Regret Bounds and Reinforcement Learning Exploration of EXP-based Algorithms

Anonymous Author(s)
Affiliation
Address
email

Abstract

EXP-based algorithms are often used for exploration in non-stochastic bandit problems assuming rewards are bounded. Motivated by the recent advancements in reinforcement learning with rewards of any scale, we propose a new algorithm, namely EXP4.P, by modifying EXP4 and establish its regret upper bounds in both bounded and unbounded sub-Gaussian contextual bandits. The unbounded reward result also holds for a revised version of EXP3.P. Moreover, we provide a lower bound on regret that suggests no sublinear regret can be achieved given short time horizon. Unbounded rewards pose challenges as the regret cannot be limited by the number of trials, and choosing suboptimal arms may result in infinite regret. We also extend EXP4.P from bandit to reinforcement learning to incentivize exploration by multiple agents given black-box rewards. The resulting algorithm has been tested on hard-to-explore games and it shows an improvement on exploration compared to state-of-the-art.

1 Introduction

Multi-armed bandit (MAB) is to maximize cumulative reward of a player throughout a bandit game by choosing different arms at each time step. It is also equivalent to minimizing the regret defined as the difference between the best rewards that can be achieved and the actual reward gained by the player. Formally, given time horizon T, in time step $t \leq T$ the player chooses one arm a_t among K arms, receives $r^t_{a_t}$ among rewards $r^t = (r^t_1, r^t_2, \ldots, r^t_K)$, and maximizes the total reward $\sum_{t=1}^T r^t_{a_t}$ or minimizes the regret. Traditionally, there are two classic versions of non-stochastic bandits: Adversarial and Contextual. For adversarial MAB, rewards of the K arms r^t can be chosen arbitrarily by adversaries at step t. When the adversary is a context-dependent reward generator, it boils down to contextual bandits. Contextual bandit is a variant of MAB by adding context or state space S and a different regret definition. At time step t, the player has context $s_t \in S$ and rewards r^t follow $f(\mu(s_t))$ where f is any distribution and $\mu(s_t)$ is the mean vector that depends on state s_t.

Computationally efficient and with abundant theoretical analyses are the EXP-type MAB algorithms. Specifically, the regret of EXP3.P for adversarial bandit achieves optimality both in the expected and high probability sense. In EXP3.P, each arm has a trust coefficient (weight). The player samples each arm with probability being the sum of its normalized weights and a bias term, receives reward of the sampled arm and exponentially updates the weights based on the corresponding reward estimates. It achieves the regret of the order $O(\sqrt{T})$ in a high probability sense. To incorporate the context information in contextual bandits, a variant of EXP-type algorithms is proposed as EXP4 [3]. In EXP4, there are any number of experts. Each expert has a sample rule over actions (arms) and a trust coefficient. The player samples according to the weighted average of experts' sample rules and updates the weights respectively. Then the regret is defined by comparing the actual reward with the reward that can be achieved by the best expert instead of by the best arm. The expectation of regret is proven to be optimal for contextual bandit. Independently, [11] propose a modification of EXP4 that

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

achieves high probability guarantee, which, however, requires changes in the reward estimates. A high probability regret has not yet been studied in its original form of EXP4.

Recently, contextual bandit has been further aligned with Reinforcement Learning (RL) where state and reward transitions follow a Markov Decision Process (MDP) represented by transition kernel $P(s_{t+1}, r^t | a_t, s_t)$. A key challenge in RL is the trade-off between exploration and exploitation. Exploration is to encourage the player to try new arms in bandit or new actions in RL to understand the game better. It helps to plan for the future, but with the sacrifice of potentially lowering the current reward. Exploitation aims to exploit currently known states and arms to maximize the current reward, but it potentially prevents the player to gain more information to increase future reward. To maximize the cumulative reward, the player needs to learn the game by exploration, while guaranteeing current reward by exploitation.

How to incentivize exploration in RL has been a main focus in RL. Since RL is built on bandits, it is natural to extend bandit techniques to RL and UCB is such a success. UCB [2] motivates count-based exploration [18] in RL and the subsequent Pseudo-Count exploration [4], though it is initially developed for stochastic bandits. Another line of work on RL exploration is based on deep learning techniques. Using deep neural networks to keep track of the Q-values by means of Q-networks in RL is called DQN [9]. This combination of deep learning and RL has shown great success. ϵ-greedy in [10] is a simple exploration technique based on DQN. Besides ϵ-greedy, intrinsic model exploration computes intrinsic rewards that directly measure and thereby incentivizing exploration when added to extrinsic (actual) rewards of RL, e.g. DORA [6] and [17]. Random Network Distillation (RND) [5] is a more recent suggestion relying on a fixed target network. A drawback of RND is its local focus without global exploration. EXP-type algorithms in contextual bandits work by integrating arbitrary experts and hence providing exploration possibilities for RL, which, however, has not yet been studied. Furthermore, the existing EXP4 or its variant cannot be directly adapted to RL. It is worth noting that EXP-type algorithms are optimal under the assumption that $0 \leq r_i^t \leq 1$ for any arm i and step t. The uniformly bounded assumption is crucial in the proof of regret bounds for existing EXP-type algorithms. It requires the rewards to be scalable with the knowledge of a uniform bound for all rewards in all states or context vectors. Nevertheless, reward in RL can be unbounded and unscalable in real-world scenarios, which violates the bounded assumption. Examples include navigation tasks, where the reward is unbounded for each step that brings the agent closer to the goal, and racing tasks, where the reward is the distance covered by the agent. The counterpart of bandit algorithms in the unbounded or scale-free case remained unexplored, unite the work herein and it necessitates a new algorithm based on EXP3.P and EXP4.

