INTRODUCING COORDINATION IN CONCURRENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Research on exploration in reinforcement learning has mostly focused on problems with a single agent interacting with an environment. However many problems are better addressed by the concurrent reinforcement learning paradigm, where multiple agents operate in a common environment. Recent work has tackled the challenge of exploration in this particular setting (Dimakopoulou & Van Roy, 2018; Dimakopoulou et al., 2018). Nonetheless, they do not completely leverage the characteristics of this framework and agents end up behaving independently from each other. In this work we argue that coordination among concurrent agents is crucial for efficient exploration. We introduce coordination in Thompson Sampling based methods by drawing correlated samples from an agent's posterior. We apply this idea to extend existing exploration schemes such as randomized least squares value iteration (RLSVI). Empirical results emphasize the merits of our approach and call attention to coordination as a key objective for efficient exploration in concurrent reinforcement learning.

1 INTRODUCTION

At the heart of reinforcement learning is the exploration-exploitation trade-off that describes an agent's dilemma to balance maximizing its cumulative rewards and improving its knowledge of the environment. While there have been a lot of attempts in recent years to improve practical exploration (Bellemare et al., 2016; Osband et al., 2016a; Pathak et al., 2017; Burda et al., 2019), existing work has mainly focused on fully sequential problems where a single agent interacts with an environment. However, it is sometimes beneficial or even necessary to have multiple agents interacting concurrently with the environment. For example, clinical trials are run in different phases, in each of them a group of patients are studied simultaneously, results are then collected to facilitate the design of the next phase (Perchet et al., 2015). Likewise, in many commercial applications a company might interact with many customers at the same time before gathering their feedback to improve its strategy.

This setting corresponds to concurrent reinforcement learning (Silver et al., 2013; Pazis & Parr, 2016), a framework where multiple agents interact with an environment while sharing data to learn in parallel. To learn efficiently within this paradigm agents should coordinate their exploratory effort which adds a layer of complexity to the already difficult exploration problem. There has been some interest in designing exploration methods suited for concurrent learning (Dimakopoulou & Van Roy, 2018; Dimakopoulou et al., 2018; Amin et al., 2021) however none of them actually achieve *coordinated exploration*. That is, agents do not leverage other agents acting at the same time to influence their decisions. Instead, agents will usually behave independently from each other conditioned on the history. In the end, these techniques fail to fully satisfy the diversity property that Dimakopoulou & Van Roy (2018) deemed necessary for efficient exploration in concurrent reinforcement learning.

Our aim in this work is to highlight the importance of coordination in concurrent exploration. As an introduction we provide a regret bound for Thompson Sampling based algorithms for concurrent learning such as seed sampling (Dimakopoulou & Van Roy, 2018). Then, we propose a simple coordination mechanism to introduce cooperation between parallel agents. This mechanism is first used to create a bandit algorithm for multi-armed bandits with Gaussian rewards. We also use the same strategy to extend Randomized Least Square Value Iteration (RLSVI; Wen (2014); Osband et al. (2016b; 2018)). We evaluate the empirical performance of these algorithms on tabular problems and show that it leads to improved performance over non-coordinated baselines. Altogether, given

the growing interest in concurrent reinforcement learning (Kalashnikov et al., 2018; Bodnar et al., 2020; Kalashnikov et al., 2021), our work demonstrates the relevance of coordination as a research direction to improve exploration in reinforcement learning.

2 Problem Formulation

We consider a Markov decision process (MDP) M represented by a tuple $\langle \mathcal{S}, \mathcal{A}, P^M, r^M, H, \rho_1 \rangle$ with $\mathcal{S}$ the state space, $\mathcal{A}$ the finite set of actions, $P^M : \mathcal{S} \times A \leftarrow [0,1]^{\mathcal{S}}$ the transition probability distribution, $r^M : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$ the reward function, $H \in \mathbb{N}^*$ the horizon and ρ_1 the initial state distribution. We will write $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$ for the number of states and actions. Let $G \in \mathbb{N}^*$ be the number of agents that operate in parallel. At the beginning of episode k, the g-th agent begins at state $s^g_{k,1}$ sampled according to ρ_1. Then, at depth h each agent takes an action $a^g_{k,h}$, gets a reward $r^g_{kh} \sim r(s^g_{k,h}, a^g_{k,h})$ and transits to a new state $s^g_{h+1} \sim P(s^g_{k,h}, a^g_{k,h})$. We assume that agents share information in real time. We also write $\mathcal{H}_{kh}$ the concatenated history for all agents up to episode k at depth h.

Given a policy π, we define the value function for each agent g at depth h to be the expected sum of rewards to be collected from the current state when following policy π:

$$V^{M,\pi}_h(s) \coloneqq \mathbb{E}\big[\sum_{h'=h}^{H} r^M(s_{h'}, a_{h'}) | s_h = s, a_{h'} \sim \pi(s_{h'}, h')\big]. \tag{1}$$

We will also write $V^{M,\pi}_{H+1}(s) = 0$ for all $s \in \mathcal{S}$. The Bellman Equation applied in MDP M states that:

$$V^{M,\pi}_h(s) = \mathbb{E}_{a \sim \pi(s,h)}\Big[r^M(s,a) + P^M(s,a)^\top V^{M,\pi}_{h+1}\Big] \tag{2}$$

We denote by M^* the true MDP with unknown rewards and transition R^* and P^*. Let π^* be an optimal policy. We define the regret incurred by an agent g following policy π up to time T by

$$\text{Regret}(T, \pi, M^*) \coloneqq \sum_{k=1}^{\lfloor T/H \rfloor} \Delta^g_k, \tag{3}$$

where Δ^g_k denotes regret over the kth episode

$$\Delta^g_k \coloneqq \mathbb{E}_{s \sim \rho_1}\big[V^{M^*,\pi^*}_1(s) - V^{M^*,\pi^g_k}_1(s)\big]. \tag{4}$$

We model the initial uncertainty of the environment with a prior distribution ϕ and assume that M^* is a sample from that prior. We then define the Bayesian regret as:

$$\text{BayesRegret}(T, \pi, \phi) \coloneqq \mathbb{E}_{M^* \sim \phi}\big[\text{Regret}(T, \pi, M^*)\big]. \tag{5}$$

The per-agent regret experienced by the population of agents is given by, for $K = \lfloor T/H \rfloor$:

$$\text{MeanBayesRegret}(T, \pi, \phi) = \mathbb{E}_{M^* \sim \phi}\big[\sum_{k=1}^{K} \frac{1}{G} \sum_{g=1}^{G} \Delta^g_k\big].$$

3 How can a team of agents explore efficiently?

In this section we take a closer look at exploration in the concurrent setting and the potential benefits of coordination.

3.1 Concurrent exploration

Recently Dimakopoulou & Van Roy (2018) have shown that posterior sampling for reinforcement learning (PSRL) could be extended with success to concurrent learning. PSRL extends Thompson Sampling to reinforcement learning and operates by sampling a statistically plausible model of the environment given a prior and data previously collected. The agent then acts according to this model

Algorithm 1 Concurrent PSRL (Dimakopoulou & Van Roy, 2018)

Input: Prior distribution ϕ, G number of agents
for $k = 1, 2, \cdots$ **do**
  for agent $g = 1, \cdots, G$ **do**
    Sample $M_k^g \sim \phi(\cdot|\mathcal{H}_{k1})$
    Compute the optimal policy for M_k^g

$$\pi_k^g = \arg\max_{\pi} V_{k,1}^{M_k^g,\pi}$$

  end for
  for $h = 1, \cdots, H$ **do**
    Act: $a_{k,h}^g \sim \pi_k^g(s_{k,h}^g)$
    Observe r_{kh}^g and s_{kh+1}^g
  end for
  Update: $\mathcal{H}_{k+1,1} \leftarrow \mathcal{H}_{k,1} \cup \{(s_{k,h}^g, a_{k,h}^g, r_{kh}^g)_{h,g}\}$
end for

to gather more information and update its posterior. A natural extension of PSRL is to draw multiple samples from the posterior – one for each concurrent agent – as done in Algorithm 1. Dimakopoulou & Van Roy (2018) follow this idea by drawing multiple independent samples from the posterior. Our first contribution is to provide the following regret bound for Concurrent PSRL

Theorem 1. *The mean Bayes regret of Algorithm 1 is bounded by*

$$\textit{MeanBayesRegret}(T, \pi, \phi) \leq \tilde{O}(HS\sqrt{\frac{AT}{G}}). \tag{6}$$

Proof. All proofs are in the appendix. □

This results shows that the average regret of concurrent posterior sampling scales in $O(1/\sqrt{G})$ and that the total regret of Algorithm 1 scales in $O(HS\sqrt{AGT})$. It is significant because it means that it is as efficient to run concurrent posterior sampling with G agents for T timesteps as it is to run posterior sampling with a single agent for GT timesteps. Similar results were found in the bandit setting by Contal et al. (2013) and Kandasamy et al. (2018).

