The Online Discovery Problem and Its Application to Lifelong Reinforcement Learning

Emma Brunskill
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
ebrunskill@cs.cmu.edu

Lihong Li
Microsoft Research
One Microsoft Way
Redmond, WA 98052
lihongli@microsoft.com

Abstract

Transferring knowledge across a sequence of related tasks is an important challenge in reinforcement learning. Despite much encouraging empirical evidence that shows benefits of transfer, there has been very little theoretical analysis. In this paper, we study a class of lifelong reinforcement-learning problems: the agent solves a sequence of tasks modeled as finite Markov decision processes (MDPs), each of which is from a finite set of MDPs with the same state/action spaces and different transition/reward functions. Inspired by the need for cross-task exploration in lifelong learning, we formulate a novel online discovery problem and give an optimal learning algorithm to solve it. Such results allow us to develop a new lifelong reinforcement-learning algorithm, whose overall sample complexity in a sequence of tasks is much smaller than that of single-task learning, with high probability, even if the sequence of tasks is generated by an adversary. Benefits of the algorithm are demonstrated in a simulated problem.

1 Introduction

Transfer learning, the ability to take prior knowledge and use it to perform well on a new task, is an essential capability of intelligence. Tasks themselves often involve multiple steps of decision making under uncertainty. Therefore, lifelong learning across multiple reinforcement-learning (RL) [24] tasks is of significant interest. Potential applications are enormous, from leveraging information across customers, to speeding robotic manipulation in new environments. In the last decades, there has been much previous work on this problem, which predominantly focuses on providing promising empirical results but with little formal performance guarantees (*e.g.*, [21, 26, 25, 22] and the many references therein), or in the offline/batch setting [15], or for multi-armed bandits [1].

In this paper, we focus on a special case of lifelong reinforcement learning which captures a class of interesting and challenging applications. We assume that all tasks, modeled as finite Markov decision processes or MDPs, have the same state and action spaces, but may differ in their transition probabilities and reward functions. Furthermore, the tasks are elements of a finite collection of MDPs that are initially unknown to the agent. Such a setting is particularly motivated by applications to user personalization, such as domains like education, healthcare and online marketing, where one can consider each "task" as interacting with one particular individual, and the goal is to leverage prior experience to improve performance with later users. Indeed assuming all users can be treated as roughly falling into a finite set of groups has already been explored in multiple such domains [7, 17, 19], as it offers a form of partial personalization, allowing the system to more quickly learn good interactions with the user (than learning for each user separately) but still offering much more personalization than modeling all individuals as the same.

A critical issue in transfer or lifelong learning is how and when to leverage information from previous tasks in solving the current one. If the new task represents a different MDP with a different optimal policy, then leveraging prior task information may actually result in substantially worse performance than learning with no prior information, a phenomenon known as *negative transfer* [25]. Intuitively, this is partly because leveraging prior experience (in the form of samples, value functions, policies or others) can prevent an agent from visiting parts of the state space which differ in the new task, and yet would be visited under the optimal policy for the new task. In other words, there is a unique need for *multi-level exploration* in lifelong reinforcement learning: in addition to exploration typically needed to obtain optimal policies in single-task RL (*i.e.*, within-task learning), the agent also needs sufficient exploration to uncover *relations among tasks* (*i.e.*, cross-task transfer).

To this end, the agent faces an *online discovery* problem: the new task may be the same[1] as one of the prior tasks, or may be a novel one. The agent can choose to treat each new task as an novel task or as an instance of a prior task. Failing to correctly treat a novel task as new, or treating an existing task as the same as a prior task, will lead to sub-optimal performance. In Section 2, we formulate a novel online-discovery problem that captures such a challenge, and present an algorithm that achieves optimal performance with matching upper and lower regret bounds. These results are then used in Section 3 to create a new lifelong learning algorithm. Not only does the new algorithm relax multiple critical assumptions needed by prior work, it can also immediately start to share information across tasks and is guaranteed to have substantially lower overall sample complexity than single-task learning over a sequence of tasks.

The main contributions are as follows. First, we propose a novel lifelong reinforcement-learning algorithm, designed to have efficient, *simultaneous* exploration for within-task learning and cross-task transfer when tasks are drawn from a finite set of discrete state and action MDPs. Second, we analyze the algorithm's sample complexity, a theoretical measure of learning speed in online reinforcement learning. Our results show how knowledge transfer provably decreases sample complexity, compared to single-task reinforcement learning, when solving a sequence of tasks. Third, we provide simulation results that compare our algorithms to single-task learning as well as to state-of-the-art lifelong learning algorithms, that illustrate the benefits and relative advantages of the new algorithm. Finally, as a by-product, we formalize a novel online discovery problem and give optimal algorithms, as a means to facilitate development of our lifelong learning algorithm. This contribution may be of broader interest in other related meta-learning problems with a need for similar exploration to uncover inter-task relation.

Related Work. There have been substantial interests in lifelong learning across sequential decision making tasks for multiple decades (see, *e.g.*, [21, 22], and the many references therein). Lifelong RL is closely related to *transfer RL*, in which information (or data) from source MDPs is used to accelerate learning in the target MDP: [25] provides an excellent survey of work in this area. A distinctive element in lifelong RL is that every task is both a target and a source task. Consequently, the agent has to explore the current task once in a while to allow better knowledge to be transferred to solve tasks in the future; this is the motivation for the online discovery problem we study here.