In this paper, we are the first to propose a new algorithm, EXP4.P based on EXP4 without changing the reward estimates. We show its optimal regret holds with high probability and in expectation for contextual bandits with possibly unbounded (scale-free) rewards. The regret bounds for unbounded bandits studied herein are significantly different from prior works. Compared to the high probability version in [11], our algorithm only requires one parameter, is consistent with the reward estimate in EXP4 and EXP3.P, removes the reward assumption of $[0, 1]$, and generalizes to the expected regret. The proof extension to the unbounded case is non-trivial since it requires several deep results from information theory and probability, by first establishing a high probability regret in the bounded case with exponential terms and then using the Randemacher complexity theory to capture the dynamics of arm selection in the unbounded case. Combining all these together is very technical and requires new ideas. As a by-product, the analysis can be applied to EXP3.P to deliver a similar result for bandits without expert advice. The upper bound for unbounded bandits requires T to be sufficiently large, i.e. unbounded rewards may lead to extremely large regret without enough exploration, which is computationally expensive in an RL setting. We herein provide a worst-case analysis implying no sublinear regret can be achieved below an instance-specific minimal T, by our brand new construction of instances. Precisely, we derive lower bounds of order $\Omega(T)$ for certain fixed T and upper bounds of order $O^*(\sqrt{T})$ for T being large enough. The question of bounds for any value of T remains open.

Given the challenges of RL context where rewards are possibly unbounded or unrescalable which have not been addressed by existing methods, we combine the proposed scale-free EXP-type algorithms with deep RL. To this end, we extend the new EXP4.P to RL that allows for general experts by generalizing the concept of experts to be any RL algorithms. Here experts improve local policies with the underlying Markov process and exponential weights are assigned to the experts to produce a global optimal policy. This is the first RL algorithm using several experts enabling global exploration, where the overall performance is comparable to the best model even if we do not know which one is the best beforehand, and thereby achieving model selections [8]. To address the issue of EXP4's inefficiency with a large number of experts, we combine EXP4-RL with at least one state-of-the-art expert algorithm for improved efficiency and performance thus having only a few experts. Focusing on DQN, in the computational study we focus on two agents consisting of RND and ϵ-greedy DQN. We implement the EXP4-RL algorithm on hard-to-explore RL games Montezuma's Revenge and Mountain Car and compare it with the benchmark RND [5]. The numerical results show that the algorithm gains more exploration than RND and it gains the ability of global exploration by avoiding local maxima of RND. Its total reward also increases with training. Overall, our algorithm improves exploration on the benchmark games.

The main contributions are as follows. We introduce sub-Gaussian bandits with the unique aspect and challenge of unbounded and scale-free rewards both in contextual bandits and MAB when EXP-based algorithms are considered. We propose a new EXP4.P algorithm based on EXP4 and EXP3.P and analytically establish its optimal regret both in unbounded and bounded cases. Unbounded rewards and contextual setting pose non-trivial challenges in the analyses. We also provide the very first regret lower bound in such a case that indicates a threshold of T for sublinear regret, by constructing a novel family of Gaussian bandits. We also provide the very first extension of EXP4.P to RL exploration using multiple agents and show its superior performance on two hard-to-explore RL games.

A literature review is provided in Section 2. Then in Section 3 we develop a new algorithm EXP4.P by modifying EXP4, and exhibit its regret bounds for contextual bandits and that of the EXP3.P algorithm for unbounded MAB, and lower bounds. Section 4 discusses the EXP4.P algorithm for RL exploration. Finally, in Section 5, we present numerical results related to the proposed algorithm.

2 Literature Review

The importance of exploration in RL is well understood. Count-based exploration in RL is such a success with the UCB technique. [18] develop Bellman value iteration $V(s) = \max_a \hat{R}(s,a) + \gamma E[V(s')] + \beta N(s,a)^{-\frac{1}{2}}$, where $N(s,a)$ is the number of visits to (s,a) for state s and action a. Value $N(s,a)^{-\frac{1}{2}}$ is positively correlated with curiosity of (s,a) and encourages exploration. This method is limited to tableau model-based MDP for small state spaces. While [4] introduce Pseudo-Count exploration for non-tableau MDP with density models, it is hard to model. However, UCB achieves optimality if bandits are stochastic and may suffer linear regret otherwise [21]. The work on CORRAL in [1] considers a group of bandit algorithms, but it requires a parameter search in the parameter space. In the RL setting, such updates are inefficient and do not fit the dynamic RL setting. EXP-type algorithms for non-stochastic bandits can generalize to RL with fewer assumptions about the statistics of rewards, which have not yet been studied. In conjunction with DQN, ϵ-greedy in [10] is a simple exploration technique using DQN. Besides ϵ-greedy, intrinsic model exploration computes intrinsic rewards by the accuracy of a model trained on experiences. Intrinsic rewards directly measure and incentivize exploration if added to actual rewards of RL, e.g. see [6, 17, 5]. Random Network Distillation(RND) in [5] define it as $e(s',a) = \|\hat{f}(s') - f(s')\|_2^2$ where $\hat{f}$ is a parametric model and f is a randomly initialized but fixed model. Here $e(s',a)$, independent of the transition, only depends on state s' and drives RND to outperform others on Montezuma's Revenge. None of these algorithms use several experts which is a significant departure from our work.