3.2 The issue with independent sampling

Concurrent Posterior Sampling requires that each sample is drawn from the posterior but it does not impose any constraint on their joint distribution. This gives us some flexibility in the choice of this joint distribution. Options include

- Draw a single sample from the posterior that is then used for every agent.
- Draw independent samples from the posterior.
- Draw correlated samples from the posterior.

Dimakopoulou & Van Roy (2018) showed that the first option is inefficient because agents do not work together to explore faster. Instead, their algorithm follows the second option and draw independent samples from the posterior. Though they argue it allows agents to diversify their exploratory effort, independence does not guarantee that agents will coordinate. There remains some probability that some actions are repeated while other equally promising actions are ignored. In the context of exploration, it would mean trying few state-action pairs several times as opposed to trying more state-action pairs fewer times. The former provides less information about the environment and leads to sub-optimal exploration. We propose to reach the favored behavior by adding correlation between samples from the posterior so that parallel agents can cooperate. The main idea in this paper is to introduce coordination through the following principle.

Principle 1. Let s be a state such that all actions in s are equally likely to be the optimal (from the exploration algorithm point-of-view) at episode k. We define the probability distribution

$$\mathcal{Q}_{s,k,h} = \left(\frac{\sum_g \mathbb{1}_{\{a^g_{k,h}=a\}}}{G} \right)_{a \in \mathcal{A}}. \tag{7}$$

$\mathcal{Q}_{s,k,h}$ is the distribution of actions sampled by agents in s at episode k and timestep h. Then a coordinated exploration algorithm should maximize the entropy of $\mathcal{Q}_{s,k,h}$.

In words, Principle 1 expresses that when all actions in a state are equally promising, agents should try all possible actions roughly the same number of times. We will see in the section how this can help us design new coordinated exploration algorithms.

4 Bandit with Gaussian rewards

To demonstrate the benefits of coordination we start with a simple multi-armed bandit problem with A arms. The reward function for each arm is normally distributed with a known variance σ^2. Agents are uncertain about the mean reward for each arm over which they share a common $\mathcal{N}(\mu_0, \sigma_0)$ prior.

4.1 Coordinated Thompson Sampling

To derive an implementation of Algorithm 1 that validates Principle 1 we must find a way to influence the joint distribution of posterior samples without affecting their marginals. To do so we introduce the concept of a *preference function*. Let $f_k : [\![1, \cdots, G]\!] \to \mathcal{A}$ be a preference function. For each agent g, $f_k(g)$ encodes its affinity for a specific action at timestep k. We design our algorithm so that *when all possible actions are equally likely to be optimal under the posterior agent g will pull arm $f_k(g)$*. By enforcing that the set $\{f_k(g)\}_{g \in [\![1,\cdots,G]\!]}$ covers the action space as well as possible we can validate Principle 1 and promote coordination between agents. Algorithm 2 presents Coordinated Thompson Sampling (CoordTS) an extension of Thompson Sampling with a coordination mechanism.

This mechanism relies on the relationship between sampling from a normal distribution and sampling from a standard normal distribution. For any arm with posterior $\mathcal{N}(\mu_k, \sigma_k^2)$

$$z \sim \mathcal{N}(\mu_k, \sigma_k^2) \iff \epsilon \sim \mathcal{N}(0,1),\ z = \mu_k + \sigma_k \epsilon. \tag{8}$$

Thompson Sampling proceeds by selecting the arm with the highest posterior sample. When all arms have the same posterior distribution (i.e $\mu_1 = \cdots = \mu_N$ and $\sigma_1 = \cdots = \sigma_N$) this maximum is achieved by the arm with the highest sample ϵ. We can create a preference for a specific arm by assigning the maximum of ϵ to this arm. Dimakopoulou & Van Roy (2018)'s seed sampling algorithm also made use of Equation equation 8 and they referred to ϵ as a *seed*, in the remainder of this paper we will choose instead the term *standard normal vector*.

The sampling procedure used to draw a preference function at each timestep is given by Algorithm 3. The algorithm ensures that preferences are distributed almost equally among all actions. The following lemma shows that if preferences are sampled according to Algorithm 3 then Principle 1 will be verified.

Lemma 1. *We define*

$$\mathcal{P}_k = \left(\frac{\sum_g \mathbb{1}_{\{f_k(g)=a\}}}{G} \right)_{a \in \mathcal{A}}, \qquad \mathcal{Q}_k = \left(\frac{\sum_g \mathbb{1}_{\{a^g_k=a\}}}{G} \right)_{a \in \mathcal{A}}. \tag{9}$$

$\mathcal{P}_k$ and $\mathcal{Q}_k$ correspond to the distribution of preferences and actions at timestep k. Then, Algorithm 3 maximizes the entropy of $\mathcal{P}_k$. Furthermore, when all arms have the same distribution under the posterior then the entropy of $\mathcal{Q}_k$ is also maximized.

4.2 Analysis

To better understand the behavior of CoordTS we study the simpler two armed bandit setting ($A = 2$, $\mathcal{A} = \{a_1, a_2\}$). Let g be an agent with a preference for action a_1. At each timestep, g will draw independently $\epsilon_1, \epsilon_2 \sim \mathcal{N}(0,1)$ then use $\epsilon = (\max(\epsilon_1, \epsilon_2), \min(\epsilon_1, \epsilon_1))$ to compute a sample from the posterior. Because of this reshuffling, ϵ is no longer Gaussian. The distribution of ϵ is made more precise by the following lemma.

Algorithm 2 Coordinated Thompson Sampling

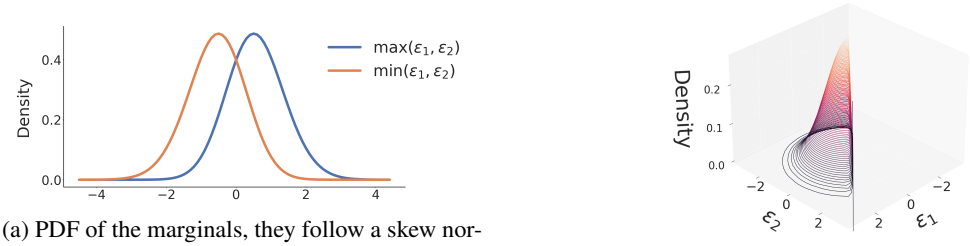
Input: Gaussian prior $p : \mathcal{N}(\mu_0, \sigma_0^2 I_n)$, $G \in \mathbb{N}^+$ number of agents, $N \in \mathbb{N}^+$ number of arms, $f_k : [\![1, \cdots, G]\!] \to \mathcal{A}$ preference functions
for $k = 1, \cdots$ **do**
  for agent $g = 1, \cdots, G$ **do**
    // Sample a standard normal vector
    Sample $\epsilon^g \sim \mathcal{N}(0, I_n)$
    $j \leftarrow \arg\max_i \epsilon^g[i]$
    $m \leftarrow f_k(g)$
    Swap values by setting $\epsilon^g[m] = \epsilon^g[j]$ and $\epsilon^g[j] = \epsilon^g[m]$
    // Compute posterior sample
    $\tilde{\theta}_h^k = \mu_k + \sigma_k \epsilon^g$
  end for
  Act: $a_k^g = \arg\max_{a \in \mathcal{A}} \tilde{\theta}_k^g[a]$, $\forall g$
  Observe $r(a_k^g)$, $\forall g$
  Update: $p \leftarrow \mathbb{P}\Big(\cdot \,|p, a_k^1, r(a_k^1), \cdots, a_k^G, r(a_k^G)\Big)$
end for