The setting we consider, of sampling MDP models from a finite set, is closely related to multiple previously considered setups. [13] describe hidden parameter MDPs, which cover our setting in this work as well as others where there is a latent variable that captures some key aspects of each task encountered. [26] tackle a similar problem using a hierarchical Bayesian model for the distribution from which tasks are generated. To our best knowledge, the vast majority of prior work on lifelong learning and transfer learning has focused on algorithmic and empirical innovations, and there has been very little formal analysis before our presented work for *online* learning. An exception is a two-phase algorithm [3] with provably small sample complexity, but makes a few critical assumptions.

The online discovery problem appears new, although it has connections to several other existing problems. One is bandit problems [4] that also require an effective exploration/exploitation trade-off. However, in bandits every action leads to an observed loss, while in online discovery, only one action has observable loss. The apple tasting (AT) problem [9] has a similar flavor, but with a different structure in the loss matrix; furthermore, the analysis is in the mistake bound model that is not suitable here. [5] tackles "optimal discovery" in a very different setting, focusing on quick identification of hidden elements given access to different sampling distributions (called "experts"). Finally, ODP is related to the missing mass problem (MMP) [18]. While MMP is a pure prediction problem, ODP involves decision making, hence requires balancing exploration and exploitation.

2 The Online Discovery Problem

Motivated by the need for cross-task exploration in lifelong RL, in this section, we study a novel online discovery problem that will play a crucial role in developing new lifelong RL algorithms in Section 3. In addition to the appli-

[1]Even if no identical MDP is experienced, MDPs with similar model parameters have similar value functions. Thus, fintely many policies suffice to represent ϵ-optimal policies for all MDPs with shared state/actions.

Table 1: Loss matrix in online discovery: rows correspond to the action of exploration ($A = 1$) or exploitation ($A = 0$); columns indicate whether the current item is novel or not. Ideally, exploration happens when and only when the item is novel (*i.e.*, unidentified in the past). The ρs specify costs of actions in different situations.

	0	1
$A = 0$	ρ_0	ρ_3
$A = 1$	ρ_1	ρ_2

cation here, this problem may be of independent interest in other meta-learning problems where there is a need for efficient exploration to uncover cross-task relation.

2.1 Formulation

We now describe the *online discovery problem (ODP)*, a sequential game where the agent decides in each round whether to explore the item in that round. Let $\mathcal{M}$ be an unknown set of C items to be discovered by a learner, and $\mathcal{A} = \{0\ (\text{“exploitation”}), 1\ (\text{“exploration”})\}$ a set of two actions. The learner need not know C. Initially, the set of discovered items $\mathcal{M}_1$ is $\emptyset$. The learner is also given four constants, $\rho_0 < \rho_1 \le \rho_2 \le \rho_3$, specifying the loss matrix L in Table 1.
The game proceeds as follows. For round $t = 1, 2, \ldots, T$:

- Environment selects an item $M_t \in \mathcal{M}$.
- Without knowing identity of M_t, the learner chooses action $A_t \in \mathcal{A}$, and suffers a loss $L_t = L(A_t, \mathbb{I}\{M_t \in \mathcal{M}_t\})$, where L is the loss matrix in Table 1. The learner observes L_t when $A_t = 1$, and $\perp$ (“no observation”) otherwise.
- If $A_t = 1$, then $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t \cup \{M_t\}$; otherwise, $\mathcal{M}_{t+1} \leftarrow \mathcal{M}_t$.

At the beginning of round t, we define $H_t := (A_1, L_1, M_2, A_2, L_2, \ldots, A_{t-1}, L_{t-1}, M_{t-1})$, the *history* up to t. An algorithm is *admissible*, if it chooses actions A_t based on H_t and possibly an external source of randomness. As in the online-learning literature, we distinguish two settings. In the *stochastic* setting, the environment picks M_t in an i.i.d. (independent and identically distributed) manner from an unknown distribution μ over $\mathcal{M}$. In the *adversarial* setting, the sequence $(M_t)_t$ can be generated by an adversarial in an arbitrary way that depends on H_t.

If the learner *knew* the identity of M_t, the optimal strategy is to choose $A_t = 1$ if $M_t \notin \mathcal{M}_t$ and $A_t = 0$ otherwise. After T rounds, this ideal strategy has the optimal loss of $L^*(T) := \rho_2 C^* + \rho_0(T - C^*)$, where $C^* \le C$ is the number of distinct items in the sequence $(M_t)_t$. The challenge, of course, is that the learner does not know M_t before selecting action A_t. She thus has to balance exploration (taking $A_t = 1$ to see if M_t is novel) and exploitation (taking $A_t = 0$ to yield small loss ρ_0 if it is likely that $M_t \in \mathcal{M}_t$). Clearly, over- and under-exploration can lead to suboptimal strategies. Therefore, we are interested in finding algorithms $\mathbf{A}$ to have smallest cumulative loss as possible. Formally, an algorithm $\mathbf{A}$ for online discovery is a (possibly stochastic) policy that maps histories to actions: $A_t = \mathbf{A}(H_t)$. The total T-round loss suffered by $\mathbf{A}$ is $L(\mathbf{A}, T) := \sum_{t=1}^{T} L_t$. The *$T$-round expected regret* of an algorithm $\mathbf{A}$ is defined by $\bar{R}(\mathbf{A}, T) = \mathbb{E}[L(\mathbf{A}, T) - L^*(T)]$, where the expectation is taken with respect to any randomness in the environment as well as in $\mathbf{A}$.