Along the line of work on regret analyses focusing on EXP-type algorithms, [3] first introduce EXP3.P for bounded adversarial MAB and EXP4 for bounded contextual bandits. For the EXP3.P algorithm, an upper bound on regret of order $O(\sqrt{T})$ holds with high probability and in expectation, which has no gap with the lower bound and hence it establishes that EXP3.P is optimal. EXP4 is optimal for contextual bandits in the sense that its expected regret is $O(\sqrt{T})$. Then [11] extend it to a high probability counterpart by modifying the reward estimates. These regret bounds are invalid for bandits with unbounded support. Though [16] demonstrate a regret bound $O(\sqrt{T \cdot \gamma_T})$ for noisy Gaussian process bandits, information gain γ_T is not well-defined in a noiseless setting. For noiseless Gaussian bandits, [7] show both the optimal lower and upper bounds on regret, but the regret definition is not consistent with [3]. We tackle these problems by establishing an upper bound of order $O^*(\sqrt{T})$ on regret 1) with high probability for bounded contextual bandit and 2) for sub-Gaussian bandit both in expectation and with high probability.

3 Regret Bounds

We first introduce notations. Let T be the time horizon. For bounded bandits, at step t, $0 < t \leq T$ rewards r^t can be chosen arbitrarily under the condition that $-1 \leq r^t \leq 1$. For unbounded bandits, let rewards r^t follow multi-variate distribution $f_t(\mu, \Sigma)$ where $\mu = (\mu_1, \mu_2, \ldots, \mu_K)$ is the mean vector and $\Sigma = (a_{ij})_{i,j\in\{1,\ldots,K\}}$ is the covariance matrix of the K arms and f_t is the density. We specify f_t to be non-degenerate sub-Gaussian for analyses on light-tailed distributions where $\min_j a_{j,j} > 0$. A random variable X is σ^2-sub-Gaussian if for any $t > 0$, the tail probability satisfies $P(|X| > t) \leq Be^{-\sigma^2 t^2}$ where B is a positive constant.

The player receives reward $y_t = r^t_{a_t}$ by pulling arm a_t. The regret is defined as $R_T = \max_j \sum_{t=1}^T r^t_j - \sum_{t=1}^T y_t$ in adversarial bandits that depends on realizations of rewards. For contextual bandits with experts, besides the above let N be the number of experts and c_t be the context information. We denote the reward of expert i by $G_i = \sum_{t=1}^T z_i(t) = \sum_{t=1}^T \xi_i(t)^T x(t)$, where $x(t) = r^t$ and $\xi_i(t) = (\xi_i^1(t), \ldots, \xi_i^K(t))$ is the probability vector of expert i. Then regret is defined as $R_T = \max_i G_i - \sum_{t=1}^T y_t$, which is with respect to the best expert, rather than the best arm in MAB. This is reasonable since a uniform optimal arm is a special expert assigning probability 1 to the optimal arm throughout the game and experts can potentially perform better and admit higher rewards. This coincides with our generalization of EXP4.P to RL where the experts can be well-trained neural networks. We follow established definitions of pseudo regret $R'_T = T \cdot \max_k \mu_k - \sum_t E[y_t]$ and $\sum_{t=1}^T \max_i \sum_{j=1}^K \xi_i^j(t)\mu_j - \sum_t E[y_t]$ in adversarial and contextual bandits, respectively.

3.1 Contextual Bandits and EXP4.P Algorithm

For contextual bandits, [3] give the EXP4 algorithm and prove its expected regret to be optimal under the bounded assumption on rewards and under the assumption that a uniform expert is always included, where by uniform expert we refer to an expert that always assigns equal probability to each arm. Our goal is to extend EXP4 to RL where rewards are often unbounded, such as several games in OpenAI gym, for which the theoretical guarantee of EXP4 may be absent. To this end, herein we propose a new Algorithm, named EXP4.P, as a variant of EXP4. Its effectiveness is two-fold. First, we show that EXP4.P has an optimal regret with high probability in the bounded case and consequently, we claim that the regret of EXP4.P is still optimal given unbounded bandits. All the proof are in the Appendix under the aforementioned assumption on experts. Second, it is successfully extended to RL where it achieves computational improvements.