Algorithm 3 Pigeonhole sampling

Input: $G \in \mathbb{N}^+$ number of agents, $N \in \mathbb{N}^+$ number of arms
Initialize $m \in \mathbb{R}^N$, $f \in \mathbb{R}^G$, with $m[i] = 0$ and $f[i] = 0$ $\forall i$
for $g = 1, \cdots, G$ **do**
  c $\leftarrow \arg\min_i m[i]$
  Draw index j uniformly random from c
  f[g] = j
  m[j] = m[j] + 1
end for
Return f

Lemma 2. *Let X_1, X_2 be two iid random variables distributed according to $\mathcal{N}(0, 1)$. Let $Z = (X, Y)$ with $X = \max(X_1, X_2)$ and $Y = \min(X_1, X_2)$. Then X and Y follow a skew normal distribution with scale $+1$ and -1 and Z follows a half-normal distribution with support on $\{(x, y) \in \mathbb{R}^2 | x \geq y\}$.*

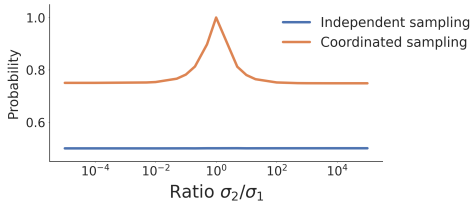
Figure 1 shows the joint distribution and marginals of ϵ. If instead g were to have a preference for action a_2 then its standard normal vector's marginals would also follow a skew normal distribution with scale -1 and $+1$ and its joint distribution would be the missing half of the truncated normal distribution in Figure 1b. Altogether, while ϵ is not Gaussian when the preference of g is known, it is when the preference of g has been marginalized. Algorithm 2 achieves coordination by enforcing a



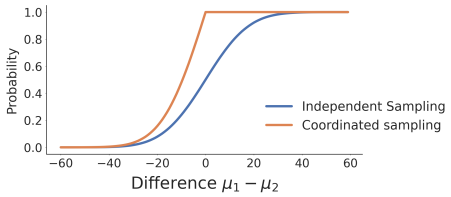
(a) PDF of the marginals, they follow a skew normal distribution with scale $+1$ and -1.

(b) PDF of the joint distribution, a half-normal distribution with support on $\{\epsilon_1 \geq \epsilon_2\}$

Figure 1: Marginals and joint distribution for a standard normal vector with a preference for the first action in a two armed Gaussian bandit.



(a) Probability of selecting action a_1 as a function of σ_2/σ_1 when $\mu_1 = \mu_2$.



(b) Probability of selecting action a_1 as a function of $\mu_1 - \mu_2$ when $\sigma_1 = \sigma_2$

Figure 2: Probability to sample action 0 in a two arm bandit problem with posterior $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$ with independent sampling and coordination with a preference for action 0.

good coverage of the posterior, ensuring that when agents sample their standard normal vector they draw as many times from the two halves of the distribution separated by the line $\{\epsilon_1 = \epsilon_2\}$.

The condition mentioned in Principle 1 may never be verified. We may wonder how CoordTS behave outside of this rare event and if it still allows agents to coordinate. To answer that question we can numerically estimate the probability of choosing each action. Figure 2 shows the probability of selecting action a_1 for independent and coordinated sampling as a function of the posterior distribution of the two arms. We observe that an agent with a preference for an action will select it more than it would if agents were independent. This shows that, even when actions do not have the exact same probability to be optimal, parallel agents can coordinate. This highlights the relevance of the term *preference* to describe the agent's affinity for a particular action.

We now consider the general case ($A \geq 2$). Let $e = (e_1, \cdots, e_N)$ be the canonical basis of the Euclidean space $\mathbb{R}^N$. For $i \in [\![1, \cdots, N]\!]$ we define $F_i = \{x \in \mathbb{R}^N | \forall j,\ x^T e_i \geq x^T e_j\}$. The distribution of the standard normal vector of an agent with a preference for action a_i will be a truncated normal distribution with support on F_i. Sampling from a truncated normal distribution is usually done through rejection sampling, in this particular case the same can be done by sampling from a standard normal distribution and swapping two coordinates. Algorithm 2 ensures that at each timestep standard normal vectors sampled are distributed almost equally among each F_i. This is made more precise by the following lemma which follows directly from Lemma 1

Lemma 3. *Consider a team of G agents and $N_k(F_i)$ the number of standard normal vectors sampled at timestep k that belong to subspace F_i. Then, we have $\forall i, j \in [\![1, ..., N]\!]$, $|N_k(F_i) - N_k(F_j)| \leq 1$.*

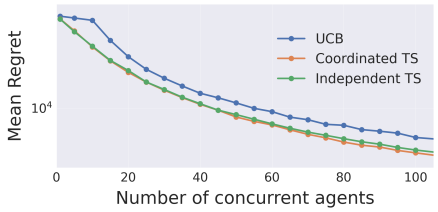
Because $(F_i)_{i \in [\![1, \cdots, N]\!]}$ define a partition of $\mathbb{R}^N$, standard normal vectors sampled by Algorithm 4 are normally distributed when agents' preferences have been marginalized.

4.3 Empirical evaluation

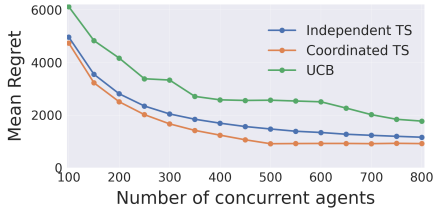
We now evaluate CoordTS empirical performance on our Gaussian bandit problem. As a baseline, we compare with Upper Confidence Bounds (UCB) and Thompson Sampling with independent agents. We use the following parameters for the problem: $\mu_0 = 0, \sigma_0 = 300, \sigma = 10, A = 500$. Figures 3a and 3b show the mean regret as a function of the number of concurrent agents after 50 episodes, results are averaged over 500 runs. We observe a clear improvement using correlated sampling over independent sampling. Both algorithms perform similarly when the number of agents is small however the gap widens as the number of agents grows. With 10 agents CoordTS provides a 0.5% improvement over independent Thompson Sampling, this gap increases to almost 40% with 500 agents. As a whole, our experiments underline the potential of coordinated exploration.

5 Coordination with linear function approximation

The previous section provided a simple example where coordination between agent can be beneficial. In this section we extend those ideas and focus on the more general reinforcement learning problem with linear function approximation. We assume that we are provided a state representation $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. We write $\Phi_h \in \mathbb{R}^{SA \times d}$ the feature matrix $\Phi_h := [\phi_h(s, a)]_{(s,a) \in \mathcal{S} \times \mathcal{A}}$. We assume



(a) Number of concurrent agents varying between 1 and 100.



(b) Number of concurrent agents varying between 100 and 800.