It should be noted that we could set $\rho_0 = 0$ without affecting the definition of regret. However, we allow it to be positive for convenience when mapping lifelong RL into online discovery later.

2.2 The Explore-First Algorithm

In the *stochastic* case, it can be shown that if an algorithm takes a total of E explorations, its expected regret is smallest if these exploration rounds occur at the very beginning. The resulting strategy is sometimes called EXPLORE-FIRST, or EXPFIRST for short, in the multi-armed bandit literature.

With knowledge of T, C and $\mu_m := \min_{M \in \mathcal{M}} \mu(M)$, one may set E to $E^* = E(C, \mu_m) := \mu_m^{-1} \ln \frac{C}{\delta}$, so that all items in $\mathcal{M}$ will be discovered in the first E^* rounds, with probability $1 - \delta$. After that, it is safe to always exploit ($A_t \equiv 0$). The total expected loss can be upper bounded as: $\mathbb{E}[L(\textsc{ExpFirst}, T)] \le \rho_1 E^* + \rho_2 C + \rho_0 (T - E^*) + \delta \rho_3 T$, where the first two terms correspond to the loss incurred in the exploration rounds, the third the loss incurred in exploitation rounds, and the last the loss incurred in the lower-probability event (that some item in $\mathcal{M}$ does not occur in the first E^* rounds). This upper bound can be minimized by optimizing δ, provided that C and ρ_i's are known. The result is summarized in the following proposition, proved in Appendix B.1:

Proposition 1. *With $E^* = \mu_m^{-1} \ln \frac{C}{\delta}$ with $\delta = \frac{\rho_1}{\rho_3 \mu_m T}$,* ExpFirst *has the following regret bound:*

$$\bar{R}(\text{ExpFirst}, T) \leq \frac{\rho_1}{\mu_m} \left(\ln T + \ln \frac{C \mu_m \rho_3}{\rho_1} + 1 \right).$$

2.3 Forced Exploration

While ExpFirst is effective in stochastic ODPs, in many applications, the task generation distribution may be non-stationary (*e.g.*, different types of users may use the Internet at different time-of-the-day) or even adversarial (*e.g.*, an attacker may present certain MDPs in earlier rounds in lifelong RL to cause an algorithm to perform poorly in future MDPs). We now study a simple yet more general algorithm, ForcedExp (for *Forced Exploration*), and proves an upper bound for its regret. In the next subsection, we will present a matching lower bound, indicating optimality of this algorithm.

Before the game starts, the algorithm determines a *fixed* schedule for exploration. Specifically, it pre-decides a sequence of "exploration rates": $\eta_1, \eta_2, \dots, \eta_T \in [0,1]$. Then, in round t, it chooses the exploration action with probability η_t: $\mathbb{P}\{A_t = 1\} = \eta_t$ and $\mathbb{P}\{A_t = 0\} = 1 - \eta_t$.

The main result about ForcedExp is the following theorem, proved in Section B.3

Theorem 2. *If we run* ForcedExp *with non-increasing exploration rates $\eta_1 \geq \cdots \geq \eta_T > 0$, then*

$$\mathbb{E}[L(\text{ForcedExp}, T)] \leq \rho_0 T + \frac{C\rho_3}{\eta_T} + \rho_1 \sum_{t=1}^{T} \eta_t.$$

The theorem directly implies the following corollary (proved in Section B.4):

Corollary 3. *If we set $\eta_t = t^{-\alpha}$ (polynomial decaying rate) for some parameter $\alpha \in (0,1)$, then*

$$\bar{R}(\text{ForcedExp}, T) \leq C\rho_3 T^{\alpha} + \frac{\rho_1}{1-\alpha} T^{1-\alpha}.$$

If we set $\eta_t \equiv \eta$ for some $\eta \in (0,1)$ (fixed rate), then

$$\bar{R}(\text{ForcedExp}, T) \leq \frac{C\rho_3}{\eta} + \eta\rho_1 T.$$

Furthermore, the bounds above are both on the order of $O(\sqrt{T})$ by setting $\alpha = 1/2$ and $\eta = \sqrt{C\rho_3/(\rho_1 T)}$, respectively.

The results show that ForcedExp eventually performs as well as the optimal policy that knows the identity of M_t in every round t, no matter how M_t is generated. Moreover, the per-round regret decays on the order of $1/\sqrt{T}$, which we will show to be optimal. Although in the *worst* case ForcedExp with fixed rate achieves a regret of the same order as the one with polynomial rates, it is expected that the latter is better in the stochastic setting, at least empirically.

Note that knowledge of relevant quantities such as ρs and T is useful to optimize parameters in Corollary 3. In particular, the value of η as given in the corollary depends on T. When T is unknown, one can still apply the standard doubling trick to get the same $O(\sqrt{T})$ regret bound (Section B.4).