3.1.1 EXP4.P Algorithm

Algorithm 1 EXP4.P

Initialization: Weights $w_i(1) = \exp\left(\frac{\alpha\gamma}{3K}\sqrt{NT}\right)$, $i \in \{1, 2, \ldots, N\}$ for $\alpha > 0$ and $\gamma \in (0, 1)$;
for $t = 1, 2, \ldots, T$ **do**
  Get probability vectors $\xi_1(t), \ldots, \xi_N(t)$ of arms from experts where $\xi_i(t) = (\xi_i^j(t))_j$;
  For any $j = 1, 2, \ldots, K$, set $p_j(t) = (1-\gamma)\sum_{i=1}^N \dfrac{w_i(t) \cdot \xi_i^j(t)}{\sum_{j=1}^N w_j(t)} + \frac{\gamma}{K}$;
  Choose i_t randomly according to the distribution $p_1(t), \ldots, p_K(t)$;
  Receive reward $r_{i_t}(t) = x_{i_t}(t)$;
  For any $j = 1, \ldots, K$, set $\hat{x}_j(t) = \frac{r_j(t)}{p_j(t)} \cdot \mathbb{1}_{j=i_t}$;
  Set $\hat{x}(t) = (\hat{x}_j(t))_j$;
  For any $i = 1, \ldots, N$, set
  $\hat{z}_i(t) = \xi_i(t)^T \hat{x}(t)$ and $w_i(t+1) = w_i(t)\exp\left(\frac{\gamma}{3K}\left(\hat{z}_i(t) + \dfrac{\alpha}{\left(\frac{w_i(t)}{\sum_{j=1}^N w_j(t)} + \frac{\gamma}{K}\right)\sqrt{NT}}\right)\right)$;
end for

Our proposed EXP4.P is shown as Algorithm 1. The main modifications compared to EXP4 lie in the update and the initialization of trust coefficients of experts as highlighted. The upper bound of the confidence interval of the reward estimate is added to the update rule for each expert, in the spirit of EXP3.P (see Algorithm 2) and removing the need of changing the reward estimate. However, this term and initialization of EXP4.P are quite different from that in EXP3.P for MAB.

3.1.2 Bounded Rewards

Borrowing the ideas of [3], we claim EXP4.P has an optimal sublinear regret with high probability by first establishing two lemmas presented in Appendix. The main theorem is as follows. We assume that the expert family includes a uniform expert, which is also assumed in the analysis of EXP4 in [3].

Theorem 1. *Let $0 \le r^t \le 1$ for every t. For any fixed time horizon $T > 0$, for all $K,\ N \ge 2$ and for any $1 > \delta > 0$, $\gamma = \sqrt{\frac{3K \ln N}{T(\frac{2N}{3}+1)}} \le \frac{1}{2}$, $\alpha = 2\sqrt{K \ln \frac{NT}{\delta}}$, we have that with probability at least $1-\delta$, $R_T \le 2\sqrt{3KT\left(\frac{2N}{3}+1\right)\ln N} + 4K\sqrt{KNT\ln\left(\frac{NT}{\delta}\right)} + 8NK\ln\left(\frac{NT}{\delta}\right)$.*

Theorem 1 implies $R_T \le O^*(\sqrt{T})$. The regret bound does depend on N. In practice the number of experts is small compared to the time horizon and the independence among experts makes parallelism a possibility. Note that $\gamma < \frac{1}{2}$ for large enough T. The proof of Theorem 1 essentially relies on the convergence of the reward estimators, similar to that in [3]. However, the objectives are different from [3], since our estimations and update of trust coefficients in EXP4.P are for experts, instead of EXP3.P for arms. This characterize the relationships among EXP4.P estimates and the actual value of experts' rewards and the total rewards gained by EXP4.P and brings non-trivial challenges.

3.1.3 Unbounded Rewards

We proceed to show optimal regret bounds of EXP4.P for unbounded contextual bandit. Again, a uniform expert is assumed to be included in the expert family. Surprisingly, we report that the analysis can be adapted to the existing EXP3.P in next section, which leads to optimal regret in MAB under no bounded assumption which is also a new result.

Theorem 2. *For sub-Gaussian bandits, any time horizon T, for any $0 < \eta < 1$, $0 < \delta < 1$ and γ, α as in Theorem 1, with probability at least $(1-\delta)(1-\eta)^T$, EXP4.P has regret $R_T \le 4\Delta(\eta)\left(2\sqrt{3KT\left(\frac{2N}{3}+1\right)\ln N}\right) + 4\Delta(\eta)\left(4K\sqrt{KNT\ln\left(\frac{NT}{\delta}\right)} + 8NK\ln\left(\frac{NT}{\delta}\right)\right)$ where $\Delta(\eta)$ is determined by $\int_{-\Delta}^{\Delta}\cdots\int_{-\Delta}^{\Delta} f\left(x_1,\ldots,x_K\right)dx_1\ldots dx_K = 1-\eta$ which yields $\Delta(\eta)$ of $O(\frac{1}{a}\log\frac{1}{\eta})$.*

In the proof of Theorem 2, we first perform truncation of the rewards of sub-Gaussian bandits by dividing the rewards to a bounded part and unbounded tail. For the bounded part, we directly apply the upper bound on regret of EXP4.P presented in Theorem 1 and conclude with the regret upper bound of order $O(\Delta(\eta)\sqrt{T})$. Since a sub-Gaussian distribution is a light-tailed distribution we can control the probability of the tail, i.e. the unbounded part, which leads to the overall result.

The dependence of the bound on Δ can be removed by considering large enough T as stated next.

Theorem 3. *For sub-Gaussian bandits, for any $a > 2$, $0 < \delta < 1$, and γ, α as in Theorem 1, EXP4.P has regret $R_T \le \log(1/\delta)O^*(\sqrt{T})$ with probability $(1-\delta)\cdot(1-\frac{1}{T^a})^T$.*

Note that the constant term in $O^*(\cdot)$ depends on a. The above theorems deal with R_T; an upper bound on pseudo regret or expected regret is established next. It is easy to verify by the Jensen's inequality that $R'_T \le E[R_T]$ and thus it suffices to obtain an upper bound on $E[R_T]$.

