Autonomous Exploration for Navigating in MDPs Using Blackbox RL Algorithms

Pratik Gajane[1] , **Peter Auer**[2] , **Ronald Ortner**[2]
[1]Eindhoven University of Technology
[2]Montanuniversität Leoben
p.gajane@tue.nl, auer@unileoben.ac.at, ronald.ortner@unileoben.ac.at

Abstract

We consider the problem of navigating in a Markov decision process where extrinsic rewards are either absent or ignored. In this setting, the objective is to learn policies to reach all the states that are reachable within a given number of steps (in expectation) from a starting state. We introduce a novel meta-algorithm which can use any online reinforcement learning algorithm (with appropriate regret guarantees) as a black-box. Our algorithm demonstrates a method for transforming the output of online algorithms to a batch setting. We prove an upper bound on the sample complexity of our algorithm in terms of the regret bound of the used black-box RL algorithm. Furthermore, we provide experimental results to validate the effectiveness of our algorithm and correctness of our theoretical results.

1 Introduction

The ability to efficiently explore the environment remains key to sample-efficient reinforcement learning (RL). In settings where rewards are absent or sparse, the exploration must be autonomous, i.e., it cannot be guided by reward maximization. Furthermore, many works have argued for a learning approach in which the agent undergoes an extended developmental period during which reusable skills are autonomously learned that will be useful for a wide range of challenges later (e.g., [Kaplan and Oudeyer, 2003; Weng *et al.*, 2001]). In this article, we focus on learning to navigate in an unknown environment using such an approach.

Following [Lim and Auer, 2012], we consider a Markov decision process (MDP) equipped with at most countably many states and finitely many actions including a reset action which brings the agent back to some initial state. No extrinsic rewards are given and the state-transition probabilities are assumed to be stationary. The goal is to minimize the number of steps required by the agent to learn to reliably navigate to all reachable states. Since the number of states is unbounded, the agent is given as input a "radius" L such that it needs to consider all the states that are reachable within L steps (precise definitions will follow in the next section). This framework is particularly suitable when the task is to explore a large-scale environment and the learner only has enough resources to observe a small part of it by autonomous exploration. In such scenarios, it is imperative that the learner first solves simpler intrinsic goal-oriented tasks prescribed by the given number of steps L. This model could also be used in scenarios where the goal is to learn the transition operator directly.

1.1 Related Work

Similar problems have been considered in various research communities under the name of learning using intrinsic motivation/reward, curiosity-driven learning, automatic goal generation etc. Owing to the space restrictions, a necessarily incomplete list of these works include [Schmidhuber, 2010; Singh *et al.*, 2004; Singh *et al.*, 2010; Oudeyer and Kaplan, 2007; Baranes and Oudeyer, 2009; Lopes *et al.*, 2012; Gottlieb *et al.*, 2013; Houthooft *et al.*, 2016; Achiam and Sastry, 2017; Ostrovski *et al.*, 2017; Pathak *et al.*, 2017; Haber *et al.*, 2018; Burda *et al.*, 2019; Azar *et al.*, 2019; Hazan *et al.*, 2019; Florensa *et al.*, 2018]. Recently, [Ecoffet *et al.*, 2019; Ecoffet *et al.*, 2020] have proposed a family of algorithms for exploration when rewards are sparse or deceptive with experimental validation for the performance of their algorithms. Our approach could be applicable in scenarios similar to the ones considered in [Ecoffet *et al.*, 2019; Ecoffet *et al.*, 2020] who also exploit the same principle employed in our article – remember promising states and first return to such states before intentionally exploring.

[Gajane *et al.*, 2019] consider a variant of the problem at hand where the transition probabilities can change abruptly. Another line of work is presented by [Tarbouriech *et al.*, 2020] in which the authors consider a slightly modified goal (see [Tarbouriech *et al.*, 2020, Definition 5] for more details). Recently, [Cai *et al.*, 2022] proved a lower bound, based on the lower bound of UCRL2 [Jaksch *et al.*, 2010], for the considered problem. Another relevant work is the "reward-free RL" paradigm introduced by [Jin *et al.*, 2020]: following its exploration phase, their algorithm is able to compute near-optimal policies for a collection of given reward functions. Although related to the problem at hand at a conceptual level, their approach remains limited to the finite-horizon setting. While we focus on showing how a general RL algorithm can be used for the task of exploration, another line of work (e.g., [Agarwal *et al.*, 2020]) studies how exploration algorithms (i.e., *policy cover*) can be turned into a general RL algorithm.

Our proposed algorithm shares some ideas with "online-to-batch" conversion methods [Littlestone, 1989; Cesa-Bianchi *et al.*, 2006]. It has been showed that one can obtain expected risk bounds for a hypothesis drawn randomly among those generated from online learning algorithms [Helmbold and Warmuth, 1995; Freund and Schapire, 1999; Cesa-Bianchi *et al.*, 1997]. [Shalev-Shwartz, 2012] prove that if we have an online learning algorithm that is guaranteed to have low regret, then the cost of such a random hypothesis is close to the optimal cost.

1.2 Main Contribution

We show that any RL algorithm with sublinear regret T^{α}, $\alpha < 1$, can be converted into an exploration algorithm. To the best of our knowledge, our work provides the inaugural algorithm to convert any RL algorithm with sublinear regret into an exploration algorithm with suitable guarantees on its sample complexity. This formally verifies the intuition that any small regret algorithm needs to explore its environment, at least implicitly. This also shows that RL algorithms that take advantage of a particular transition structure of an environment, resulting in improved regret bounds, can be converted into a corresponding exploration algorithm for this environment. Our black-box approach of using an RL algorithm as a subroutine leads to the generality of our proposed solution, which is one its main strengths. The structure of the proposed algorithm and its analysis may also serve as a worthy addition to the literature of online-to-batch conversion methods.

2 Problem Setting

In this section, we present the problem setting first introduced by [Lim and Auer, 2012] [1]. We consider a discrete-time Markov decision process $\mathcal{M}$ with no external rewards. We assume a countable, possibly infinite state space $\mathcal{S}$ and a finite action space $\mathcal{A}$. Upon executing an action a in state s, the environment moves to the next state s' selected randomly according to the unknown transition probabilities $P(s'|s,a)$.

The learning agent is expected to solve the autonomous exploration problem in which the goal is to find a policy for each state reachable from a starting state. In the following, we assume (without loss of generality) the starting state s_0 to be fixed, and hence it will be omitted from any notation.