Figure 3: Mean regret per agent after 50 episodes on multi-armed bandit with normal rewards as a function of the number of concurrent agents.

that the Q-function can be approximated by a linear function with weight vector $\theta_h \in \mathbb{R}^d$ such that

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A},\ Q_h^*(s,a) \approx \theta_h^T \phi_h(s,a). \tag{10}$$

Randomized Least-Squares Value Iteration (RLSVI, Wen, 2014; Osband et al., 2016b) has been introduced as a computationally tractable way to approximate PSRL with linear function approximation. RLSVI uses random perturbations to approximate a posterior over value functions. RLSVI extends naturally to the concurrent setting by drawing multiple independent samples.

5.1 Coordinated RLSVI

To add a coordination mechanism to RLSVI we start by generalizing the concept of preference previously introduced. Let $f_{k,h} : [\![1, \cdots, G]\!] \times \mathcal{S} \to \mathcal{A}$ be a preference function, $f_{k,h}(g,s)$ encodes the preference of agent g at state s, episode k and timestep h. We also need the generalization of Equation 8, for $\mu \in \mathbb{R}^d, \Sigma \in \mathbb{R}^{d\times d}$

$$Z \sim \mathcal{N}(\mu, \Sigma) \iff \epsilon \sim \mathcal{N}(0, I_d),\ Z = \mu + D\epsilon, \tag{11}$$

where $D \in \mathbb{R}^{d\times d}$ is the square root of Σ, i.e $\Sigma = DD^\top$. Following Principle 1, we wish to coordinate agents when all actions in a state are equally promising. At state s, RLSVI acts according the greedy action $\arg\max_{\alpha\in\mathcal{A}} \left(\Phi\tilde{\theta}\right)(s,\alpha)$ where $\tilde{\theta}$ is a sample from the posterior. All actions in s are equally likely to be selected if we have

$$\forall a_1, a_2 \in \mathcal{A},\ \theta_h^\top \phi_{s,a_1} \overset{D}{=} \theta_h^\top \phi_{s,a_2}, \tag{12}$$

with equality in distribution and θ_h the posterior distribution over value functions learned by RLSVI. The computation of a standard normal vector is made more precise by the following lemma

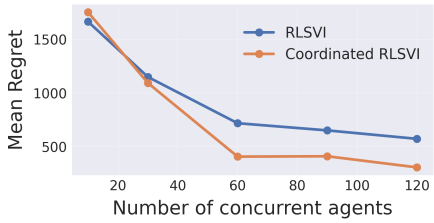
Lemma 4. *Let g be an agent in state s at timestep k. Let $\mathcal{N}(\bar{\theta}, \Sigma)$ be the posterior distribution over value functions for the current timestep, $\epsilon \sim \mathcal{N}(0, I_d)$ and $\tilde{\theta} = \bar{\theta} + D\epsilon$, where D is the square root of Σ. If all actions in s are equally likely to be the greedy action chosen by RLSVI we have, with $a = f_k(g,s)$:*

$$\begin{aligned} a \in \arg\max_{\alpha\in\mathcal{A}} \left(\Phi\tilde{\theta}\right)(s,\alpha) \text{ iff} \\ \forall \alpha \neq a,\ \epsilon^\top \psi_{s,a} \geq \epsilon^\top \psi_{s,\alpha}, \end{aligned} \tag{13}$$

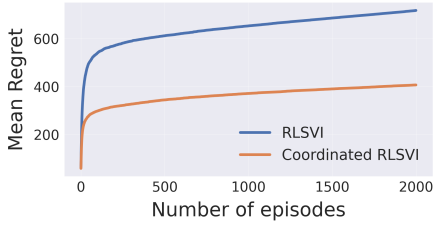
where $\psi_{s,\alpha} = \frac{D^\top \phi_{s,\alpha}}{\|D^\top \phi_{s,\alpha}\|_2}$. If $\psi_{s,a}$ is different from every other $\psi_{s,\alpha}$, the greedy action a is unique.

This lemma shows that if an agent has a preference for an action a its standard normal vector will be distributed according to a truncated Gaussian with support on $F_{s,a} = \{x \in \mathbb{R}^d | \forall \alpha \neq a,\ \psi_{s,a}^\top x \geq \psi_{s,\alpha}^\top x\}$. This result generalizes CoordTS' sampling procedure to the linear function approximation setting. Because $\{F_{s,a}\}_{a\in A}$ define a partition of $\mathbb{R}^d$, we recover the fact that when agents' preferences are marginalized standard normal vectors are normally distributed.

We are now ready to present a new coordinated algorithm that extends RLSVI. This algorithm that we call Coordinated RLSVI (CoordRLSVI) is made up of two parts. The first part describes the



(a) Average regret after 2000 episodes as function of the number of concurrent agents.



(b) Average regret over the first 2000 episodes with 60 concurrent agents.

Figure 4: Performance of RLSVI and CoordRLSVI in the "Binary Tree Environment", results are averaged over 500 seeds

update of the posterior distribution and is presented in the appendix as Algorithm 4. It assumes a prior $\mathcal{N}(0, \lambda I)$ over weights and injects Gaussian noise with variance σ^2. The only difference with the original RLSVI algorithm is that the algorithm returns the mean and covariance matrix of the posterior instead of samples. To obtain a complete exploration algorithm, action selection must be done according to the procedure specified by Algorithm 5. Algorithm 5 adds two steps to RLSVI's greedy action selection. In the first step the algorithm samples a preference for each agent at its current state. To keep track of preferences assigned during the episode a buffer $\mathcal{B}$ is created. Two states within a certain distance are considered similar and share the same entry in the buffer to monitor preferences. This distance is defined using a metric $\|\cdot\|$ and a radius $r > 0$. In the second step a standard normal vector is drawn for each agent at its current state.

5.2 Empirical evaluation

In this section we present empirical results that support the effectiveness of CoordRLSVI. Our testing environment is a full binary tree of height H as displayed by Figure 5. Each agent starts at the root s_0 then they can transition left or right at every step. All states have zero reward except for the leaves. At each leaf c the agent chooses a last action to observe a reward distributed according to $\mathcal{N}(\mu_c, \nu^2)$ with unknown mean μ_c and known variance ν^2 after which the episode ends. Agents share a common prior $\mathcal{N}(\mu_0, \Sigma_0)$ over rewards mean. In this environment coordination can be useful as agents can cooperate in order to reach individual leaves faster.

We compare RLSVI and CoordRLSVI on this environment with a one-hot encoding for features and the following hyperparameters $\lambda = 50, \sigma = 25, H = 6, \nu = 25, \Sigma_0 = \nu^2 I, \mu_0 = 0$. Results are displayed in Figures 4 and 4a, each data point was averaged over 500 seeds. The two algorithms perform similarly when the number of agents is low, though CoordRLSVI achieves a slightly higher regret because coordination leads to choosing a wider set of actions. But as the number of agents increases, CoordRLSVI average regret decreases faster. With 60 agents, CoordRLSVI mean regret is 43% smaller than RLSVI's. By coordinating agents, CoordRLSVI is faster to achieve a lower regret, highlighting the benefit of coordination for linear function approximation.

6 Discussion

Concurrent reinforcement learning presents an enticing offer; trading time with computational cost by increasing the number concurrent of agents. However, the increased computational requirements can become restrictive. Coordination can provide a solution to this problem by maximizing the efficiency of concurrent reinforcement learning algorithms. In this work we presented the first algorithms that achieve coordination by drawing correlated samples from a posterior distribution. While these algorithms performed better than existing ones, much work remains to be done to capitalize on coordinated exploration. First, it is unclear how we can mathematically model coordinated algorithms to better understand their behavior and derive sharper regret bounds. Then, moving beyond linear function approximation will be an important hurdle, yet it is where coordination will be most valuable. As concurrent reinforcement is becoming more sought after, it is in complex applications with intractable state and action spaces that it will shine the most.

REFERENCES

Karbasi Amin, Mirrokni Vahab, and Shadravan Mohammad. Parallelizing thompson sampling. In *Advances in Neural Information Processing Systems*, 2021.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21:89–96, 2008.

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9, 2018.