For bounded bandits, the upper bound for $E[R_T]$ is of the same order as R_T which follows by a simple argument. For sub-Gaussian bandits, establishing an upper bound on $E[R_T]$ or R'_T based on R_T requires more work. We show an upper bound on $E[R_T]$ by using certain inequalities, limit theories, and Rademacher complexity. To this end, the main result reads as follows.

Theorem 4. *The regret of EXP4.P for sub-Gaussian bandits satisfies $R'_T \le E\left[R_T\right] \le O^*(\sqrt{T})$ under the assumptions stated in Theorem 3.*

3.2 MAB and EXP3.P Algorithm

In this section, we establish upper bounds on regret in MAB given a high probability regret bound achieved by EXP3.P in [3]. We revisit EXP3.P and analyze its regret in unbounded scenarios in line with EXP4.P. Formally, we show that EXP3.P achieves regret of order $O^*(\sqrt{T})$ in sub-Gaussian MAB, with respect to R_T, $E[R_T]$ and R'_T. The results are summarized as follows.

Theorem 5. *For sub-Gaussian MAB, any T, for any $0 < \eta, \delta < 1$, $\gamma = 2\sqrt{\frac{3K\ln K}{5T}}$, $\alpha = 2\sqrt{\ln\frac{NT}{\delta}}$, EXP3.P has regret $R_T \leq 4\Delta(\eta)\cdot(\sqrt{KT\log(\frac{KT}{\delta})} + 4\sqrt{\frac{5}{3}KT\log K} + 8\log(\frac{KT}{\delta}))$ with probability $(1-\delta)(1-\eta)^T$ where $\Delta(\eta) = O(\frac{1}{a}\log\frac{1}{\eta})$, i.e. $\int_{-\Delta}^{\Delta}\cdots\int_{-\Delta}^{\Delta} f\Big(x_1,\ldots,x_K\Big)dx_1\ldots dx_K = 1-\eta$.*

To proof Theorem 5, we again do truncation. We apply the bounded result of EXP3.P in [3] and achieve a regret upper bound of order $O(\Delta(\eta)\sqrt{T})$. The proof is similar to the proof of Theorem 2 for EXP4.P.

Similarly, we remove the dependence of the bound on Δ in Theorem 6 and claim a bound on the expected regret for sufficiently large T in Theorem 7.

Algorithm 2 EXP3.P

Initialization: Weights $w_i(1) = \exp\left(\frac{\alpha\gamma}{3}\sqrt{\frac{T}{K}}\right)$, $i \in \{1, 2, \ldots, K\}$ for $\alpha > 0$ and $\gamma \in (0, 1)$;
for $t = 1, 2, \ldots, T$ **do**
  For any $i = 1, 2, \ldots, K$, set $p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{K} w_j(t)} + \frac{\gamma}{K}$;
  Choose i_t randomly according to the distribution $p_1(t), \ldots, p_K(t)$;
  Receive reward $r_{i_t}(t)$;
  For $1 \leq j \leq K$, set $\hat{x}_j(t) = \frac{r_j(t)}{p_j(t)} \cdot \mathbb{1}_{j=i_t}$ and $w_j(t+1) = w_j(t)\exp\frac{\gamma}{3K}(\hat{x}_j(t) + \frac{\alpha}{p_j(t)\sqrt{KT}})$;
end for

Theorem 6. *For sub-Gaussian MAB, for $a > 2$, $0 < \delta < 1$, and γ, α as in Theorem 5, EXP3.P has regret $R_T \leq \log(1/\delta)O^*(\sqrt{T})$ with probability $(1-\delta)\cdot(1-\frac{1}{T^a})^T$.*

Theorem 7. *The regret of EXP3.P in sub-Gaussian MAB satisfies $R'_T \leq E\left[R_T\right] \leq O^*(\sqrt{T})$ with the same assumptions as in Theorem 6.*

3.3 Lower Bounds on Regret

Algorithms can suffer extremely large regret without enough exploration when playing unbounded bandits given small T. To argue that our bounds on regret are not loose, we derive a lower bound on the regret for sub-Gaussian bandits that essentially suggests that no sublinear regret can be achieved if T is less than an instance-dependent bound. The main technique is to construct instances that have certain regret, no matter what strategies are deployed. We need the following assumption.

Assumption 1 There are two types of arms with general K with one type being superior (S is the set of superior arms) and the other being inferior (I is the set of inferior arms). Let $1-q, q$ be the proportions of the superior and inferior arms, respectively which is known to the adversary and clearly $0 \leq q \leq 1$. The arms in S are indistinguishable and so are those in I. The first pull of the player has two steps. First the player selects an inferior or superior set of arms based on $P(S) = 1-q, P(I) = q$ and once a set is selected, the corresponding reward of an arm from the selected set is received.

An interesting special case of Assumption 1 is the case of two arms and $q = 1/2$. In this case, the player has no prior knowledge and in the first pull chooses an arm uniformly at random.