Definition 1 (Navigation time)**.** *For any policy π, let $\tau_\pi(s)$ be the expected number of steps before reaching s for the first time when executing policy π starting from s_0.*

We further set $\tau^*(s) := \min_\pi \tau_\pi(s)$. The learner will be given a number $L > 0$ and we may naively demand that it finds all the states reachable in at most L steps:

Definition 2 (L-reachable states)**.** *We let $\mathcal{S}_L$ denote the set of all the states reachable in at most L steps i.e.,*

$$\mathcal{S}_L := \{s \in \mathcal{S} : \tau^*(s) \leq L\}.$$

Since the state space might be infinite, a learner could wander off in some direction or get stuck without being able to return to the starting state. To exclude this possibility, we make Assumption 1.

Assumption 1. *In every state s, there is a designated* RESET *action available, such that $P(s_0|s, \text{RESET}) = 1$.*

Definition 3 (Policy on $\mathcal{S}' \subset \mathcal{S}$)**.** *We define a* policy π on $\mathcal{S}' \subset \mathcal{S}$ *to be a policy with $\pi(s) = \text{RESET}$ for any $s \notin \mathcal{S}'$.*

[Lim and Auer, 2012] show that, in general, efficient learning to discover all the states in S_L is not possible. Rather, we require the learners to discover only the *incrementally reachable states*, $\mathcal{S}_L^{\rightarrow}$.

Definition 4 (Incrementally reachable states)**.** *Let $\prec$ be some partial order on $\mathcal{S}$. The set $\mathcal{S}_L^{\prec}$ of states reachable in L steps with respect to $\prec$, is defined inductively as follows:*

- $s_0 \in \mathcal{S}_L^{\prec}$,
- *if there is a policy π on $\{s' \in \mathcal{S}_L^{\prec} : s' \prec s\}$ with $\tau_\pi(s) \leq L$, then $s \in \mathcal{S}_L^{\prec}$.*

We define the set $\mathcal{S}_L^{\rightarrow}$ of states incrementally reachable in L steps with respect to some partial order to be $\mathcal{S}_L^{\rightarrow} := \bigcup_{\prec} \mathcal{S}_L^{\prec}$, where the union is over all possible partial orders.

As [Lim and Auer, 2012], we are interested in the number of exploration steps needed to be able to navigate to incrementally reachable states from s_0 efficiently. Thus, given parameters L and ϵ, a suitable exploration algorithm will be able to determine

- a set $\mathcal{K} \supseteq \mathcal{S}_L^{\rightarrow}$, and
- for every $s \in \mathcal{K}$, a policy π_s with $\tau_{\pi_s}(s) \leq (1+\epsilon)L$,

after a certain number of exploration steps dependent on L and ϵ.

3 Algorithm and Main Result

In this section, we present our proposed algorithm called META-EXPLORE (given in Algorithm 1) and an upper bound on its number of exploration steps. The motivation for using a black-box approach in META-EXPLORE is that it provides the generality of converting an arbitrary online RL algorithm into an exploration algorithm. The main idea of META-EXPLORE is to consider reaching a particular state as a sub-problem that is solved by using an arbitrary online RL algorithm $\mathfrak{A}$ with regret guarantees (e.g. UCRL2 from [Jaksch *et al.*, 2010], REGAL from [Bartlett and Tewari, 2009]). To reach a particular state, several *hypotheses* (defined in Section 3.1 below) are formed using the black-box RL algorithm $\mathfrak{A}$.

META-EXPLORE proceeds in *rounds*. In each round, it evaluates a *target state* to examine if a $(1+\epsilon)L$-step policy for that state can be found, for a given threshold ϵ. At the end of a particular round, if the algorithm determines (with high confidence) that a $(1+\epsilon)L$-step policy for the chosen target state has been found, the round is deemed a success round, otherwise it is called a failure round. At the end of a successful round, the chosen target state is added to the set of "known" states $\mathcal{K}$. On the other hand, at the end of a failure round, the algorithm is said to have rejected the target state. The algorithm maintains another set $\mathcal{U}$, called the set of candidate states i.e., states that are potential members of $\mathcal{S}_L^{\rightarrow}$. Note that, in any round, the algorithm tries to find a policy on $\mathcal{K}$ at that time in order to reach the target state.

[1] Readers may consult the list of symbols given in the extended version of this article to quickly look up the notation.

3.1 Major Steps of the Algorithm

Each round consists of three steps – state discovery, choice of target state, and target state evaluation.

State Discovery

Whenever the algorithm discovers a new known state s_{new}, it explores the neighborhood of s_{new} to add to the set of candidate states $\mathcal{U}$. That is, in s_{new} every action $a \in A$ is sampled $\left\lceil L \log \frac{8AL|\mathcal{K}|^2}{\delta} \right\rceil$ times. By definition of a known state, the algorithm has a $(1+\epsilon)L$-step policy $\pi_{s_{\text{new}}}$ to reach s_{new} from the starting state s_0. To sample any action $a \in A$ in s_{new}, the algorithm first resets to s_0 and then uses $\pi_{s_{\text{new}}}$ to reach s_{new}. Thus, sampling each action once requires on average $(1+\epsilon)L + 1$ steps at most. Any neighboring states of s_{new} which are not already in $\mathcal{K}$ are added to $\mathcal{U}$. The algorithm also keeps hold of the neighboring states discovered by each known state. Every time a new state becomes known, all the neighboring states $\{s\}$ of the previously known states such that $s \notin \mathcal{K}$ are also added to the set of candidate states $\mathcal{U}$. Note that a neighboring state is added to $\mathcal{U}$ even if it has been previously rejected by the algorithm. This is done to ensure that if a target state $s_r \in \mathcal{S}_L^{\rightarrow}$ is erroneously rejected by the algorithm because the preceding states from some partial order (see Definition 4) were not in $\mathcal{K}$ at that time, s_r will be considered as a target state again (this reasoning is explained in detail in Section 4.5 below).

Choice of Target State

The target state for the current round is chosen arbitrarily without replacement from the set of candidate states $\mathcal{U}$. The algorithm stops when $\mathcal{U}$ is empty.

Target State Evaluation

This step forms the crux of the algorithm. Consider a round r with target state $\bar{s}$. Let us define an *environmental episode* (or, simply an *episode*) as a series of time steps beginning at the initial state s_0 and ending when $\bar{s}$ is reached. An online RL algorithm $\mathfrak{A}$ (with given regret guarantees) is run on the *induced* MDP $\mathcal{M}_{\bar{s}}$ for Λ environmental episodes, where Λ is as defined in Step 3 in Algorithm 1.