2.4 Lower Bound

The main result in this section, Theorem 4, shows the $O(\sqrt{T})$ regret bound for ForcedExp is essentially not improvable, in term of T-dependence, even in the stochastic case. The idea of the proof, given in Section B.5, is to construct a hard instance of stochastic ODP. On one hand, $\Omega(\sqrt{T})$ regret is suffered unless all C items in $\mathcal{M}$ are discovered. On the other hand, most of the items have low probability μ_m of being sampled, requiring the learner to take the exploration action $A = 1$ many times to discover all C items. The lower bound follows from an appropriate value of μ_m.

Theorem 4. *There exists an online discovery problem, where every admissible algorithm suffers an expected regret of* $\Omega(\sqrt{T})$.

Although the lower bound matches the upper bounds in terms of T, we have not attempted to match dependence on other quantities like C, which are often less important than T.

This lower bound may seem to contradict EXPFIRST's logarithmic upper bound in Proposition 1. However, that upper bound is a problem-specific bound and requires knowledge of C and μ_m. Without knowing μ_m, the algorithm has to choose $\mu_m = \Theta(\frac{1}{\sqrt{T}})$ in the exploration phrase; otherwise, there is a chance it may not be able to discover an item M with $\mu(M) = \Omega(\frac{1}{\sqrt{T}})$, suffering $\Omega(\sqrt{T})$ expected regret. With this value of μ_m, the bound in Proposition 1 has an $\tilde{O}(\sqrt{T})$ dependence.

3 PAC-MDP Lifelong Reinforcement Learning

Building on the ODP results established in Section 2, we now turn to lifelong reinforcement learning.

3.1 Preliminaries

We consider reinforcement learning [24] in discrete-time, finite MDPs specified by a five-tuple: $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, P the transition probability function, $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$ the reward function, and $\gamma \in (0, 1)$ the discount factor. Denote by S and A the numbers of states and actions, respectively. A policy $\pi : \mathcal{S} \to \mathcal{A}$ specifies what action to take in a given state. Its state and state–action value functions are denoted by $V^\pi(s)$ and $Q^\pi(s, a)$, respectively. The optimal value functions for an optimal policy π^* are V^* and Q^* so that $V^*(s) = \max_\pi V^\pi(s)$ and $Q^*(s, a) = \max_\pi Q^\pi(s, a)$, for all s and a. Finally, let $V_{\max}$ be a known upper bound of $V^*(s)$, which is at most $1/(1 - \gamma)$.

In RL, P and R are initially unknown to the agent, who must learn to optimize its policy via interaction with the MDP. Various frameworks have been proposed to capture the learning speed or effectiveness of a single-task online reinforcement-learning algorithm, such as regret analysis (*e.g.*, [10]). Here, we focus on another useful notion known as *sample complexity of exploration* [11], or *sample complexity* for short. However, some of our ideas, especially those related to cross-task transfer and the online discovery problem, may also be useful for regret analysis.

Fix parameters $\epsilon, \delta > 0$. Any RL algorithm $\mathbf{A}$ can be viewed as a nonstationary policy, whose value functions, $V^{\mathbf{A}}$ and $Q^{\mathbf{A}}$, are defined similarly to the stationary-policy case. When $\mathbf{A}$ is run on an unknown MDP, we call it a mistake at step t if the algorithm chooses a suboptimal action, namely, $V^*(s_t) - V^{\mathbf{A}_t}(s_t) > \epsilon$. If the number of mistakes is at most $\zeta(\epsilon, \delta)$ with probability at least $1 - \delta$, for any fixed $\epsilon > 0$ and $\delta > 0$, then the sample complexity of $\mathbf{A}$ is ζ. Furthermore, if ζ is polynomial in S, A, $1/(1 - \gamma)$, $1/\epsilon$, and $\ln(1/\delta)$, then $\mathbf{A}$ is called *PAC-MDP* [23]. Note that the definition of sample complexity does not impose any condition on when the ϵ-suboptimal steps occur: in fact, some of these sub-optimal steps may occur indefinitely far into the future.

The earliest, and most representative, PAC-MDP algorithms for finite MDPs are model-based algorithms, E^3 [12] and RMAX [2, 11]. In the heart of both algorithms is the distinction between *known* and *unknown* states. When an RMAX agent acts in an unknown environment, it maintains an estimated model for the unknown MDP, using empirically observed transitions and rewards. When it has taken a certain action in a state sufficiently often, as specified by a threshold of how many times the action has been taken in that state, it has high confidence in the accuracy of its estimated model in that state–action pair (thanks to concentration inequalities like the Azuma-Hoeffding inequality), and that state–action pair is considered known. Other (unknown) state–actions are assigned maximal reward to encourage exploration. With such an optimism-in-the-face-of-uncertainty principle, RMAX can be shown to either explore (reaching an unknown state–action in a short amount of time) or exploit (achieving near-optimal discounted cumulative reward). Since the number of visits to unknown state–action pairs is bounded by a polynomial function in relevant quantities, RMAX is PAC-MDP. The intuition behind E^3 is similar.

