.

Theorem 5 (UCBVI-TS Performance Guarantee)**.** *For any $\delta \in (0,1)$, it holds with probability greater than $1-\delta$ that for all $K > 0$,*

$$\mathrm{R}(K) \le \mathcal{O}\left(SH(SA+H)\sqrt{AHK \log K} \log\left(\frac{SAHK}{\delta}\right)^{\frac{3}{2}} \right) + \mathcal{O}\left(H^2 \sqrt{S}(SA+H)^2 \log\left(\frac{SAHK}{\delta}\right)^2 \sqrt{\log K} \right)$$

Algorithm 3 UCBVI-TS for RL with Trajectory Feedback

Require: $\delta \in (0,1)$, $\lambda = H$, $v_k = \sqrt{9SAH \log \frac{kH^2}{\delta/10}}$,
$l_k = \sqrt{\frac{1}{4} SAH \log\Big(\frac{1+kH^2/\lambda}{\delta/10}\Big)} + \sqrt{\lambda SA}$,
$b_k^{pv}(s,a) = \sqrt{\frac{H^2 \log \frac{40SAH^2k^3}{\delta}}{n_k(s,a) \vee 1}}$
Initialize: $A_0 = \lambda I_{SA}$, $Y_0 = \mathbf{0}_{SA}$,
Counters $n_0(s,a) = 0$, $\forall s, a$.
for $k = 1, ..., K$ **do**
Calculate $\hat{r}_{k-1}$ via LS estimation (4) and $\bar{P}_k$ by (7)
Draw noise $\xi_k \sim \mathcal{N}(0, v_k^2 A_{k-1}^{-1})$ and define $\tilde{r}_k^b = \hat{r}_{k-1} + \xi_k + b_{k-1}^{pv}$
Solve empirical MDP with optimistic-perturbed reward, $\pi_k \in \arg\max_\pi d(\bar{P}_{k-1})^T \tilde{r}_k^b$
Play π_k, observe $\hat{V}_k$ and $\{(s_h^k, a_h^k)\}_{h=1}^H$
Update counters $n_k(s,a)$, $A_k = A_{k-1} + \hat{d}_k \hat{d}_k^T$ and $Y_k = Y_{k-1} + \hat{d}_k \hat{V}_k$.
end for

thus, discarding logarithmic factors and constants and assuming $SA \ge H$, $\mathrm{R}(K) \le \tilde{\mathcal{O}}\Big(S^2 A^{3/2} H^{3/2} \sqrt{K}\Big)$.

5.3 Improving the Computational Efficiency of UCBVI-TS

In this section, we present a modification to UCBVI-TS that uses a doubling trick to improve the computational efficiency of Algorithm 3. Specifically, for $m = SA$, the complexity of different parts of UCBVI-TS is as follows:

- A_k^{-1} can be iteratively updated using $\mathcal{O}(m^2)$ computations by the Sherman-Morrison formula (Bartlett 1951).
- Given A_k^{-1}, calculating $\hat{r}_k$ requires $\mathcal{O}(m^2)$ computations.
- Generating the noise ξ_k requires calculating $A_k^{-1/2}$ that, in general, requires performing a singular value decomposition to A_k^{-1} at a cost of $\mathcal{O}(m^3)$ computations.
- Finally, calculating the policy using dynamic programming requires $\mathcal{O}(S^2AH)$ computations.

In overall, UCBVI-TS requires $\mathcal{O}\big((SA)^3 + S^2A(H+A)\big)$ computations per episode, where the most demanding part is the noise generation. Thus, we suggest a variant of our algorithm, called Rarely-Switching UCBVI-TS (see Appendix F for the full description of the algorithm), that updates A_k (and, as a result, the LS estimator) only after the updates increase the determinant of A_k by a factor of $1+C$, for some $C > 0$, similarly to (Abbasi-Yadkori, Pál, and Szepesvári 2011). Specifically, we let $B_k = \lambda I_{SA} + \sum_{s=1}^k \hat{d}_s \hat{d}_s^T$ and update $A_k = B_k$ if $\det(B_k) > (1+C)\det(A_k)$, where $B_0 = A_0 = \lambda I_{SA}$. Otherwise, we keep $A_k = A_{k-1}$. By the matrix-determinant lemma, $\det(B_k)$ can be iteratively updated by $\det(B_k) = \Big(1 + \hat{d}_k^T B_{k-1}^{-1} \hat{d}_k\Big)\det(B_{k-1})$, which requires $\mathcal{O}(SA)$ calculations given B_{k-1}^{-1}; in turn, B_{k-1}^{-1} can be updated by the Sherman-Morrison formula. Notably, A_k is rarely updated, as we prove in the following lemma:

Lemma 6. *Under the update rule of Rarely-Switching UCBVI-TS, and for any $C > 0, \lambda > 0$, it holds that $\sum_{k=1}^{K} 1(A_k \neq A_{k-1}) \leq \frac{SA}{\log(1+C)} \log\left(1 + \frac{KH^2}{\lambda SA}\right)$.*

Therefore, the average per-round computational complexity of Rarely-Switching UCBVI-TS after K episodes is

$$\mathcal{O}\left(S^2A(H+A) + \frac{(SA)^4}{\log(1+C)} \frac{\log \frac{KH}{SA}}{K} \right) .$$

Moreover, rarely updating A_k only affects the lower-order terms of the regret, as we prove in the following theorem:

Theorem 7 (Rarely-Switching UCBVI-TS Performance Guarantee)**.** *For any $\delta \in (0,1)$, it holds with probability greater than $1-\delta$ that for all $K > 0$,*

$$\mathrm{R}(K) \leq \tilde{\mathcal{O}}\Big(SH(SA+H)\sqrt{AHK} + H^2\sqrt{S}(SA+H)^2\Big) + \tilde{\mathcal{O}}\Big((SA)^{3/2}H\sqrt{(1+C)K}\Big) .$$

The proof can be found in Appendix F. See that the difference from Theorem 5 is in the last term, which is negligible, compared to the first term, for reasonable values of $C > 0$.

6 Discussion and Conclusions

In this work, we formulated the framework of RL with trajectory feedback and studied different RL algorithms in the presence of such feedback. Indeed, in practical scenarios, such feedback is more reasonable to have, as it requires a weaker type of feedback relative to the standard RL one. For this reason, we believe studying it and understanding the gaps between the trajectory feedback RL and standard RL is of importance. The central result of this work is a hybrid optimistic-TS based RL algorithm with a provably bounded $\sqrt{K}$ regret that can be applied when both the reward and transition model are unknown and, thus, needed to be learned. Importantly, the suggested algorithm is computationally tractable, as it requires to solve an empirical MDPs and not a convex maximization problem.

Regret minimization for standard RL has been extensively studied. Previous algorithms for this scenario can be roughly divided into optimistic algorithms (Jaksch, Ortner, and Auer 2010; Azar, Osband, and Munos 2017; Jin et al. 2018; Dann et al. 2019; Zanette and Brunskill 2019; Simchowitz and Jamieson 2019; Efroni et al. 2019) and Thompson-Sampling (or Posterior-Sampling) based algorithms (Osband, Russo, and Van Roy 2013; Gopalan and Mannor 2015; Osband and Van Roy 2017; Russo 2019). Nonetheless, and to the best of our knowledge, we are the first to present a hybrid approach that utilizes both concepts in the same algorithm. Specifically, we combine the optimistic confidence-intervals of UCBVI (Azar, Osband, and Munos 2017) alongside linear TS for the reward and also take advantage of analysis tools for posterior sampling in RL (Russo 2019).

In the presence of trajectory-feedback, our algorithms make use of concepts from linear bandits to learn the reward. Specifically, we use both OFUL (Abbasi-Yadkori, Pál, and Szepesvári 2011) and linear TS (Agrawal and Goyal 2013; Abeille, Lazaric et al. 2017), whose regret bounds for m-dimension problems after K time-steps with 1-subgaussian noise are $\tilde{\mathcal{O}}\Big\{m\sqrt{K}\Big\}$ and $\tilde{\mathcal{O}}\Big\{m^{3/2}\sqrt{K}\Big\}$, respectively. These bounds directly affect the performance in the RL setting, but the adaptation of OFUL leads to a computationally-intractable algorithm. In addition, when there are at most N context, it is possible to achieve a regret bound of $\tilde{\mathcal{O}}\big\{\sqrt{mK\log N}\big\}$ (Chu et al. 2011); however, the number of deterministic policies, which are the number of 'contexts' for RL with trajectory-feedback, is exponential in S, namely, A^{SH}. Therefore, such approaches will lead to similar guarantees to OFUL and will also be computationally intractable.