Definition 5 (Induced MDP)**.** *In the induced MDP $\mathcal{M}_{\bar{s}}$, all the actions in state $\bar{s}$ suffer loss* 0 *and lead back to the initial state. All the states $\{s | s \notin \mathcal{K} \wedge s \neq \bar{s}\}$ are merged into a single auxiliary state at which only the* RESET *action is possible suffering loss* 1*. Actions in all the other states behave the same as in the original MDP and suffer loss* 1*.*

Thus, minimizing the total loss in $\mathcal{M}_{\bar{s}}$ is equivalent to minimizing the number of steps to reach the target state $\bar{s}$.

At the beginning of each episode, the history of $\mathfrak{A}$ in the current round is recorded. The recorded history comprises the state-action-state transition counts from the current round.

Definition 6 (Hypothesis)**.** *A hypothesis is a run of the RL algorithm $\mathfrak{A}$ from a particular history point. This means that $\mathfrak{A}$ uses the history up to this point to determine its behavior.*

The number of time steps spent in episode i are recorded in T_i^r and let $\Gamma := \left\lceil \left(1 + \frac{1}{\epsilon}\right) L \right\rceil$. Let

$$\hat{p}_r := \frac{\sum_{i=1}^{\Lambda} \mathbb{1}_{\{T_i^r > \Gamma\}}}{\Lambda}, \tag{1}$$

Algorithm 1 META-EXPLORE

```
Input: A confidence parameter $\delta \in (0,1)$, an error threshold $\epsilon > 0$, $L \geq 1$, the initial state $s_0$ and an algorithm $\mathfrak{A}$ with a regret bound of $B(\#States, \#Actions) \cdot T^\alpha \cdot D^\beta$.
Output: A set of reachable states $\mathcal{K}$ and corresponding policies $\pi_{\bar{s}}$ for all $\bar{s} \in \mathcal{K}$.
Initialization: Initialize $s_{\text{new}} \leftarrow s_0$, the set of candidate states $\mathcal{U} \leftarrow \{\}$ and the set of known states $\mathcal{K} \leftarrow \{\}$.
Set $\epsilon \leftarrow \frac{\min(1,\epsilon)}{8}$.
In each round $r = 1, 2, \ldots$:
  1. State Discovery: If $s_{\text{new}} \notin \mathcal{K}$, add $s_{\text{new}}$ to $\mathcal{K}$ and then sample each action $a \in \mathcal{A}$ in $s_{\text{new}}$ for $\left\lceil L \log \frac{8AL|\mathcal{K}|^2}{\delta} \right\rceil$ times. Add any neighboring state $\notin \mathcal{K}$ to the set of candidate states $\mathcal{U}$. Furthermore, all the neighboring states $\{s\}$ of the previously known states such that $s \notin \mathcal{K}$ are also added to the set of candidate states $\mathcal{U}$. Here, we add a neighboring state to $\mathcal{U}$ even if it has been previously rejected by the algorithm.
  2. Choice of Target State: Stop the algorithm if $\mathcal{U}$ is empty. Otherwise choose an arbitrary candidate state from $\mathcal{U}$ as the target state $\bar{s}$ and $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\bar{s}\}$.
  3. Target State Evaluation: Run $\mathfrak{A}$ with confidence parameter $\delta_r := \frac{\delta}{4\pi^2 |\mathcal{K}|^3 AL \log \frac{8AL|\mathcal{K}|^2}{\delta}}$ on the induced MDP $\mathcal{M}_{\bar{s}}$ for
        $\Lambda := \left[B(|\mathcal{K}|, A)(1+3\epsilon)^{\alpha+\beta} L^{\alpha+\beta-1}\right]^{\frac{1}{1-\alpha}} \cdot \frac{2^{\max(4,\beta/1-\alpha)}}{\epsilon^{\max(4,1/1-\alpha)}} \cdot \log\left(\frac{1}{\delta_r}\right)$
     environmental episodes. Note that an environmental episode in $\mathcal{M}_{\bar{s}}$ begins at the initial state $s_0$ and is only completed when the target state $\bar{s}$ is reached. At the beginning of each environmental episode, store the history of $\mathfrak{A}$ in the current round $r$.
     If $\Lambda$ environmental episodes are not completed in $T_r := (1+3\epsilon)L\Lambda$ time steps, then the current round $r$ is stopped with it being considered a failure round; the state $\bar{s}$ is said to be rejected by the algorithm; and the algorithm proceeds to the next round.
     Otherwise, at the end of $\Lambda$ environmental episodes perform the following check. Let $\Gamma := \left\lceil \left(1 + \frac{1}{\epsilon}\right) L \right\rceil$. Let $T_i^r$ be the number of time steps spent in the environmental episode $i$. Let
        $\hat{p}_r := \frac{\sum_{i=1}^{\Lambda} \mathbb{1}_{\{T_i^r > \Gamma\}}}{\Lambda}$.
     If $\left(\frac{(1+5\epsilon)L + (\hat{p}_r + \epsilon)}{1 - (\hat{p}_r + \epsilon)} \leq (1+8\epsilon)L\right)$
         The current round is a success round.
         Associate the $\Lambda$ history points of the current round with the state $\bar{s}$.
         Set $s_{\text{new}} \leftarrow \bar{s}$.
     Else
         The state $\bar{s}$ is said to be rejected by the algorithm and the current round is deemed a failure round.
     Proceed to the next round.
```

i.e., the fraction of the total Λ episodes that failed to end in Γ time steps. At the end of Λ environmental episodes, a performance check is carried out. If

$$\frac{(1+5\epsilon)L \ + \ (\hat{p}_r+\epsilon)}{1 \ - \ (\hat{p}_r+\epsilon)} \leq (1+8\epsilon)L, \tag{2}$$

then the current round is declared as a success round and a new round will begin with the target state $\bar{s}$ added to $\mathcal{K}$. This performance check is to ensure that the number of times $\mathfrak{A}$ failed to reach the target state $\bar{s}$ within Γ time steps is low. To prevent expending too many time steps on unpromising target states, if Λ environmental episodes are not completed in T_r (defined in Step 3 in Algorithm 1) time steps, then the current round is a failure round and a new round will begin.

3.2 Description of the Output Policy

On the completion of the algorithm, the output is composed of $\mathcal{K}$ and the corresponding Λ history points for each state $s \in \mathcal{K}$. The output policy π_s for a state $s \in \mathcal{K}$ is as follows:

1. Draw a history point (with replacement) from the Λ history points associated with s uniformly at random.
2. Start running the algorithm $\mathfrak{A}$ from the history point chosen in the previous step.
3. If s is not reached in Γ steps, execute the RESET action and go back to step 1.