The lower bound is defined as $R_L(T) = \inf\sup R'_T$, where, first, $\inf$ is taken among all the strategies and then $\sup$ is among all Gaussian MAB. The following is the main result for lower bounds based on inferior arms being distributed as $\mathcal{N}(0,1)$ and superior as $\mathcal{N}(\mu,1)$ with $\mu > 0$.

Theorem 8. *In Gaussian MAB under Assumption 1, for any $q \geq 1/3$ we have $R_L(T) \geq (q-\epsilon)\cdot\mu\cdot T$, where μ has to satisfy $G(q,\mu) < q$ with ϵ and T determined by $G(q,\mu) < \epsilon < q, T \leq \frac{\epsilon - G(q,\mu)}{(1-q)\cdot\int\left|e^{-\frac{x^2}{2}} - e^{-\frac{(x-\mu)^2}{2}}\right|} + 2$ where $G(q,\mu)$ is $\max\{\int|qe^{-\frac{x^2}{2}} - (1-q)e^{-\frac{(x-\mu)^2}{2}}|dx,$ $\int|(1-q)e^{-\frac{x^2}{2}} - qe^{-\frac{(x-\mu)^2}{2}}|dx\}$.*

To prove Theorem 8, we construct a special subset of Gaussian MAB with equal variances and zero covariances. On these instances we find a unique way to explicitly represent any policy. This builds a connection between abstract policies and this concrete mathematical representation. Then we show that pseudo regret R'_T must be greater than certain values no matter what policies are deployed, which indicates a regret lower bound on this subset of instances.

Feasibility of the aforementioned conditions is established in the following theorem.

Theorem 9. *In Gaussian MAB under Assumption 1, for any $q \geq 1/3$, there exist μ and $\epsilon, \epsilon < \mu$ such that $R_L(T) \geq (q - \epsilon) \cdot \mu \cdot T$.*

The following result with two arms and equal probability in the first pull deals with general MAB. It shows that for any fixed $\mu > 0$ there is a minimum T and instances of MAB so that no algorithm can achieve sublinear regret. Table 1 (see Appendix) exhibits how the threshold of T varies with μ.

Theorem 10. *For general MAB under Assumption 1 with $K = 2, q = 1/2$, we have that $R_L(T) \geq \frac{T \cdot \mu}{4}$ holds for any distributions f_0 for the arms in I and f_1 for the arms in S with $\int |f_1 - f_0| > 0$ (possibly with unbounded support), for any $\mu > 0$ and T satisfying $T \leq \frac{1}{2 \cdot \int |f_0 - f_1|} + 1$.*

4 EXP4.P Algorithm for RL

EXP4 has shown effectiveness in contextual bandits with statistical validity. Therefore, in this section, we extend EXP4.P to RL in Algorithm 3 where rewards are assumed to be nonnegative.

The player has experts that are represented by deep Q-networks trained by RL algorithms (there is a one to one correspondence between the experts and Q-networks). Each expert also has a trust coefficient. Trust coefficients are also updated exponentially based on the reward estimates as in EXP4.P. At each step of one episode, the player samples an expert (Q-network) with probability that is proportional to the weighted average of expert's trust coefficients. Then ϵ-greedy DQN is applied on the chosen Q-network. Here different from EXP4.P, the player needs to store all the interaction tuples in the experience buffer since RL is a MDP. After one episode, the player trains all Q-networks with the experience buffer and uses the trained networks as experts for the next episode. The basic

Algorithm 3 EXP4-RL

Initialization: Trust coefficients $w_k = 1$ for any $k \in \{1, \dots, E\}$, $E =$ number of experts (Q-networks), $K =$ number of actions, $\Delta, \epsilon, \eta > 0$ and temperature $z, \tau > 0$, $n_r = -\infty$ (an upper bound on reward);
while True **do**
  Initialize episode by setting s_0
  for $i = 1, 2, \dots, T$(length of episode) **do**
    Observe state s_i;
    Let probability of Q_k-network be $\rho_k = (1 - \eta) \frac{w_k}{\sum_{j=1}^{E} w_j} + \frac{\eta}{E}$;
    Sample network $\bar{k}$ according to $\{\rho_k\}_k$;
    For $Q_{\bar{k}}$-network, use ϵ-greedy to sample an action: $a^* = \arg\max_a Q_{\bar{k}}(s_i, a), j \in \{1, 2, \dots, K\}, \pi_j = (1 - \epsilon) \cdot \mathbb{1}_{j=a^*} + \frac{\epsilon}{K-1} \cdot \mathbb{1}_{j \neq a^*}$;
    Sample action a_i based on π;
    Interact with the environment to receive reward r_i and next state s_{i+1};
    $n_r = \max\{r_i, n_r\}$;
    Update the trust coefficient w_k of each Q_k-network as follows: $P_k = \epsilon\text{-greedy}(Q_k), \hat{x}_{kj} = 1 - \frac{\mathbb{1}_{j=a^*}}{P_{kj} + \Delta}(1 - \frac{r_i}{n_r}), \forall j, y_k = E[\hat{x}_{kj}], w_k = w_k \cdot e^{\frac{y_k}{z}}$;
    Store (s_i, a_i, r_i, s_{i+1}) in experience replay buffer B;
  end for
  Update each expert's Q_k-network from buffer B
end while