Marc G Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pp. 1471–1479, 2016.

Lilian Besson and Emilie Kaufmann. Multi-Player Bandits Revisited. In *Proceedings of Algorithmic Learning Theory*, pp. 56–92, 2018.

Cristian Bodnar, Adrian Li, Karol Hausman, Peter Pastor, and Mrinal Kalakrishnan. Quantile qt-opt for risk-aware vision-based robotic grasping. In *Proceedings of Robotics: Science and Systems*, 2020.

Simina Branzei and Yuval Peres. Multiplayer bandit learning, from competition to cooperation. In *Proceedings of Thirty Fourth Conference on Learning Theory*, pp. 679–723, 2021.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *Proceedings of the International Conference on Learning Representations*, 2019.

Emile Contal, David Buffoni, Alexandre Robicquet, and Nicolas Vayatis. Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 225–240. Springer, 2013.

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Advances in neural information processing systems*, 2017.

Maria Dimakopoulou and Benjamin Van Roy. Coordinated exploration in concurrent reinforcement learning. In *ICML*, 2018.

Maria Dimakopoulou, Ian Osband, and Benjamin Van Roy. Scalable coordinated exploration in concurrent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4219–4227, 2018.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Tanmay Gangwani, Qiang Liu, and Jian Peng. Learning self-imitating diverse policies. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HyxzRsR9Y7`.

Whiyoung Jung, Giseung Park, and Youngchul Sung. Population-guided parallel policy search for reinforcement learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rJeINp4KwH`.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. 2018.

Dmitry Kalashnikov, Jake Varley, Yevgen Chebotar, Ben Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv*, 2021.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabas Poczos. Parallelised bayesian optimisation via thompson sampling. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pp. 133–142, 2018.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8198–8210, 2020.

Lifeng Lai, Hai Jiang, and H Vincent Poor. Medium access in cognitive radio networks: A competitive multi-armed bandit framework. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 98–102. IEEE, 2008.

Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. Stein variational policy gradient. In *International Conference on Learning Representations*, 2017.

Muhammad A. Masood and Finale Doshi-Velez. Diversity-inducing policy gradient: Using maximum mean discrepancy to find a set of diverse policies. In *IJCAI*, 2019.

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

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pp. 4026–4034, 2016a.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 2377–2386, 2016b.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8617–8629, 2018.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the International Conference on Machine Learning*, 2017.

Jason Pazis and Ronald E. Parr. Efficient pac-optimal exploration in concurrent, continuous state mdps with delayed updates. In *AAAI*, 2016.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. In *Proceedings of The 28th Conference on Learning Theory*, pp. 1456–1456, 2015.

Daniel Russo. Worst-case regret bounds for exploration via randomized value functions. In *Advances in neural information processing systems*, 2019.

David Silver, Leonard Newnham, David Barker, Suzanne Weller, and Jason McFall. Concurrent reinforcement learning from customer interactions. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 924–932, 2013.

Malcolm Strens. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pp. 943–950, 2000.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 1933.

Ahmed Touati, Harsh Satija, Joshua Romoff, Joelle Pineau, and Pascal Vincent. Randomized value functions via multiplicative normalizing flows. *arXiv preprint arXiv:1806.02315*, 2018.

Zheng Wen. *Efficient reinforcement learning with value function generalization*. Stanford University, 2014.

Andrea Zanette, David Brandfonbrener, Matteo Pirotta, and Alessandro Lazaric. Frequentist regret bounds for randomized least-squares value iteration. In *AISTATS*, 2020.

Algorithm 4 Randomized Least-Squares Value Iteration for coordination

Input: Data $\Phi_1(s^g_{i1}, a^g_{i1}), r^g_{i1}, \cdots, \Phi_H(s^g_{iH-1}, a^g_{iH-1}), r^g_{iH} : i < k, g < G$, parameters $\lambda > 0, \sigma > 0$.
Output: $\bar{\theta}_{k1}, \Sigma_{k1}, \cdots, \bar{\theta}_{kH}, \Sigma_{kH}$
for $h = H, \cdots, 1$ **do**
  Generate regression problem $M \in \mathbb{R}^{(k*G)\times d}, b \in \mathbb{R}^{k*G}$:

$$M \leftarrow \begin{bmatrix} \Phi_h(s^1_{1h}, a^1_{1h}) \\ \vdots \\ \Phi_h(s^G_{k,h}, a^G_{k,h}) \end{bmatrix}$$

$$b_i \leftarrow \begin{cases} r^g_{ih} + \max_\alpha \left(\Phi_{h+1}\tilde{\theta}^g_{k,h+1}\right)(s^g_{i,h+1}, \alpha) & \text{if } h < H \\ \\ r^g_{ih} + r^g_{i,h+1} & \text{if } h = H \end{cases}$$

  Bayesian linear regression for the value function

$$\bar{\theta}_{kh} \leftarrow \frac{1}{\sigma^2}\Big(\frac{1}{\sigma^2}M^\top M + \frac{1}{\lambda}I\Big)^{-1} M^\top b$$

$$\Sigma_{kh} \leftarrow \Big(\frac{1}{\sigma^2}M^\top M + \frac{1}{\lambda}I\Big)^{-1}$$

  for $g = 1, \cdots, G$ **do**
    Sample $\tilde{\theta}^g_{kh} \sim \mathcal{N}(\bar{\theta}_{kh}, \Sigma_{kh})$
  end for
end for

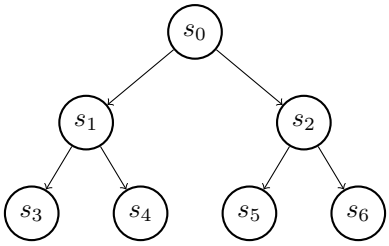


Figure 5: Full binary tree environment, H = 3

APPENDIX A RELATED WORK

Thompson Sampling. Thompson Sampling (Thompson, 1933) has enjoyed a resurgence in recent years as an efficient method for exploration in reinforcement learning (Strens, 2000; Osband et al., 2013). RLSVI (Wen, 2014; Osband et al., 2016b) is a theory-sound algorithm (Russo, 2019; Zanette et al., 2020) that was proposed as a tractable approximation of Thompson Sampling that could be applied to linear function approximation. Other work have attempted to extend the Thompson Sampling paradigm to non-linear function approximation (Osband et al., 2016a; 2018; Touati et al., 2018; Azizzadenesheli et al., 2018)

Concurrent Thompson sampling. As noted by Dimakopoulou & Van Roy (2018), Thompson Sampling based algorithms are easily scalable to the concurrent reinforcement learning setting. Kandasamy et al. (2018) proposed a parallel version of Thompson Sampling for Bayesian optimization. Amin et al. (2021) introduced batch Thompson Sampling as a way to parallelize Thompson Sampling. The regret of their algorithm scales in $O(\log T)$ batch queries as opposed to a linear scaling for a fully sequential algorithm. However it requires a dynamic batch size with a varying number of concurrent agents. This could be problematic for some real world scenarios.