3.3 Performance Guarantee

The following theorem constitutes our main result.

Theorem 1. *If* META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$ *with a regret upper bound of* $B(\#States, \#Actions) \cdot T^{\alpha} \cdot D^{\beta}$*, then with probability* $1-\delta$*, it*

(a) discovers a set of states $\mathcal{K} \supseteq S_L^{\rightarrow}$*, such that* $S := |\mathcal{K}| \leq |\mathcal{S}_{(1+\epsilon)L}^{\rightarrow}|$*,*

(b) terminates after $\tilde{O}\left(\frac{S^2 A \cdot [B(S,A)]^{\frac{1}{1-\alpha}} \cdot L^{2+\frac{\alpha+\beta-1}{1-\alpha}}}{\epsilon^{\max\left(4, \frac{1}{1-\alpha}\right)}}\right)$ *exploration steps* [2]*,*

(c) for each $s \in \mathcal{K}$*, outputs a policy* π_s *with* $\tau_{\pi_s}(s) \leq (1+\epsilon)L$*,*

where D *is the diameter as defined by [Jaksch* et al.*, 2010, Definition 1] and* $B(\#States, \#Actions)$ *is some function of the number of states and the number of actions.*

The number of exploration steps stated in the theorem could be seen as the sample complexity of META-EXPLORE.

3.4 Relation to Existing Work

[Lim and Auer, 2012] introduced this exploration problem and provided a specific algorithm whereas [Tarbouriech *et al.*, 2020] consider a variant of this problem with small branching (i.e. a small set of successor states for each action).

Disregarding log factors, the sample complexity in [Lim and Auer, 2012] is SAL^3/ϵ^3 for $L(1+\epsilon)$ reachability, and in [Tarbouriech *et al.*, 2020] it is $SAGL^5/\epsilon^2$ for $L+\epsilon$ reachability, where G is the branching factor. To make the results comparable, we set $\epsilon = L\epsilon$, which gives $SAGL^3/\epsilon^2$ for [Tarbouriech *et al.*, 2020]. Thus, [Tarbouriech *et al.*, 2020] give a better dependency on ϵ for a small branching factor, but note that G could be as large as S.

We can instantiate META-EXPLORE with UCRL2 [Jaksch *et al.*, 2010] which is also the basis for the algorithm in [Lim and Auer, 2012]. The regret bound for UCRL2 is $\tilde{O}(DS\sqrt{AT})$, which gives the sample complexity bound of $\tilde{O}(S^4A^3L^3/\epsilon^4)$ for META-EXPLORE when using UCRL2 as a black-box subroutine. Alternatively, we can instantiate META-EXPLORE with UCRL2b [Fruit *et al.*, 2020] which has the regret bound of $\tilde{O}\sqrt{DSGAT}$. The sample complexity bound of META-EXPLORE using UCRL2b is $\tilde{O}(S^3GA^2L^2/\epsilon^4)$.

As seen above, in terms of L, the sample complexity bound of META-EXPLORE using UCRL2b is better than that of either [Lim and Auer, 2012] or [Tarbouriech *et al.*, 2020]. However, in terms of S and A, the sample complexity bound for META-EXPLORE using either UCRL2 or UCRL2b is more than that of [Lim and Auer, 2012], [Tarbouriech *et al.*, 2020] and [Cai *et al.*, 2022]. This is mainly caused by additional exploration due to the use of a black-box algorithm. It is worthwhile to note that the main strength of our approach is generality of converting any RL algorithm with sublinear regret into an exploration algorithm. By the generality of our conversion, it is reasonable to expect that our meta-algorithm using black-box RL algorithms can not achieve the sample complexity of specifically optimized algorithms.

4 Analysis

First, we provide a brief road-map of the proof of Theorem 1. To prove (a), we prove, with high probability, that none of the states in $S_L^{\rightarrow}$ is "missed". To prove (b), it suffices to prove a high probability upper bound on the total number of rounds, as the number of time steps in any round is upper-bounded by $(1+3\epsilon)L\Lambda$. Finally, to prove (c), we show that any state in $\mathcal{K}$ satisfies a bound on the navigation time of its corresponding output policy.

In the following, we see that the below statements hold with high probability:

- If there is a neighboring state s that is reachable with probability at least $1/L$ from a known state, then s will be among the candidate states $\mathcal{U}$ after state discovery.
- If a state $\bar{s}$ reachable in L steps (in expectation) using a policy on $\mathcal{K}$ (at the beginning of the round) is selected as a target state, then $\bar{s}$ it will be added to $\mathcal{K}$ at the end of the round.
- If a state $\bar{s}$ is added to $\mathcal{K}$, then the corresponding output policy reaches $\bar{s}$ in $(1+\epsilon)L$ steps (in expectation).

While our analysis uses some results from [Lim and Auer, 2012], there are major difficulties resulting from the black-box RL algorithm: there is no notion of the most promising state, exploration needs to be organized differently, and – most importantly – individual runs of the black-box algorithm need to be stitched together to obtain policies for the

[2] The notation $\tilde{O}$ ignores logarithmic factors.

reachable states. The stitching requires a completely different approach in the analysis.

4.1 Decomposition into Episodes and Hypotheses

In the following, we consider a fixed successful round r. Note that although some of the values considered subsequently (such as $\mathcal{K}$, Λ, or the target state $\bar{s}$) depend on r, this dependence is not reflected in the notation. For $i = 1, \ldots, \Lambda$, let $\mathcal{H}_{i-1}$ be the run of the algorithm $\mathfrak{A}$ from the i^{th} history point (i.e. $\mathcal{H}_{i-1}$ is the policy run in episode i).

Let the cumulative loss of policy π after T steps in MDP $\mathcal{M}$ with initial state s be denoted by $\mathcal{L}(\mathcal{M}, \pi, s, T)$. Since both the MDP $\mathcal{M}_{\bar{s}}$ and the starting state s_0 are fixed for the considered round r, we use $\mathcal{L}(\pi, T)$ in place of $\mathcal{L}(\mathcal{M}_{\bar{s}}, \pi, s_0, T)$ wherever it is clear from the context.