idea is the same as in EXP4.P by using the experts that give advice vectors with deep Q-networks. It is a combination of deep neural networks with EXP4.P updates. From a different point of view, we can also view it as an ensemble in classification [20], by treating Q-networks as ensembles in RL. While general experts can be used, these are natural in a DQN framework. In our implementation and experiments we use two experts, thus $E = 2$ with two Q-networks. The first one is based on RND [5] while the second one is a simple DQN. To this end, in the algorithm before storing to the buffer, we also record $c_r^i = ||\hat{f}(s_i) - f(s_i)||^2$, the RND intrinsic reward as in [5]. This value is then added to the 4-tuple pushed to B. When updating Q_1 corresponding to RND at the end of an iteration in the algorithm, by using $r_j + c_r^j$ we modify the Q_1-network and by using c_r^j an update to $\hat{f}$ is executed. Network Q_2 pertaining to ϵ-greedy is updated directly by using r_j. Intuitively, Algorithm 3 circumvents RND's drawback with the total exploration guided by two experts with EXP4.P updated trust coefficients. When the RND expert drives high exploration, its trust coefficient leads to a high total exploration. When it has low exploration, the second expert DQN should have a high one and it incentivizes the total exploration accordingly. Trust coefficients are updated by reward estimates iteratively as in EXP4.P, so they keep track of the long-term performance of experts and then guide the total exploration globally. These dynamics of EXP4.P combined with intrinsic rewards guarantee global exploration. The experimental results exhibited in the next section verify this intuition regarding exploration behind Algorithm 3.

We point out that potentially more general RL algorithms based on Q-factors can be used, e.g., boostrapped DQN [13], random prioritized DQN [12] or adaptive ϵ-greedy VDBE [19] are a possibility. Furthermore, experts in EXP4 can even be policy networks trained by PPO [15] instead of DQN for exploration. A recommendation is to have a good enough expert and a small number of experts.

5 Computational Study

As a numerical demonstration of the superior performance and exploration incentive of Algorithm 3, we show the improvements on baselines on two hard-to-explore RL games, Mountain Car and Montezuma's Revenge. More precisely, we present that the real reward on Mountain Car improves significantly by Algorithm 3 in Section 5.1. Then we implement Algorithm 3 on Montezuma's Revenge and show the growing and remarkable improvement of exploration in Section 5.2. Intrinsic reward $c_r^i = ||\hat{f}(s_i) - f(s_i)||^2$ given by intrinsic model $\hat{f}$ represents the exploration of RND in [5] as introduced in Sections 2 and 4. We use the same criterion for evaluating exploration performance of our algorithm and RND herein. RND incentivizes local exploration with the single step intrinsic reward but with the absence of global exploration.

5.1 Mountain Car

In this part, we summarize the experimental results of Algorithm 3 on Mountain Car, a classical control RL game. This game has very sparse positive rewards, which brings the necessity and hardness of exploration. Blog post [14] shows that RND based on DQN improves the performance of traditional DQN, since RND has intrinsic reward to incentivize exploration. We use RND on DQN from [14] as the baseline and show the real reward improvement of Algorithm 3, which supports the intuition and superiority of the algorithm.

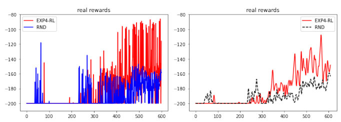
The comparison between Algorithm 3 and RND is presented in Figure 1. Here the x-axis is the epoch number and the y-axis is the cumulative reward of that epoch. Figure 1a shows the raw data comparison between EXP4-RL and RND. We observe that though at first RND has several spikes exceeding those of EXP4-RL, EXP4-RL has much higher rewards than RND after 300 epochs. Overall, the relative difference of areas under the curve (AUC) is 4.9% for EXP4-RL over RND, which indicates the significant improvement of our algorithm. This improvement is better illustrated in Figure 1b with the smoothed reward values. Here there is a notable difference between EXP4-RL and RND. Note that the maximum reward hit by EXP4-RL is -86 and the one by RND is -118, which additionally demonstrates our improvement on RND.

We conclude that Algorithm 3 performs better than the RND baseline and that the improvement increases at the later training stage. Exploration brought by Algorithm 3 gains real reward on this hard-to-explore Mountain Car, compared to the RND counterpart (without the DQN expert). The power of our algorithm can be enhanced by adopting more complex experts, not limited to only DQN.

5.2 Montezuma's Revenge and Pure exploration setting

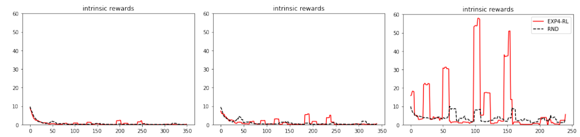
In this section, we show the experimental details of Algorithm 3 on Montezuma's Revenge, another notoriously hard-to-explore RL game. The benchmark on Montezuma's Revenge is RND based on DQN which achieves a reward of zero in our environment (the PPO algorithm reported in [5] has reward 8,000 with many more computing resources; we ran the PPO-based RND with 10 parallel environments and 800 epochs to observe that the reward is also 0), which indicates that DQN has room for improvement regarding exploration.