Learning diverse policies. The idea of learning a diverse set of policies in reinforcement learning has been investigated in the past in the context of single or multi agent systems. Often these techniques

Algorithm 5 RLVSI with coordinated greedy action selection

Input: Feature matrices $\Phi_1, \cdots, \Phi_H$, preference function buffer $\mathcal{B} \leftarrow \{\}$, radius $r > 0$.
Output: $\tilde{\theta}_{k1}, \cdots, \tilde{\theta}_{k,H}$
for $k = 1, \cdots$ **do**
  Compute $\bar{\theta}_{k1}, \Sigma_{k1}, ..., \bar{\theta}_{k,H}, \Sigma_{kH}$ using Algorithm 4
  Observe s^g_{k1}, $\forall g$
  // Reset preferences counts
  $\mathcal{B} \leftarrow \{\}$
  for $h = 1, \cdots, H$ **do**
    Compute D_{kh} such that $\Sigma_{kh} = D_{kh} D_{kh}^\top$
    for $g = 1, \cdots, G$ **do**
      // Sample a preference for agent g at state s^g_{kh}
      if $s \leftarrow$ NearestNeighborLessThan$(s^g_{k,h}, \mathcal{B}, r)$ **then**
        $m \leftarrow \mathcal{B}[s]$
      else
        Create vector $m = 0$, with $m \in \mathbb{R}^A$
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{s^g_{kh} : m\}$
      end if
      $c \leftarrow \arg\min_\alpha m[\alpha]$
      Draw a uniformly random from c
      $m[a] = m[a] + 1$ and update $\mathcal{B}$
      // Sample a standard normal vector using rejection sampling
      $\forall \alpha \in \mathcal{A}$, $\psi_{s^g_{kh},\alpha} \leftarrow D_{kh}^\top \phi_{s^g_{kh},\alpha} / \|D_{kh}^\top \phi_{s^g_{kh},\alpha}\|_2$
      Sample $\epsilon^g \sim \mathcal{N}(0, I_d)$
      while $\exists \alpha$ such that $\epsilon^{g\top} \psi_{s^g_{kh},\alpha} > \epsilon^{g\top} \psi_{s^g_{kh},a}$ **do**
        Sample $\epsilon^g \sim \mathcal{N}(0, I_d)$
      end while
      // Draw sample and select a greedy action
      Compute $\tilde{\theta}^g_{kh} = \bar{\theta}_{kh} + D_{kh}\epsilon^g$
      Sample $a^g_{kh} \in \arg\max_{\alpha \in \mathcal{A}} \left(\Phi_h \tilde{\theta}^g_{kh}\right)(s^g_{kh}, \alpha)$
      Observe r^g_{kh} and $s^g_{k,h+1}$
    end for
  end for
end for

rely on a penalty – such as a constraint on stationary state distributions – to enforce diversity between policies (Liu et al., 2017; Conti et al., 2017; Eysenbach et al., 2018; Gangwani et al., 2019; Masood & Doshi-Velez, 2019; Jung et al., 2020; Kumar et al., 2020). These approaches usually trade off diversity with the optimality of the policies. They do not guaranty that all policies will converge to an optimal policy, or even that the set of policies will indeed be diverse when possible.

Multi-player bandits. Our setup is also related to the multi-player bandits literature (Lai et al., 2008; Besson & Kaufmann, 2018; Branzei & Peres, 2021). In this framework cooperating players select arms then they get to observe the corresponding rewards, however if two players choose the same arm they receive a zero reward instead.

APPENDIX B PROOFS

Theorem 1. *The mean Bayes regret of Algorithm 1 is bounded by*

$$\textit{MeanBayesRegret}(T, \pi, \phi) \leq \tilde{O}(HS\sqrt{\frac{AT}{G}}). \tag{6}$$

Proof. To simplify notations, we write:

- $V_h^{M^*,\pi^*} = V_h^*$
- $V_h^{M^*,\pi} = V_h^{*,\pi}$
- $V_h^{M_g^k,\pi_k^g} = V_h^{k,g}$

We have:

$$\begin{aligned}\mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\big[\mathbb{E}_{s\sim\rho_1}\big[V_1^*(s) - V_1^{*,\pi_k^g}(s)\big]\big] &= \mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\big[\mathbb{E}_{s\sim\rho_1}\big[V_1^*(s) - V_1^{k,g}(s) + V_1^{k,g}(s) - V_1^{*,\pi_k^g}(s)\big]\big] \\ &= \mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\big[\mathbb{E}_{s\sim\rho_1}\big[V_1^{k,g}(s) - V_1^{*,\pi_k^g}(s)\big]\big]\end{aligned}$$

The second equality is true since $M^{k,g}$ and M^* are identically distributed given the history $\mathcal{H}_{k1}$. Hence the average Bayesian regret can be written as:

$$\text{MeanBayesRegret}(T,K,\phi,G) = \frac{1}{G}\sum_{g=1}^{G}\mathbb{E}_{M^*\sim\phi}\big[\sum_{k=1}^{K}\mathbb{E}_{s\sim\rho_1}\big[V_1^{k,g}(s) - V_1^{*,\pi_k^g}(s)\big]\big] \tag{14}$$

This new term can then decomposed as, with $x_{k,h}^g \coloneqq (s_{kh}^g, a_{kh}^g)$

$$\begin{aligned}&\mathbb{E}_{s\sim\rho_1}\big[V_1^{k,g}(s) - V_1^{*,\pi_k^g}(s)\big] \\ &\quad= \mathbb{E}_{s\sim\rho_1}\Big[(r^{M_g^k} - r^*)(x_{k1}^g) + P^{M_g^k}(x_{k1}^g)^\top V_2^{k,g} - P^{M^*}(x_{k1}^g)^\top V_2^{*,\pi_g^k}\Big] \\ &\quad= \mathbb{E}_{s\sim\rho_1}\Big[(r^{M_g^k} - r^*)(x_{k1}^g) + (P^{M_g^k} - P^{M^*})(x_{k1}^g)^\top V_2^{k,g} + \mathbb{E}_{x_{k2}^g\sim P^{M^*}(x_{k1}^g)}(V_2^{k,g} - V_2^{*,\pi_g^k})(x_{k2}^g)\Big] \\ &\quad= \cdots \\ &\quad= \mathbb{E}_{s\sim\rho_1}\Big[\sum_{h=1}^{H}(r^{M_g^k} - r^*)(x_{kh}^g) + (P^{M_g^k} - P^{M^*})(x_{kh}^g)^\top V_h^{k,g}\Big]\end{aligned}$$

Let $\bar{r}_k$ and $\bar{P}_k$ be the empirical reward and transition at timestep k. Wede fine the confidence set $\mathbb{M}_k$ for episode k:

$$\mathbb{M}_k \coloneqq \big\{M : \forall(s,a),\ |r^M(s,a) - \bar{r}_k(s,a)| \le \beta_1^k(s,a) \text{ and} \tag{15}$$

$$\big\|P^M(\cdot|s,a) - \bar{P}_k(\cdot|s,a)\big\|_1 \le \beta_2^k(s,a)\big\}. \tag{16}$$

Where $\beta_k^1 \coloneqq \sqrt{\frac{7\log(2SAt_k/\delta)}{2\max\{1,N_{t_k}(s,a)\}}}$ and $\beta_2 \coloneqq \sqrt{\frac{14S\log(2At_k/\delta)}{\max\{1,N_{t_k}(s,a)\}}}$ are chosen so that M^* and $M^{k,g}$ belong to $\mathbb{M}_k$ with high probability and $t_k = (k-1)H + 1$

Lemma 5. *Lemma 17 from Auer et al. (2008). Let $\delta \in (0,1)$, β_k^1, β_2 chosen as previously, then we have*

$$P(M^* \notin \mathbb{M}_k) = P(M^{k,g} \notin \mathbb{M}_k) \le \frac{\delta}{15t_k^6} \tag{17}$$

$$\begin{aligned}&\frac{1}{G}\sum_{g=1}^{G}\sum_{k=1}^{K}\mathbb{E}_{\mathcal{H}_{k1}}\mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\Big[\mathbb{E}_{s\sim\rho_1}\big[V_1^*(s) - V_1^{*,\pi_k^g}(s)\big]\Big] \\ &\quad\le \frac{1}{G}\sum_{g=1}^{G}\sum_{k=1}^{K}\mathbb{E}_{\mathcal{H}_{k1}}\mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\Big[\mathbb{E}_{s\sim\rho_1}\big[V_1^*(s) - V_1^{*,\pi_k^g}(s)\big]1_{\{M^{k,g},M^*\in\mathbb{M}_k\}} + H(1_{\{M^{k,g}\notin\mathbb{M}_k\}} + 1_{\{M*\notin\mathbb{M}_k\}})\Big] \\ &\quad= \frac{1}{G}\sum_{g=1}^{G}\sum_{k=1}^{K}\mathbb{E}_{\mathcal{H}_{k1}}\mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\Big[\mathbb{E}_{s\sim\rho_1}\big[V_1^*(s) - V_1^{*,\pi_k^g}(s)\big]1_{\{M^{k,g},M^*\in\mathbb{M}_k\}}\Big] + 2H\sum_{k=1}^{K}P(M^*\notin\mathbb{M}_k)\end{aligned}$$