The value $\frac{1}{T}\mathbb{E}[\mathcal{L}(\pi, T)]$ is the expected average loss up to step T. Let the average loss of policy π be defined as

$$\nu(\pi) \coloneqq \lim_{T\to\infty} \frac{1}{T}\mathbb{E}[\mathcal{L}(\pi, T)]. \tag{3}$$

Let ν^* be the optimal average loss. Then as per the (high-probability) regret bound of $\mathfrak{A}$,

$$\mathcal{L}(\mathfrak{A}, T) - T\nu^* \le B(\#States, \#Actions) \cdot T^\alpha \cdot D^\beta$$

with probability at least $1 - \delta$ where δ is the confidence parameter. Such high-probability regret bounds (e.g., [Jaksch *et al.*, 2010, Theorem 2]) also contain a log term in δ. For ease of exposition, we choose to ignore the log term in the regret bound of $\mathfrak{A}$ as it will only contribute a log term in the sample complexity bound for META-EXPLORE and the claimed sample complexity bound in Theorem 1 is in terms of $\tilde{O}$.

Let the cumulative loss of the hypothesis $\mathcal{H}_{i-1}$ during the environmental episode i be $\mathcal{L}_i^r$. Let us also define $\mathcal{L}_{i,\Gamma}^r$ as the cumulative loss in episode i before reaching either the target state or $\Gamma \coloneqq \left\lceil \left(1 + \frac{1}{\epsilon}\right) L \right\rceil$ time steps. Then

$$\mathcal{L}_{i,\Gamma}^r = \begin{cases} \mathcal{L}_i^r, & \text{if } \mathcal{L}_i^r \le \Gamma, \\ \Gamma & \text{otherwise.} \end{cases} \tag{4}$$

4.2 Number of Steps before Reaching either the Target State or Γ Time Steps

Let $T_{\pi,\Gamma}(\bar{s})$ be the random number of steps taken by policy π before reaching either the target state $\bar{s}$ or Γ time steps starting from the initial state s_0.

For $i = 1, 2, \ldots, \Lambda$, let us define:

$$Z_i \coloneqq \mathbb{E}[T_{\mathcal{H}_{i-1},\Gamma}(\bar{s})] - \mathcal{L}_{i,\Gamma}^r. \tag{5}$$

Let $\mathcal{F}_i$ be the filtration of history till the end of episode i. Since the policy run in episode i is $\mathcal{H}_{i-1}$ and $\mathcal{H}_{i-1}$ is $\mathcal{F}_{i-1}$-measurable, $\mathbb{E}[\mathcal{L}_{i,\Gamma}^r | \mathcal{F}_{i-1}] = \mathbb{E}[T_{\mathcal{H}_{i-1},\Gamma}(\bar{s})]$. Then Z_i with respect to $\mathcal{F}_{i-1}$ is a martingale difference sequence.

Let $\mathcal{Q}_r$ be a random hypothesis drawn from the uniform distribution over the Λ hypotheses formed at the start of each of the Λ episodes. Then the expected number of steps taken by $\mathcal{Q}_r$ before reaching either the target state $\bar{s}$ or Γ time steps can be written as

$$\begin{aligned} \mathbb{E}\left[T_{\mathcal{Q}_r,\Gamma}(\bar{s})\right] &= \frac{1}{\Lambda}\sum_{i=1}^{\Lambda} \mathbb{E}\left[T_{\mathcal{H}_{i-1},\Gamma}(\bar{s})\right] \\ &\le \frac{1}{\Lambda}\sum_{i=1}^{\Lambda} \mathcal{L}_i^r + \frac{1}{\Lambda}\sum_{i=1}^{\Lambda} Z_i. \end{aligned} \tag{6}$$

Here we use Eq. (4) and the definition of Z_i given in Eq. (5). In the appendix given with the extended version of this article, we prove the upper bounds given in Eq. (7) and Eq. (8) using the regret bound of subroutine $\mathfrak{A}$ and the construction of our algorithm META-EXPLORE.

$$\frac{1}{\Lambda}\sum_{i=1}^{\Lambda} \mathcal{L}_i^r \le \frac{1}{\Lambda} T_r \nu^* \; + \; \frac{1}{\Lambda} B(|\mathcal{K}|, A) \cdot T_r^\alpha \cdot D^\beta \tag{7}$$

with probability at least $1 - \delta_r$.

$$\frac{1}{\Lambda}\sum_{i=1}^{\Lambda} Z_i \le \epsilon L. \tag{8}$$

with probability at least $1 - \delta_r$.

From Eq. (6), (7) and (8), and using $T_r \coloneqq (1 + 3\epsilon)L\Lambda$,

$$\begin{aligned} \mathbb{E}\left[T_{\mathcal{Q}_r,\Gamma}(\bar{s})\right] &\le \frac{1}{\Lambda} T_r \nu^* + \frac{1}{\Lambda} B(|\mathcal{K}|, A) \cdot T_r^\alpha D^\beta + \epsilon L \\ &\le \frac{T_r}{\Lambda} + \frac{1}{\Lambda} B(|\mathcal{K}|, A) \cdot T_r^\alpha D^\beta + \epsilon L \\ &\le (1 + 5\epsilon) L. \end{aligned} \tag{9}$$

The second inequality follows from the application of Lemma 4 while the last inequality is true using Proposition 1. Gathering the error probabilities, Eq. (9) is true with probability at least $1 - (2\delta_r + 2\delta_r) = 1 - 4\delta_r$.

4.3 Failure Probability to Reach the Target State in Γ Steps

Let $p_{\pi,\Gamma}$ be the true probability of failure to reach the target state $\bar{s}$ in Γ steps while following policy π from s_0.

In the appendix given with the extended version of this article, we prove the upper bound given in Eq. (10). In this proof, we show that, for $i = 1, 2 \ldots, \Lambda$, $Y_i \coloneqq p_{\mathcal{H}_{i-1},\Gamma} - \mathbb{1}_{\{T_i^r > \Gamma\}}$ is a martingale difference sequence and then use Azuma-Hoeffding inequlaity [Azuma, 1967; Hoeffding, 1963].

$$p_{\mathcal{Q}_r,\Gamma} \le \hat{p}_r + \epsilon \tag{10}$$

with probability at least $1 - \delta_r$.

4.4 Bound on the Optimum Navigation Time and the Navigation Time of the Output Policy

The output policy for $\bar{s} \in \mathcal{K}$ is denoted as $\pi_{\bar{s}}$.

Lemma 1. *Let r be a successful round and $\bar{s}$ be the concerned target state, then with probability at least* $1 - 2\delta_r$,

$$\tau^*(\bar{s}) \le (1 + \epsilon)L.$$

Lemma 2. *Let r be a successful round and $\bar{s}$ be the concerned target state, then with probability at least* $1 - 5\delta_r$,

$$\tau_{\pi_{\bar{s}}}(\bar{s}) \le (1 + \epsilon)L.$$

Please see the extended version submitted with the supplementary material for this article for the proof of Lemma 1 and 2. We prove these lemmas by first forming an equation for $\tau_{\pi_{\bar{s}}}(\bar{s})$ in terms of $\mathbb{E}\left[T_{\pi_{\bar{s}},\Gamma}(\bar{s})\right]$. Then we use the upper bound on $\mathbb{E}\left[T_{\pi_{\bar{s}},\Gamma}(\bar{s})\right]$ from Eq. (9), Eq. (10) and the construction of META-EXPLORE, particularly the check given in Eq. (2).