3.2 Balancing Cross-task Exploration/Exploitation in Lifelong RL

In lifelong reinforcement learning, the agent seeks to maximize its reward as it acts in a sequence of T MDPs. If prior tasks are related to future tasks, we expect leveraging knowledge of this prior experience may lead to enhanced performance. Formally, following previous work [3, 26], and motivated by numerous applications [21, 26, 25, 22] we assume a finite set $\mathcal{M}$ of possible MDPs. The agent solves a sequence of T tasks, with $M_t \in \mathcal{M}$ denoting the MDP corresponding to the t-th task. Before solving the task, the agent does not know whether or not M_t has been encountered before. It then acts in M_t for H steps, where H is a given horizon, and is allowed to take advantage of any information extracted from solving previous tasks $\{M_1, \ldots, M_{t-1}\}$.

Our setting, however, is more general, as the sequence of tasks may be chosen in an *adversarial* (instead of stochastic [3, 26]) way. A consequence of is that there is no minimum task sampling probability as in previous work (such as the quantity $p_{\min}$ in [3]). Furthermore, we do not assume knowledge of the number of distinct MDPs, $C = |\mathcal{M}|$, or an upper bound of C. All these distinctions make the setting more applicable to capture a broader range of problems.

While provably efficient exploration-exploitation tradeoffs have been extensively studied in single-task RL [10, 23], there is an additional, similar trade-off at the task level in lifelong learning. This arises because the agent does not know in advance if the new task is identical[2] to a previously solved MDP, or if it is a novel MDP. The only way to identify similarity between the new task and previous ones is to explore the task sufficiently. The aim of such exploration is not to maximize reward in the current task, but to infer task identity (which may not help maximize total reward in the current task). Therefore, a lifelong learning agent needs to mix and balance such task-level exploration with within-task exploration that is common in single-task RL.

This observation inspired our abstraction of the online discovery problem, which we now apply to lifelong RL. Here, the exploration action ($A_t = 1$ in Section 2) corresponds to doing complete exploration in the current task, while the exploitation action ($A_t = 0$) corresponds to applying transferred knowledge to accelerate learning. We employ FORCEDEXP for this case, which is outlined in Algorithm 2 of Section A. Overloading terminology, we will also use FORCEDEXP to refer to FORCEDEXP applied to continuous lifelong RL. At round t, if exploration is to happen, it performs PAC-EXPLORE [8] (Algorithm 1 of Section A) to get an accurate model for M_t in all states, which allows it to discover a new distinct MDP from $\mathcal{M}$. If the empirical MDP, $\hat{M}_t$, is considered new[3], it is added to the set $\hat{\mathcal{M}}$ of discovered MDPs. If exploration is *not* to happen, the agent assumes M_t is in $\hat{\mathcal{M}}$, and follows the Finite-Model-RL algorithm [3], which is an extension of RMAX to work with finitely many MDP models. Due to space limitation, full algorithmic details are given in Section A.

In its current form, the algorithm chooses A_t for task M_t before seeing any data collected by acting in M_t. It is straightforward to change the algorithm so that it can switch from an exploitation mode ($A_t = 0$) to exploration after collecting data in M_t, if there is sufficient evidence that M_t is different from all MDPs already found in $\hat{\mathcal{M}}$. Although this change does not improve worst-case sample complexity, it can be beneficial in practice. On the other hand, switching from exploration to exploitation is in general not helpful, as shown in the following example. Let $\mathcal{S} = \{s\}$ contains a single state s, so that $P(s|s) = 1$, and MDPs in $\mathcal{M}$ differ only in their reward functions. Suppose at some step t, the agent has discovered a set of distinct MDPs $\hat{\mathcal{M}}$ from the past, and chooses to do exploration ($A_t = 1$). After taking some steps in M_t, if the agent decides to switch to exploitation before making every action known, there is a risk of under-exploration: M_t may be a new MDP not encountered before, but it has the same rewards on optimal actions for already-discovered MDPs in $\hat{\mathcal{M}}$, but the optimal action in M_t yields even higher reward. By discontinuing exploration, an agent may fail to find the real optimal action in M_t and suffers high sample complexity.

3.3 Sample Complexity Analysis

This section gives a sample-complexity analysis for the lifelong algorithm in the previous subsection. For convenience, we use θ_M to denote the dynamics of an MDP $M \in \mathcal{M}$: for each (s, a), $\theta_M(\cdot|s, a)$ is an $(S + 1)$-dimensional vector, with the first S components giving the transition probabilities to corresponding next states, $P(s'|s, a)$, and the last component the average immediate reward, $R(s, a)$. The model difference in (s, a) between M and M', denoted $\|\theta_M(\cdot|s, a) - \theta_{M'}(\cdot|s, a)\|$, is the ℓ_2-distance between the two vectors. Finally, we let N be an upper bound on the number of next states in the transition models in all MDPs $M \in \mathcal{M}$; $N \leq S$ and can be much smaller in many problems.

The following assumptions are needed:

[2]Or has a near-identical MDP model that leads to ϵ-optimal policies of a prior task.

[3]Specifically, we check if the new MDP parameters differ by at least Γ (an input parameter) in at least one state–action pair to all existing MDPs. If so, we add the MDP to the set. More details about Γ are given shortly.

1. There exists a known quantity $\Gamma > 0$ such that for every two distinct MDPs $M, M' \in \mathcal{M}$, there exists some (s, a) so that $\|\theta_M(\cdot|s, a) - \theta_{M'}(\cdot|s, a)\| > \Gamma$;
2. There is a known diameter D, such that for every MDP in $\mathcal{M}$, any state s' is reachable from any state s in at most D steps *on average*;
3. There are $H \geq H_0 = O\left(SAN \log \frac{SAT}{\delta} \max\{\Gamma^{-2}, D^2\}\right)$ steps per task.