In terms of regret bounds, the minimax regret in the standard RL setting is $\tilde{\mathcal{O}}\Big\{\sqrt{SAHT}\Big\}$ (Osband and Van Roy 2016; Azar, Osband, and Munos 2017), however, for standard RL the reward feedback is much stronger than for RL with trajectory feedback. For linear bandits with $\sqrt{H}$-subgaussian noise, the minimax performance bounds are $\tilde{\mathcal{O}}\Big\{m\sqrt{HK}\Big\}$ (Dani, Hayes, and Kakade 2008). Specifically, in RL we set $m = SA$, which leads to $\tilde{\mathcal{O}}\Big\{SA\sqrt{HK}\Big\}$. Nonetheless, for RL with trajectory feedback and known model, the context space is the average occupancy measures d_π, which is heavily-structured. It is an open question whether the minimax regret bound remains $\tilde{\mathcal{O}}\Big\{SA\sqrt{HK}\Big\}$ for RL with trajectory feedback, when the transition model is known, or whether it can be improved. Moreover, when the model is unknown, our algorithm enjoys a regret of $\tilde{\mathcal{O}}\Big(S^2A^{3/2}H^{3/2}\sqrt{K}\Big)$ when $H \leq SA$. A factor of $\sqrt{SA}$ is a direct result of the TS-approach, that was required to make to algorithm tractable, and an additional $\sqrt{S}$ appears when the model is unknown. Moreover, extending OFUL to the case of unknown model and following a similar analysis to Theorem 5 would still yield this extra $\sqrt{S}$ factor (and would result in a computationally hard algorithm), in comparison to when we know the model. It is an open question whether this additional factor can also be improved.

Finally, we believe that this work paves the way to many interesting future research directions, notably, studying RL with additional, more realistic, feedback models of the reward. Furthermore, we believe that the results can be adapted to cases where the feedback is a more complex mapping from state-actions into trajectory-reward, and, specifically, a noisy generalized linear model (GLM) of the trajectory (Filippi et al. 2010; Abeille, Lazaric et al. 2017; Kveton et al. 2020). In this case, even though the reward function is not Markovian, our approach should allow deriving regret bounds. More generally, this can be viewed as a form of reward-shaping with theoretical guarantees, which is, in general, an open question.

Acknowledgments

This work was partially funded by the Israel Science Foundation under ISF grant number 2199/20. YE is partially supported by the Viterbi scholarship, Technion. NM is partially

supported by the Gutwirth Scholarship.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2312–2320.

Abeille, M.; Lazaric, A.; et al. 2017. Linear thompson sampling revisited. *Electronic Journal of Statistics* 11(2): 5165–5197.

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2020. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, 64–66. PMLR.

Agrawal, S.; and Goyal, N. 2013. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 127–135.

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Atamtürk, A.; and Gómez, A. 2017. Maximizing a class of utility functions over the vertices of a polytope. *Operations Research* 65(2): 433–445.

Azar, M. G.; Osband, I.; and Munos, R. 2017. Minimax regret bounds for reinforcement learning. *arXiv preprint arXiv:1703.05449* .

Bartlett, M. S. 1951. An inverse matrix adjustment arising in discriminant analysis. *The Annals of Mathematical Statistics* 22(1).

Chu, W.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

Dani, V.; Hayes, T.; and Kakade, S. 2008. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*. Citeseer.

Dann, C.; Li, L.; Wei, W.; and Brunskill, E. 2019. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, 1507–1516.

Efroni, Y.; Merlis, N.; Ghavamzadeh, M.; and Mannor, S. 2019. Tight regret bounds for model-based reinforcement learning with greedy policies. In *Advances in Neural Information Processing Systems*, 12224–12234.

Filippi, S.; Cappe, O.; Garivier, A.; and Szepesvári, C. 2010. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, 586–594.

Gopalan, A.; and Mannor, S. 2015. Thompson sampling for learning parameterized Markov decision processes. In *Conference on Learning Theory*, 861–898.

Jaksch, T.; Ortner, R.; and Auer, P. 2010. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research* 11(Apr): 1563–1600.

Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. I. 2018. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 4863–4873.

Kveton, B.; Zaheer, M.; Szepesvari, C.; Li, L.; Ghavamzadeh, M.; and Boutilier, C. 2020. Randomized exploration in generalized linear bandits. In *International Conference on Artificial Intelligence and Statistics*, 2066–2076.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research* 17(1): 1334–1373.

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

Osband, I.; Russo, D.; and Van Roy, B. 2013. (More) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, 3003–3011.

Osband, I.; and Van Roy, B. 2016. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732* .

Osband, I.; and Van Roy, B. 2017. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning*, 2701–2710.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc.

Russo, D. 2019. Worst-case regret bounds for exploration via randomized value functions. In *Advances in Neural Information Processing Systems*, 14433–14443.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of Go without human knowledge. *Nature* 550(7676): 354.

Simchowitz, M.; and Jamieson, K. G. 2019. Non-asymptotic gap-dependent regret bounds for tabular mdps. In *Advances in Neural Information Processing Systems*, 1153–1162.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4): 229–256.

Zanette, A.; and Brunskill, E. 2019. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. *arXiv preprint arXiv:1901.00210* .

Zimin, A.; and Neu, G. 2013. Online learning in episodic Markovian decision processes by relative entropy policy search. In *NIPS*, 1583–1591.