4.5 Bound on the Probability of Erroneously Rejecting a Target State $\bar{s} \in \mathcal{S}_L^{\rightarrow}$

Since $\mathcal{S}_L^{\rightarrow}$ is unknown, the algorithm needs to make sure that, with high probability, none of the states in $\mathcal{S}_L^{\rightarrow}$ are "missed". This is proven by Lemma 12 in [Lim and Auer, 2012] which states that with high probability, unless all of $\mathcal{S}_L^{\rightarrow}$ is known, at least one of the states in $\mathcal{S}_L^{\rightarrow} \setminus \mathcal{K}$ is also in $\mathcal{U}$ and that state is reachable in L steps with a policy on $\mathcal{K}$ at that time. Then, below we prove that if a state $\bar{s} \in \mathcal{S}_L^{\rightarrow}$, such that it is reachable in L steps with a policy on K at that time, is selected as the target state, then with high probability, it becomes known.

Lemma 3. *Consider a round r with target state $\bar{s} \in \mathcal{S}_L^{\rightarrow}$. Let K be the set of known states at the beginning of round r. If $\exists \pi$ on K such that $\tau_\pi(\bar{s}) \leq L$, then the probability of failure is at most $2\delta_r$.*

Here, we provide a proof-sketch of Lemma 3. Please see the extended version of this article for the complete proof.

Proof-skecth. First we use Lemma 11 from [Tarbouriech *et al.*, 2019] to show that if $\bar{s} \in \mathcal{S}_L^{\rightarrow}$ then $T_r \leq (1+3\epsilon)L\Lambda$ with probability at least $1-\delta_r$. Then using a similar process as in the proof of Eq. (10), we show that the performance check in Eq. (2) is satisfied with probability at least $1-\delta_r$. □

Let $s_r \in \mathcal{S}_L^{\rightarrow}$ be a state which was erroneously rejected by the algorithm at the end of round r. Let $\mathcal{K}'$ be the set of known states at the beginning of round r. Then either one of the following is true :

1. $\exists \pi$ on $\mathcal{K}'$ such that $\tau_\pi(s_r) \leq L$.
2. $\nexists \pi$ on $\mathcal{K}'$ such that $\tau_\pi(s_r) \leq L$.

The first case is handled by Lemma 3.

Assume that Lemma 3 holds for each round. For the second case, by the definition of incrementally reachable states $\mathcal{S}_L^{\rightarrow}$ (see Definition 4), there exists a sequence of states $s_0, s_1, s_2, \ldots, s_n$ and a policy π on those states such that $\tau_\pi(s_r) \leq L$. Moreover, each state s_i in the sequence is also incrementally reachable with respect to the states in the sequence preceding s_i. So for each state s_i in the sequence, there exists a policy on $s_0, \ldots, s_{i-1}$ that can reach s_i in L steps (in expectation). Thus for each state s_i, there is an $s \in s_0, \ldots, s_{i-1}$ and an action a with $P(s_i|s,a) \geq \frac{1}{L}$. Using Lemma 17 from [Lim and Auer, 2012], each of these states would be discovered using state discovery with high probability. In the sequence $s_0, s_1, s_2, \ldots, s_n$, there exists a state s' which is immediately reachable from s_0. When s' is considered as the target state, it would be added to $\mathcal{K}$. Similarly, each state s_i in the sequence would be added to $\mathcal{K}$ if $s_0, s_1, \ldots, s_{i-1}$ is in $\mathcal{K}$ at the beginning of the round. If not, we repeat the same argument. Then, eventually when s_r is considered as a target state in a round with $s_0, s_1, \ldots, s_n$ already in $\mathcal{K}$, it would be added to $\mathcal{K}$ at the end of the round.

4.6 Supplementary Lemmas and Propositions

In the following, we provide proof-sketches. Please consult the extended version of this article for the complete proofs.

Lemma 4. *Let $\nu(\pi)$ be as defined in Eq. (3). Then,*

$$\nu(\pi) = \frac{\tau_\pi(\bar{s})}{\tau_\pi(\bar{s})+1}.$$

The Lemma follows from the definition of $\nu(\pi)$ and Lemma 9 in [Tarbouriech *et al.*, 2019].

Lemma 5. *Let $\mathcal{M}_{\bar{s}}$ be an induced MDP with $\tau^*(s) \leq x$ for every state s. Then the diameter of $\mathcal{M}_{\bar{s}}$ is at most $x+1$.*

For arbitrary s_1, s_2, we construct a non-stationary policy with the expected navigation time from s_1 to s_2 of at-most $x+1$. Then we utilize the fact that for a given (fixed) MDP the optimal average reward is attained by a stationary policy and cannot be increased by using non-stationary policies.

Proposition 1. *For any successful round, with probability at least $1-2\delta_r$,*

$$\frac{1}{\Lambda} B(|\mathcal{K}|, A) \cdot T_r^\alpha \cdot D^\beta \leq \epsilon L.$$

Proposition 1 follows from Lemma 1 and Lemma 5.

4.7 Proof of Theorem 1

Proof. Here, we prove the three claims stated in Theorem 1.

(a) From Section 4.5, the probability of erroneously rejecting a state from $\mathcal{S}_L^{\rightarrow}$ is bounded by the sum of the probability that Lemma 3 does not hold in some round r, which is at most $\sum_r 2\delta_r$ and the total error probability of Lemma 12 and Lemma 17 from [Lim and Auer, 2012], which is $\delta/4$. From Lemma 1, all the states $s \in \mathcal{K}$ have $\tau^*(s) \leq (1+\epsilon)L$ with probability at least $1-\sum_r 2\delta_r$. Hence, $S := |\mathcal{K}| \leq |\mathcal{S}_{(1+\epsilon)L}^{\rightarrow}|$.

(b) **Exploration steps during state discovery**: As explained in Section 3.1, sampling each action in a new known state s requires at most $(1+\epsilon)L+1$ steps on average. Since each action $a \in \mathcal{A}$ is sampled $\left\lceil L \log \frac{8AL|\mathcal{K}|^2}{\delta} \right\rceil$ times for a state $s \in \mathcal{K}$, the number of exploration steps during state discovery due to all S states in $\mathcal{K}$ is $O\left(SAL^2 \log \frac{8ALS^2}{\delta}\right)$.