The first assumption requires two distinct MDPs differ by a sufficient amount in their dynamics in at least one state–action pair, and is made for convenience to encode prior knowledge about Γ. Note that if Γ is not known beforehand, one can set $\Gamma = \Gamma_0 = O\left(\frac{\epsilon(1-\gamma)}{\sqrt{N}V_{\max}}\right)$: if two MDPs differ by no more than Γ_0 in every state–action pair, an ϵ-optimal policy in one MDP will be an $O(\epsilon)$-optimal policy in another. The second and third assumptions are the major ones needed in our analysis. The diameter, introduced by [10], and the long enough horizon H together make it possible for an agent to uncover whether the current task has been discovered before.

With these assumptions, the main result is as follows. Note that tighter bounds (in $1/(1-\gamma)$ etc.) for ρ_0 and ρ_1 should be possible by leveraging the refined but more involved single-task analysis of [14], which we defer to future work.

Theorem 5. *Let Algorithm 2 be run for a sequence of T tasks, each of which is from a set $\mathcal{M}$ of C MDPs. Then, with probability at least $1 - \delta$, the* expected *number of steps in which the policy is not ϵ-optimal across all T tasks is*

$$\tilde{O}\left(\rho_0 T + C\rho_3 T^{\alpha} + \frac{\rho_1}{1-\alpha} T^{1-\alpha}\right),$$

where
$$\rho_0 = \frac{CD}{\Gamma^2}, \quad \rho_1 = \rho_2 = \frac{SANV_{\max}^3}{\epsilon^3(1-\gamma)^3}, \quad \rho_3 = H\,. \tag{1}$$

The theorem suggests a substantial reduction in sample complexity with Algorithm 2: while single-task RL typically has a per-task sample complexity ζ_s that at least scales linearly with SA and with similar dependence on ϵ, $1/(1-\gamma)$ and $V_{\max}$ as ρ_1. It is worth noting a subtle difference between between Theorems 5 and a prior result [3]. The previous result [3] gives a high-probability bound on the actual sample complexity, while Theorem 5 gives a high-probability bound on the *expected* sample complexity. However, this technical difference may not matter much in reality when $T \gg 1$.

The proof proceeds by analyzing the sample complexity bounds for all four possible cases (as in the online discovery problem) when solving the tth MDP, and then combining them with Theorem 2 to yield the desired results. The major steps is to ensure that when exploration happens ($A_t = 1$), the identity of M_t will be uncovered successfully (with high probability). This is achieved by a couple of key technical lemmas below. A detailed proof is given in Section D.3.

The first lemma ensures all state–actions can be visited sufficiently often in finitely many steps, when the MDP has a small diameter. The proof is in Section D.1.

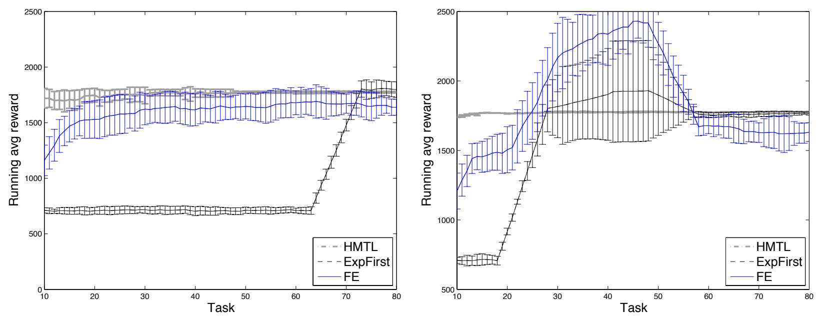
Lemma 6. *For a given MDP,* PAC-EXPLORE *with known threshold m_e will visit all state–action pairs at least m_e times in no more than $H_0(m_e) = O(SADm_e)$ steps with probability at least $1 - \delta$, where $m_e \geq m_0 = O\left(ND^2 \log \frac{N}{\delta}\right)$.*

The second lemma establishes the fact that when PAC-EXPLORE is run on a sequence of T tasks, with high probability, it successfully infers whether M_t has been included in $\hat{\mathcal{M}}$, for every t. This result follows from the Lemma 6 and the assumption involving Γ. The proof is in Section D.2.

Lemma 7. *If the known threshold is set to $m = 72N \log \frac{4SAT}{\delta} \max\{\Gamma^{-2}, D^2\}$ and horizon $H \geq H_0(m)$ (where $H_0(\cdot)$ is given in Lemma 6), then the following holds with probability at least $1 - 2\delta$: for every task in the sequence, the algorithm detects it is a new task if and only if the corresponding MDP has not been seen before.*

4 Experiments

We use a simple grid-world environment with 4 tasks to illustrate the salient properties of FORCEDEXP. All tasks had the same 25-cell square grid layout and 4 actions (up, down, left, right). We provide full details in the supplementary materials, but briefly actions are stochastic, and in each of the 4 MDPs one corner offers high reward (sampled from a Bernoulli with parameter 0.75) and all other rewards are 0. In MDP 4 both the same corner as MDP 3 is rewarding, and the opposite corner is a Bernoulli with parameters 0.99.