Because M^* and $M^{k,g}$ are identically distributed we have $1_{\{M^{k,g}\notin\mathbb{M}_k\}} = 1_{\{M*\notin\mathbb{M}_k\}} = P(M^* \notin \mathbb{M}_k)$. By choosing $\delta = \frac{1}{K}$ we have

$$2H\sum_{k=1}^{K} P(M^* \notin \mathbb{M}_k) \leq 2H$$

For the other term we have

$$\begin{aligned}
& \mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\big[\mathbb{E}_{s\sim\rho_1}\big[V_1^*(s) - V_1^{*,\pi_k^g}(s)\big]1_{\{M^{k,g},M*\in\mathbb{M}_k\}}\big] \\
= \; & \mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\Big[\Big(\mathbb{E}_{s\sim\rho_1}\sum_{h=1}^{H}(r^{M_g^k} - r^*)(x_{kh}^g) + (P^{M_g^k} - P^{M^*})(x_{kh}^g)^\top V_h^{k,g}\Big)1_{\{M^{k,g},M*\in\mathbb{M}_k\}}\Big] \\
\leq \; & \mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\Big[\sum_{h=1}^{H}\mathbb{E}_{x_{k,h}^g\sim\rho_{k,h}^g}\big[2\beta_k^1(x_{k,h}^g) + 2\beta_k^2(x_{k,h}^g)H\big]\Big]
\end{aligned}$$

where $\rho_{k,h}^g(s)$ is the probability to reach state s at depth h following policy π_k^g. We have

$$\beta_k^1(x_{k,h}^g) + \beta_k^2(x_{k,h}^g)H \leq csnt \cdot H\sqrt{\frac{S\log(SAHT)}{\max\{1, N_{t_k}(x_{k,h}^g)\}}} \tag{18}$$

Now, what remains is bounding the term

$$\frac{1}{G}\sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}_{\mathcal{H}_{k1}}\mathbb{E}_{\phi(\cdot|\mathcal{H}_{k1})}\Big[\sqrt{\frac{1}{\max\{1, N_{t_k}(x_{k,h}^g)\}}}\Big]$$

We have

$$\begin{aligned}
\frac{1}{G}\sum_{g=1}^{G}\sum_{k=1}^{K}\sum_{h=1}^{H}\Big[\sqrt{\frac{1}{\max\{1, N_{t_k}(x_{k,h}^g)\}}}\Big] &\leq \frac{1}{G}\sum_{x\in\mathcal{S}\times\mathcal{A}}\sum_{i=1}^{n_H^K(x)}\sqrt{\frac{1}{i}} \\
&\leq \frac{1}{G}\sum_{x\in\mathcal{S}\times\mathcal{A}}\int_{y=1}^{n_H^K(x)} y^{-1/2}\,dy \\
&\leq \frac{1}{G}\sum_{x\in\mathcal{S}\times\mathcal{A}}\sqrt{n_H^K(x)} \\
&\leq \frac{1}{G}\sqrt{SA}\sqrt{\sum_{x\in\mathcal{S}\times\mathcal{A}} \underset{=GT}{n_H^K(x)}} \\
&\leq \sqrt{\frac{SAT}{G}}.
\end{aligned}$$

Which gives the desired result. □

Lemma 1. *We define*

$$\mathcal{P}_k = \Big(\frac{\sum_g \mathbb{1}_{\{f_k(g)=a\}}}{G}\Big)_{a\in\mathcal{A}}, \qquad \mathcal{Q}_k = \Big(\frac{\sum_g \mathbb{1}_{\{a_k^g=a\}}}{G}\Big)_{a\in\mathcal{A}}. \tag{9}$$

$\mathcal{P}_k$ and $\mathcal{Q}_k$ correspond to the distribution of preferences and actions at timestep k. Then, Algorithm 3 maximizes the entropy of $\mathcal{P}_k$. Furthermore, when all arms have the same distribution under the posterior then the entropy of $\mathcal{Q}_k$ is also maximized.

Proof. Let $\mathcal{C} = \Big\{p \in [0,1]^A | \sum_i p_i = 1,\, \forall i,\, p_i \in [0, \frac{1}{G}, \cdots, \frac{G-1}{G}, 1]\Big\}$, $\mathcal{C}$ is the space of discrete probability distribution with values in $\{0, \frac{1}{G}, \cdots, \frac{G-1}{G}, 1\}$. Let H be the entropy function, for any distribution p, $H(p) = -\sum_i p_i \log p_i$

We define $\nu = \lfloor \frac{G}{A} \rfloor$ and $\mathcal{D} = \left\{ p \in \mathcal{C} | \forall i, p_i = \frac{\nu}{G} \text{ or } p_i = \frac{\nu+1}{G} \right\}$. It is easy to see that at each timestep the distribution of preferences sampled by Algorithm 3 is an element of $\mathcal{D}$. It is also clear that the entropy is a constant over $\mathcal{D}$.

We want to show that this constant is the maximum that can be attained for probability distributions in $\mathcal{C}$. Let $p^1, p^2 \in [0,1]^A$ be two probability distributions, we define the distance $\|\cdot\|_\infty$ by $\|p^1 - p^2\|_\infty = \max_i |p_i^1 - p_i^2|$. We define also $q = [\frac{1}{A}, \cdots, \frac{1}{A}]$ the uniform distribution over $[0,1]^A$.

The set of distributions of $\mathcal{C}$ that minimizes the distance to the uniform distribution with the distance $\|\cdot\|_\infty$ is $\mathcal{D}$

$$\mathcal{D} = \arg\min_{p \in \mathcal{C}} \|p - q\|_\infty \tag{19}$$

If it was not the case, there would be a distribution p with $p_i > \frac{\nu+1}{G}$ or $p_i < \frac{\nu}{G}$ which minimizes this distance. This is not possible because we would have $\|p - q\|_\infty > \frac{1}{G}$ and the distance between probability distributions in $\mathcal{D}$ and p is less than $\frac{1}{G}$. The solution is exactly $\mathcal{D}$ because all probability distributions in $\mathcal{D}$ have the same distance to the uniform distribution.

For any distribution $p \in \mathcal{C}$ we have

$$\begin{aligned} H(q) - H(p) &= -\sum_i q_i \log q_i + \sum_i p_i \log p_i \\ &= -\sum_i q_i \log q_i + \sum_i p_i \log q_i + \sum_i p_i (\log p_i - \log q_i) \\ &= KL(p||q) \end{aligned}$$

Because the uniform distribution maximizes the entropy over all possible distributions in $[0,1]^A$, the distribution p with maximal entropy in $\mathcal{C}$ minimizes the KL divergence $KL(p||q)$. We can conclude using Pinsker's inequality by noting that $\arg\min_{p \in \mathcal{C}} \|p - q\|_\infty = \arg\min_{p \in \mathcal{C}} KL(p||q)$.

When all arms have the same distribution under the posterior we have $\mathcal{Q}_k = \mathcal{P}_k$ and the entropy of $\mathcal{Q}_k$ is also maximized.