Exploration steps during target state evaluation: Since each action $a \in \mathcal{A}$ is sampled in each $s \in \mathcal{K}$ for $\left\lceil L \log \frac{8AL|\mathcal{K}|^2}{\delta} \right\rceil$ times, at most $\left\lceil AL \log \frac{8ALS^2}{\delta} \right\rceil$ states are added to $\mathcal{U}$ due to a single known state. As $|\mathcal{K}| = S$, and the neighbors (which are not in $\mathcal{K}$ currently) of all the previous known states are added to the set of candidate states every time a new state becomes known, the total number of rounds is at-most $O\left(\left\lceil S^2 AL \log \frac{8ALS^2}{\delta} \right\rceil\right)$. Thus, the total number of exploration steps during target state evaluation is at most

$$\begin{aligned} &= \tilde{O}\Bigg(\left\lceil S^2 AL \log \frac{8ALS^2}{\delta} \right\rceil \cdot (1+3\epsilon)L \\ &\quad \cdot \frac{2^{\max(4,\beta/(1-\alpha))}}{\epsilon^{\max(4,1/(1-\alpha))}} \\ &\quad \cdot \left[B(S,A) \cdot (1+3\epsilon)^{\alpha+\beta} \cdot L^{\alpha+\beta-1}\right]^{\frac{1}{1-\alpha}} \Bigg) \\ &= \tilde{O}\Big(\frac{1}{\epsilon^{\max(4,1/(1-\alpha))}} \cdot S^2 A [B(S,A)]^{\frac{1}{1-\alpha}} \\ &\quad \cdot L^{2+\frac{\alpha+\beta-1}{1-\alpha}} \Big). \end{aligned}$$

(c) As per Lemma 2, for all $s \in \mathcal{K}$, output policy π_s satisfies $\tau_{\pi_s}(s) \leq (1+\epsilon)L$ with probability $1 - \sum_r 5\delta_r$.

Collecting the error probabilities, the total probability of failure is at most $\frac{\delta}{4} + \sum_r 9\delta_r \leq \delta$, where we use the fact that $\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$ and at most $|\mathcal{K}| \cdot \left\lceil AL \log \frac{8AL|\mathcal{K}|^2}{\delta} \right\rceil$ states are added to $\mathcal{U}$ due to a single known state. □

5 Experimental Results

In this section, we provide experimental results on simulated examples to validate the effectiveness of our proposed algorithm and our theoretical results. Note that since this is the first work providing a general conversion from a regret-minimizing RL algorithm to an exploration algorithm, there is no benchmark available.

Below, we describe the problem setting used in our experiments. The MDP is of unbounded state space and has 2 actions – 0 and 1. The states are structured similar to a tertiary tree with the starting state being the root. On taking action 0 in a state node, transition to the left child occurs with very high probability $P_{0,l}$, transition to the middle child occurs with very low probability $P_{0,m}$ and transition to the right child occurs with very low probability $P_{0,r}$. On taking action 1 in a state node, transition to the left child occurs with very low probability $P_{1,l}$, transition to the middle child occurs with very low probability $P_{1,m}$ and transition to the right child occurs with very high probability $P_{1,r}$.

We use either UCRL2 [Jaksch *et al.*, 2010] or UCRL2b [Fruit *et al.*, 2020] as a black-box subroutine for Meta-Explore. The problem settings used in the experiments are described in Table 1. All the results shown in this article are averaged over 100 independent runs. Figure 1 shows the empirical sample complexity for Meta-Explore for problem setting 1 using $\delta = 0.2$ and $\epsilon = 2/3$. Figure 2 shows the empirical sample complexity for Meta-Explore for problem setting 2 using $\delta = 0.1$ and $\epsilon = 1/3$. For each value of the number of reachable states shown in Figure 1 and 2, both the instantiations of Meta-Explore were able to find appropriate output policies for corresponding reachable states in at least $1 - \delta$ fraction of the runs. In the appendix given with the extended version of this article, we provide additional experimental results.

The experimental results show that, using an appropriate RL algorithm as a black-box, our algorithm explores the state space properly. For both the theoretical upper bound and the empirical sample complexity, Meta-Explore using UCRL2b shows better dependence on L than Meta-Explore using UCRL2 which corroborates the dependence proven in Theorem 1. We also note, in general, for higher values of the number of reachable states (i.e., higher values of L), the empirical sample complexity tends to be much better than the sample complexity promised by the theoretical upper bounds. This further provides a testimony for our approach which is better able to leverage knowledge gained from solving simpler problems (i.e., smaller values of L) in order to solve more difficult problems efficiently. It also points to a way of improving the empirical performance of our approach in scenarios where solutions for simpler problems are known and can be provided to the algorithm which will in turn use them to find solutions for difficult problems in an efficient manner.

Setting	$P_{0,l}$	$P_{0,m}$	$P_{0,r}$	$P_{1,l}$	$P_{1,m}$	$P_{1,r}$
Setting 1	0.90	0.050	0.050	0.050	0.050	0.90
Setting 2	0.95	0.025	0.025	0.025	0.025	0.95

Table 1: Description of problem settings

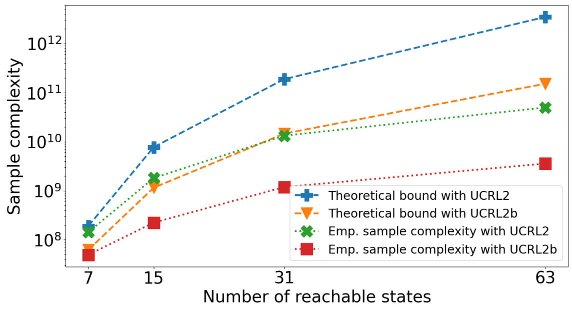

Figure 1: Results for setting 1 with $\delta = 0.2$, and $\epsilon = 2/3$

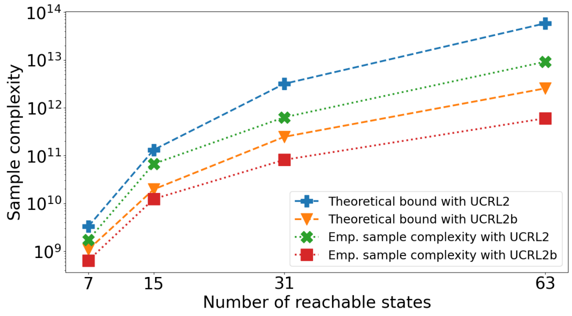

Figure 2: Results for setting 2 with $\delta = 0.1$, and $\epsilon = 1/3$

6 Concluding Remarks

We considered the problem of autonomous exploration in an unknown stationary environment. Our proposed algorithm can use any online RL algorithm (with appropriate regret guarantees) as a black-box to solve the relevant sub-tasks. We proved an upper bound on its sample complexity in terms of the regret bound of the used black-box RL algorithm. Our experimental results demonstrate the applicability of our proposed algorithm for the considered problem and the correctness of our theoretical results.