(a) One task occurs with low probability. **(b)** Nonstationary task selection.

Figure 1: 10-task smoothed running average of reward per ask. 1 standard deviation error bars are plotted.

We also compare to a Bayesian hierarchical multi-task learning algorithm, or HMTL, of [26]. HMTL learns a Bayesian multi-task posterior distribution across tasks, and leverages as a prior when interacting with each new task. HMTL performs no explicit exploration and has no formal guarantees, but performed well on a challenging real-time strategy game.

We evaluated the algorithms on three illustrative variants of the grid world task:

- *EvenProb*: All four MDPs are sampled with equal probability of 0.25.
- *UnevenProb*: 3 MDPs each have probability 0.31 and 1 MDP has a tiny probability 0.07.
- *Nonstationary*: Across 100 tasks all 4 MDPs have identical frequencies, but MDPs 1–3 appear in phase-one exploration of EXPFIRST, then MDP 4 is shown for 25 tasks, and followed by the only MDPs 1–3.

As expected, all algorithms did well in the $EvenProb$ setting, and we do not further discuss this case here. For the $UnevenProb$ setting, the EXPFIRST algorithm suffers (see Figure 1a) since it must make the first (exploration) phase very long to have a high probability of learning all tasks. In contrast, our new algorithm FORCEDEXP quickly obtains good performance. HMTL also performs well since no explicit exploration is required.

In Figure 1b, an adversary introduces MDP 4 after the initial exploration phase of EXPFIRST completes (which does not violate the minimum probability of each MDP across the entire horizon of tasks, and the upper bound on the number of MDPs given to EXPFIRST). This new task (MDP 4) can obtain similar rewards as MDP 1 using the same policy as for MDP 1, but can obtain higher rewards if the agent explicitly explores in the new domain. Our FORCEDEXP algorithm will randomly explore and identify this new optimal policy, which is why it eventually picks up the new MDP and obtains higher reward. EXPFIRST sometimes successfully infers the task belongs to a new MDP, but only if it happens to encounter the state that distinguished MDPs 1 and 4. HMTL does not explore actively, so it consistently fails to identity MDP 4 as a novel MDP. Consequently, whenever it faces MDP 4, it always learns to use the optimal policy for MDP 1, which is however sub-optimal in MDP 4.

These results suggest FORCEDEXP can have comparable or significantly better performance than prior approaches when there is a highly nonuniform or nonstationary task generation process. These benefits are direct consequences of the effective cross-task exploration built into the algorithm.

5 Conclusions

In this paper, we consider a class of lifelong reinforcement learning problems that capture a broad range of interesting applications. Our work emphasizes the need for effective cross-task exploration that is unique in lifelong learning. This led to a novel online discovery problem, for which we give optimal algorithms with matching upper and lower regret bounds. With this technical tool, we developed a new lifelong RL algorithm, and analyzed its total sample complexity across a sequence of tasks. Our theory quantifies how much gain is obtained by lifelong learning, compared to single-task learning, even if the tasks are adversarially generated. The algorithm was empirically evaluated in a simulated problem, demonstrating its relative strengths compared to prior work.

While we focus on algorithms with formal sample-complexity guarantees, recent work [20] has shown the benefit of a Bayesian approach similar to Thompson sampling for RL domains, and provided Bayesian regret guarantees. As these results rely on an input prior over the MDP, they could easily incorporate a prior learned across multiple tasks. One interesting future direction would be an empirical and theoretical investigation along this line of work for lifelong RL.

References

[1] Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In *NIPS 26*, pages 2220–2228, 2013.

[2] Ronen I. Brafman and Moshe Tennenholtz. R-max—a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 3:213–231, 2002.

[3] Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. In *UAI*, pages 122–131, 2013.

[4] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[5] Sébastien Bubeck, Damien Ernst, and Aurélien Garivier. Optimal discovery with probabilistic expert advice: Finite time analysis and macroscopic optimality. *JMLR*, 14(1):601–623, 2014.

[6] Kai Lai Chung. *A Course in Probability Theory*. Academic Press, 3rd edition, 2000.

[7] Alan Fern, Sriraam Natarajan, Kshitij Judah, and Prasad Tadepalli. A decision-theoretic model of assistance. *JAIR*, 50(1):71–104, 2014.

[8] Zhaohan Guo and Emma Brunskill. Concurrent PAC RL. In *AAAI*, pages 2624–2630, 2015.

[9] David P. Helmbold, Nick Littlestone, and Philip M. Long. Apple tasting. *Information and Computation*, 161(2):85–139, 2000.

[10] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *JMLR*, 11:1563–1600, 2010.

[11] Sham Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, UK, 2003.

[12] Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *MLJ*, 49(2–3):209–232, 2002.

[13] George Konidaris and Finale Doshi-Velez. Hidden parameter markov decision processes: An emerging paradigm for modeling families of related tasks. In *2014 AAAI Fall Symposium Series*, 2014.

[14] Tor Lattimore and Marcus Hutter. PAC bounds for discounted MDPs. In *ALT*, pages 320–334, 2012.

[15] Alessandro Lazaric and Marcello Restelli. Transfer from multiple MDPs. In *NIPS 24*, pages 1746–1754, 2011.