□

Lemma 3. *Consider a team of G agents and $N_k(F_i)$ the number of standard normal vectors sampled at timestep k that belong to subspace F_i. Then, we have $\forall i, j \in [\![1, ..., N]\!]$, $|N_k(F_i) - N_k(F_j)| \leq 1$.*

Proof. This lemma is a direct consequence of the previous lemma. Each preference $f_k(g) = a_i$ leads to a standard normal vector $\epsilon \in F_i$. We have $\sum_g \mathbb{1}_{\{f_k(g)=a\}} = \lfloor \frac{G}{A} \rfloor$ or $\sum_g \mathbb{1}_{\{f_k(g)=a\}} = \lfloor \frac{G}{A} \rfloor + 1$, so for $i \in [\![1, \cdots, N]\!]$, $N_k(F_i) = \lfloor \frac{G}{A} \rfloor$ or $\lfloor \frac{G}{A} \rfloor + 1$ which gives the desired result. □

Lemma 2. *Let X_1, X_2 be two iid random variables distributed according to $\mathcal{N}(0,1)$. Let $Z = (X, Y)$ with $X = \max(X_1, X_2)$ and $Y = \min(X_1, X_2)$. Then X and Y follow a skew normal distribution with scale $+1$ and -1 and Z follows a half-normal distribution with support on $\{(x,y) \in \mathbb{R}^2 | x \geq y\}$.*

Proof. Let f and F denote the PDF and CDF of $\mathcal{N}(0,1)$. For $x \in \mathbb{R}$

$$\begin{aligned} F_X(x) &= P(X \leq x) \\ &= P(X_1 \leq x \text{ and } X_2 \leq x) \\ &= P(X_1 \leq x) P(X_2 \leq x) \\ &= F(x)^2. \end{aligned}$$

The density function of X is now given by

$$f_X(x) = \frac{d}{dx} F_X(x) = \frac{d}{dx} F(x)^2 = 2 f(x) F(x).$$

Which is the density function of a skew normal distribution with shape 1. Similarly for $y \in \mathbb{R}$ we have

$$\begin{aligned} F_Y(y) &= P(Y \leq y) \\ &= 1 - P(Y > y) \\ &= 1 - P(X_1 > y)P(X_2 > y) \\ &= 1 - (1 - F(y))^2. \end{aligned}$$

The density function of Y is now given by

$$f_Y(y) = \frac{d}{dy} F_Y(y) = \frac{d}{dy}\big(1 - (1 - F(y))^2\big) = 2f(y)(1 - F(y)) = 2f(y)F(-y).$$

Because we have $1 - F(y) = F(-y)$. This density function corresponds to a skew normal distribution with shape -1. Finally with support on $(x, y) \in \{(x, y) \in \mathbb{R}^2 | x \geq y\}$, the CDF joint distribution is given by

$$\begin{aligned} F_{X,Y}(x, y) &= P(X \leq x, Y \leq y) \\ &= P(X \leq x) - P(Y > y, X \leq x) \\ &= F(x)^2 - P(y < X_1 \leq x, y < X_2 \leq x) \\ &= F(x)^2 - P(y < X_1 \leq x)P(y < X_2 \leq x) \\ &= F(x)^2 - \big((F(x) - F(y))\big)^2. \end{aligned}$$

And the density function is given by

$$\begin{aligned} f_{X,Y}(x, y) &= \frac{d^2}{dxdy}\big(F_{X,Y}(x, y)\big) \\ &= \frac{d}{dx}\big(2f(y)(F(x) - F(y))\big) \\ &= 2f(x)f(y). \end{aligned}$$

Which concludes the proof.

□

Lemma 4. *Let g be an agent in state s at timestep k. Let $\mathcal{N}(\bar{\theta}, \Sigma)$ be the posterior distribution over value functions for the current timestep, $\epsilon \sim \mathcal{N}(0, I_d)$ and $\tilde{\theta} = \bar{\theta} + D\epsilon$, where D is the square root of Σ. If all actions in s are equally likely to be the greedy action chosen by RLSVI we have, with $a = f_k(g, s)$:*

$$\begin{aligned} a \in \arg\max_{\alpha \in \mathcal{A}} \big(\Phi\tilde{\theta}\big)(s, \alpha) \text{ iff} \\ \forall \alpha \neq a,\ \epsilon^\top \psi_{s,a} \geq \epsilon^\top \psi_{s,\alpha}, \end{aligned} \tag{13}$$

where $\psi_{s,\alpha} = \frac{D^\top \phi_{s,\alpha}}{\|D^\top \phi_{s,\alpha}\|_2}$. If $\psi_{s,a}$ is different from every other $\psi_{s,\alpha}$, the greedy action a is unique.

Proof. For any action $a_j \in \mathcal{A}$, actions a_i and a_j having the same value under the posterior means that

$$\begin{aligned} \theta^\top \phi_{s,a_i} &\overset{D}{=} \theta^\top \phi_{s,a_j} \\ (\bar{\theta} + D\epsilon)^\top \phi_{s,a_i} &\overset{D}{=} (\bar{\theta} + D\epsilon)^\top \phi_{s,a_j}, \end{aligned}$$

with equality in distribution. Taking the expectation and variance in the last equality gives

$$\begin{aligned} \bar{\theta}^\top \phi_{s,a_i} &= \bar{\theta}^\top \phi_{s,a_j} \\ \mathrm{Var}((D\epsilon)^\top \phi_{s,a_i}) &= \mathrm{Var}((D\epsilon)^\top \phi_{s,a_j}). \end{aligned}$$

The second equation can be simplified as

$$\begin{aligned}
\mathrm{Var}((D\epsilon)^\top \phi_{s,a_i}) &= \mathrm{Var}(\epsilon^\top D^\top \phi_{s,a_i}) \\
&= \mathrm{Var}((D^\top \phi_{s,a_i})^\top \epsilon) \quad \big(\epsilon^\top (D^\top \phi_{s,a_i}) = (D^\top \phi_{s,a_i})^\top \epsilon\big) \\
&= (D^\top \phi_{s,a_i})^\top \mathrm{cov}(\epsilon)(D^\top \phi_{s,a_i}) \\
&= \phi_{s,a_i}^\top D D^\top \phi_{s,a_i} \\
&= \|D^\top \phi_{s,a_i}\|_2^2.
\end{aligned}$$

We get

$$\|D^\top \phi_{s,a_i}\|_2^2 = \|D^\top \phi_{s,a_j}\|_2^2$$

Hence we have

$$\begin{aligned}
a_i = \arg\max_{\alpha \in \mathcal{A}} \big(\Phi\tilde{\theta}\big)(s,\alpha) &\iff \tilde{\theta} \sim \mathcal{N}(\bar{\theta}, \Sigma), \forall a_j \in \mathcal{A},\ \tilde{\theta}^\top \phi_{s,a_i} > \tilde{\theta}^\top \phi_{s,a_j} \\
&\iff \epsilon \sim \mathcal{N}(0, I_d), \forall a_j \in \mathcal{A},\ (\bar{\theta} + D\epsilon)^\top \phi_{s,a_i} > (\bar{\theta} + D\epsilon)^\top \phi_{s,a_j} \\
&\iff \epsilon \sim \mathcal{N}(0, I_d), \forall a_j \in \mathcal{A},\ \frac{(D\epsilon)^\top \phi_{s,a_i}}{\|D^\top \phi_{s,a_i}\|_2} > \frac{(D\epsilon)^\top \phi_{s,a_j}}{\|D^\top \phi_{s,a_j}\|_2} \\
&\iff \epsilon \sim \mathcal{N}(0, I_d), \forall a_j \in \mathcal{A},\ \epsilon^\top \frac{D^\top \phi_{s,a_i}}{\|D^\top \phi_{s,a_i}\|_2} > \epsilon^\top \frac{D^\top \phi_{s,a_j}}{\|D^\top \phi_{s,a_j}\|_2} \\
&\iff \epsilon \sim \mathcal{N}(0, I_d), \forall a_j \in \mathcal{A},\ \epsilon^\top \psi_{s,a_i} > \epsilon^\top \psi_{s,a_j}
\end{aligned}$$

□