Interesting directions for future work include: 1) extending the problem definition to include non-stationary environments, 2) extending the solution approach to solve the multi-goal stochastic shortest path problem introduced by [Cai *et al.*, 2022].

Acknowledgments

This work is supported by the Dutch Research Council (NWO) in the framework of the TEPAIV research project (project number 612.001.752) and the Austrian Science Fund (FWF): TAI 590-N.

References

[Achiam and Sastry, 2017] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *CoRR*, abs/1703.01732, 2017.

[Agarwal *et al.*, 2020] Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[Azar *et al.*, 2019] Mohammad Gheshlaghi Azar, Bilal Piot, Bernardo A. Pires, Jean-Bastien Grill, Florent Altché, and Rémi Munos. World discovery models. *CoRR*, abs/1902.07685, 2019.

[Azuma, 1967] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Math. J. (2)*, 19(3):357–367, 1967.

[Baranes and Oudeyer, 2009] A. Baranes and P.-Y. Oudeyer. R-IAC: Robust intrinsically motivated exploration and active learning. *IEEE Transactions on Autonomous Mental Development*, 1:155–169, 2009.

[Bartlett and Tewari, 2009] Peter L. Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 35–42, 2009.

[Burda *et al.*, 2019] Yuri Burda, Harrison Edwards, Deepak Pathak, Amos J. Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *ICLR*, 2019.

[Cai *et al.*, 2022] Haoyuan Cai, Tengyu Ma, and Simon Du. Near-optimal algorithms for autonomous exploration and multi-goal stochastic shortest path. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2434–2456. PMLR, 17–23 Jul 2022.

[Cesa-Bianchi *et al.*, 1997] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, page 427–485, May 1997.

[Cesa-Bianchi *et al.*, 2006] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theor.*, 50(9):2050–2057, September 2006.

[Ecoffet *et al.*, 2019] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *CoRR*, abs/1901.10995, 2019.

[Ecoffet *et al.*, 2020] Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return then explore. *CoRR*, abs/2004.12919, 2020.

[Florensa *et al.*, 2018] Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1515–1528. PMLR, 10–15 Jul 2018.

[Freund and Schapire, 1999] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, page 277–296, 1999.

[Fruit *et al.*, 2020] Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Improved analysis of UCRL2 with empirical bernstein inequality. *CoRR*, abs/2007.05456, 2020.

[Gajane *et al.*, 2019] Pratik Gajane, Ronald Ortner, Peter Auer, and Csaba Szepesvári. Autonomous exploration for navigating in non-stationary cmps. *CoRR*, abs/1910.08446, 2019.

[Gottlieb *et al.*, 2013] Jacqueline Gottlieb, Pierre-Yves Oudeyer, Manuel Lopes, and Adrien Baranes. Information-seeking, curiosity, and attention: computational and neural mechanisms. *Trends in cognitive sciences*, 17(11):585–593, 2013.

[Haber *et al.*, 2018] Nick Haber, Damian Mrowca, Stephanie Wang, Li F Fei-Fei, and Daniel L Yamins. Learning to play with intrinsically-motivated, self-aware agents. In *Advances in Neural Information Processing Systems*, pages 8388–8399, 2018.

[Hazan *et al.*, 2019] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, volume 97, pages 2681–2691, 2019.

[Helmbold and Warmuth, 1995] David P. Helmbold and Manfred K. Warmuth. On weak learning. *J. Comput. Syst. Sci.*, page 551–573, June 1995.

[Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[Houthooft *et al.*, 2016] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Variational information maximizing exploration. In *NIPS 2016 Deep Learning Symposium*, 2016.

[Jaksch *et al.*, 2010] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

[Jin *et al.*, 2020] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of

Proceedings of Machine Learning Research, pages 4870–4879. PMLR, 2020.

[Kaplan and Oudeyer, 2003] Frédéric Kaplan and Pierre-Yves Oudeyer. Motivational principles for visual know-how development. *Lund University Cognitive Studies*, 101:73–80, 2003.

[Lim and Auer, 2012] Shiau Hong Lim and Peter Auer. Autonomous exploration for navigating in mdps. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 40.1–40.24, 2012.

[Littlestone, 1989] Nick Littlestone. From on-line to batch learning. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, COLT '89, page 269–284, 1989.

[Lopes *et al.*, 2012] Manuel Lopes, Tobias Lang, Marc Toussaint, and Pierre-Yves Oudeyer. Exploration in model-based reinforcement learning by empirically estimating learning progress. In *Advances in neural information processing systems*, pages 206–214, 2012.

[Ostrovski *et al.*, 2017] Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2721–2730, 2017.

[Oudeyer and Kaplan, 2007] Pierre-Yves Oudeyer and Frédéric Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1, 2007.

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

[Schmidhuber, 2010] Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *Autonomous Mental Development, IEEE Transactions on*, 2:230–247, 2010.

[Shalev-Shwartz, 2012] Shai Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Mach. Learn.*, 4(2):107–194, February 2012.

[Singh *et al.*, 2004] Satinder P. Singh, Andrew G. Barto, and Nuttapong Chentanez. Intrinsically motivated reinforcement learning. In *NIPS*, 2004.

[Singh *et al.*, 2010] Satinder P. Singh, Richard L. Lewis, Andrew G. Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE T. Autonomous Mental Development*, 2:70–82, 2010.

[Tarbouriech *et al.*, 2019] Jean Tarbouriech, Evrard Garcelon, Michal Valko, Matteo Pirotta, and Alessandro Lazaric. No-regret exploration in goal-oriented reinforcement learning. *CoRR*, abs/1912.03517, 2019.

[Tarbouriech *et al.*, 2020] Jean Tarbouriech, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Improved sample complexity for incremental autonomous exploration in mdps. In *Advances in Neural Information Processing Systems 33 pre-proceedings (NeurIPS 2020)*, 2020.

[Weng *et al.*, 2001] Juyang Weng, James McClelland, Alex Pentland, Olaf Sporns, Ida Stockman, Mriganka Sur, and Esther Thelen. Autonomous mental development by robots and animals. *Science*, 291(5504):599–600, 2001.