To this end, we first implement the DQN-version RND (called simply RND hereafter) on Montezuma's Revenge as our benchmark by replacing the PPO with DQN. Then we implement Algorithm 3 with



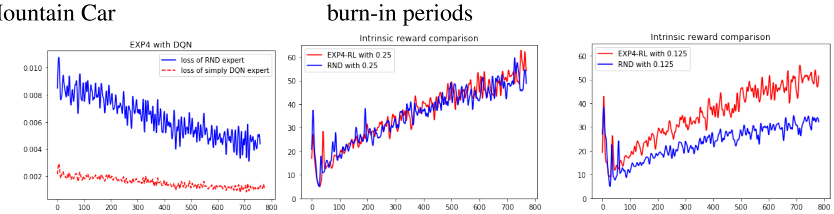
(a) original (b) smooth

Figure 1: The performance of Algorithm 3 and RND measured by the epoch-wise reward on Mountain Car



(a) small (b) medium (c) large

Figure 2: The performance of Algorithm 3 and RND measured by intrinsic reward without parallel environments with three different burn-in periods



(a) Q-network losses with 0.25 update (b) Intrinsic reward after smoothing with 0.25 update (c) Intrinsic reward after smoothing with 0.125

Figure 3: The performance of Algorithm 3 and RND with 10 parallel environments and with RND update probability 0.25 and 0.125, measured by loss and intrinsic reward.

two experts as aforementioned. Our computing environment allows at most 10 parallel environments. In subsequent figures the x-axis always corresponds to the number of epochs. RND update probability is the proportion of experience that are used for training the intrinsic model $\hat{f}$ [5].

A comparison between Algorithm 3 (EXP4-RL) and RND without parallel environments (the update probability is 100% since it is a single environment) is shown in Figure 2 with the emphasis on exploration by means of the intrinsic reward. We use 3 different numbers of burn-in periods (58, 68, 167 burn-in epochs) to remove the initial training steps, which is common in Gibbs sampling. Overall EXP4-RL outperforms RND with many significant spikes in the intrinsic rewards. The larger the number of burn-in periods is, the more significant is the dominance of EXP4-RL over RND. EXP4-RL has much higher exploration than RND at some epochs and stays close to RND at other epochs. At some epochs, EXP4-RL even has 6 times higher exploration. The relative difference in the areas under the curves are 6.9%, 17.0%, 146.0%, respectively, which quantifies the much better performance of EXP4-RL.

We next compare EXP4-RL and RND with 10 parallel environments and different RND update probabilities in Figure 3. The experiences are generated by the 10 parallel environments.

Figure 3a shows that both experts in EXP4-RL are learning with decreasing losses of their Q-networks. The drop is steeper for the RND expert but it starts with a higher loss. With RND update probability 0.25 in Figure 3b we observe that EXP4-RL and RND are very close when RND exhibits high exploration. When RND is at its local minima, EXP4-RL outperforms it. Usually these local minima are driven by sticking to local maxima and then training the model intensively at local maxima, typical of the RND local exploration behavior. EXP4-RL improves on RND as training progresses, e.g. the improvement after 550 epochs is higher than the one between epochs 250 and 550. In terms for AUC, this is expressed by 1.6% and 3.5%, respectively. Overall, EXP4-RL improves RND local minima of exploration, keeps high exploration of RND and induces a smoother global exploration.

With the update probability of 0.125 in Figure 3c, EXP4-RL almost always outperforms RND with a notable difference. The improvement also increases with epochs and is dramatically larger at RND's local minima. These local minima appear more frequently in training of RND, so our improvement is more significant as well as crucial. The relative AUC improvement is 49.4%. The excellent performance in Figure 3c additionally shows that EXP4-RL improves RND with global exploration by improving local minima of RND or not staying at local maxima.

Overall, with either 0.25 or 0.125, EXP4-RL incentivizes global exploration on RND by not getting stuck in local exploration maxima and outperforms RND exploration aggressively. With 0.125 the improvement with respect to RND is more significant and steady. This experimental evidence verifies our intuition behind EXP4-RL and provides excellent support for it. With experts being more advanced RL exploration algorithms, e.g. DORA, EXP4-RL can bring additional possibilities.

References

[1] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[4] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016.

[5] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

[6] L. Fox, L. Choshen, and Y. Loewenstein. Dora the explorer: directed outreaching reinforcement action-selection. In *International Conference on Learning Representations*, 2018.

[7] S. Grünewälder, J. Y. Audibert, M. Opper, and J. Shawe-Taylor. Regret bounds for gaussian process bandit problems. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 273–280, 2010.

[8] X. Liu, F. Xia, R. L. Stevens, and Y. Chen. Cost-effective online contextual model selection. *arXiv preprint arXiv:2207.06030*, 2022.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[11] G. Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. *Advances in Neural Information Processing Systems*, 28, 2015.

[12] I. Osband, J. Aslanides, and A. Cassirer. Randomized prior functions for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8617–8629, 2018.

[13] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.

[14] O. Rivlin. Mountaincar_dqn_rnd. `https://github.com/orrivlin/MountainCar_DQN_RND`, 2019.

[15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[17] B. C. Stadie, S. Levine, and P. Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[18] A. L. Strehl and M. L. Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[19] M. Tokic. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *Annual Conference on Artificial Intelligence*, pages 203–210. Springer, 2010.

[20] R. Xia, C. Zong, and S. Li. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181(6):1138–1152, 2011.

[21] S. Zuo. Near optimal adversarial attack on UCB bandits. *arXiv preprint arXiv:2008.09312*, 2020.