[16] Lihong Li. *A Unifying Framework for Computational Reinforcement Learning Theory*. PhD thesis, Rutgers University, New Brunswick, NJ, 2009.

[17] R. Liu and K. Koedinger. Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities. In *EDM*, 2015.

[18] David A. McAllester and Robert E. Schapire. On the convergence rate of Good-Turing estimators. In *COLT*, pages 1–6, 2000.

[19] Stefanos Nikolaidis, Ramya Ramakrishnan, Keren Gu, and Julie Shah. Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *HRI*, pages 189–196. ACM, 2015.

[20] Ian Osband, Dan Russo, and Benjamin Van Roy. (More) efficient reinforcement learning via posterior sampling. In *NIPS*, pages 3003–3011, 2013.

[21] Mark B. Ring. CHILD: A first step towards continual learning. *MLJ*, 28(1):77–104, 1997.

[22] Jürgen Schmidhuber. PowerPlay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in Psychology*, 4, 2013.

[23] Alexander L. Strehl, Lihong Li, and Michael L. Littman. Reinforcement learning in finite MDPs: PAC analysis. *JMLR*, 10:2413–2444, 2009.

[24] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, March 1998.

[25] Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *JMLR*, 10(1):1633–1685, 2009.

[26] A. Wilson, A. Fern, S. Ray, and P. Tadepalli. Multi-task reinforcement learning: a hierarchical Bayesian approach. In *ICML*, pages 1015–1022, 2007.

A Algorithm Pseudocode

In the following, define $R_{max} := 1$.

Algorithm 1 PAC-EXPLORE Algorithm [8]

```
 0: Input: $m_e$ (known threshold), $D$ (diameter)
 1: Set $L \leftarrow 3D$
 2: while some $(s, a)$ has not been visited at least $m_e$ times do
 3:    Let $s$ be the current state
 4:    if all $a$ have been tried $m_e$ times then
 5:       Start a new $L$-step episode
 6:       Construct an empirical known-state MDP $\hat{M}_K$ with the reward of all known $(s, a)$ pairs set to $0$, all unknown
          set to $R_{max}$, the transition model of all known $(s, a)$ pairs set to the estimated parameters and the unknown
          to self loops
 7:       Compute an optimistic $L$-step policy $\hat{\pi}$ for $\hat{M}_K$
 8:       From the current state, follow $\hat{\pi}$ for $L$ steps, or until an unknown state is reached
 9:    else
10:       Execute $a$ that has been tried the least
11:    end if
12: end while
```

Algorithm 2 Lifelong Learning with FORCEDEXP for Cross-task Exploration

```
 1: Input: $\alpha \in (0, 1]$, $m \in \mathbb{N}$
 2: Initialize $\hat{\mathcal{M}} \leftarrow \emptyset$
 3: for $t = 1, 2, \ldots$ do
 4:    Generate a random number $\xi \sim \mathrm{Uniform}(0, 1)$
 5:    if $\xi < t^{-\alpha}$ then
 6:       Run PAC-EXPLORE to fully explore all states in $M_t$, so that every action is taken in every state for at least
          $m$ times.
 7:       After the above exploration completes, choose actions according to the optimal policy in the empirical model
          $\hat{M}_t$.
 8:       if for all existing models $\hat{M} \in \hat{\mathcal{M}}$, $\hat{M}_t$ has a non-overlapping confidence intervals in some state–action pair
          then
 9:          $\hat{\mathcal{M}} \leftarrow \hat{\mathcal{M}} \cup \{\hat{M}_t\}$
10:       end if
11:    else
12:       Run Finite-Model-RL [3] with $\hat{\mathcal{M}}$
13:    end if
14: end for
```

B Proofs for Section 2

B.1 Proof for Proposition 1

As explained in the text, the expected total loss of EXPFIRST is at most

$$\mathbb{E}[L(\textsc{ExpFirst}, T)] \leq \rho_1 E^* + \rho_2 C + \rho_0 (T - E^*) + \delta\rho_3 T \,,$$

The optimal strategy has the loss of $L^* = \rho_2 C + \rho_0(T - C)$. Therefore, the regret of ExpFirst may be bounded as

$$\begin{aligned} \bar{R}(\textsc{ExpFirst}, T) &= \mathbb{E}[L(\textsc{ExpFirst}, T)] - L^* \\ &\le \rho_1 E^* + \delta\rho_3 T + \rho_0(C - E^*) \\ &< \rho_1 E^* + \delta\rho_3 T \\ &= \frac{\rho_1}{\mu_m}\ln\frac{C}{\delta} + \delta\rho_3 T\,, \end{aligned} \tag{2}$$

where we have made use of the fact that $E^* > C$. The right-hand side of the last equation is a function of δ, in the form of $f(\delta) := a - b\ln\delta + c\delta$, for $a = \frac{\rho_1}{\mu_m}\ln C$, $b = \frac{\rho_1}{\mu_M}$, and $c = \rho_3 T$. Because of convexity of f, its minimum can be found by solving $f'(\delta) = 0$ for δ, giving

$$\delta^* = \frac{b}{c} = \frac{\rho_1}{\rho_3\mu_m T}\,.$$

Substituting δ^* for δ in Equation 2 gives the desired bound.

B.2 Lemma 8

Lemma 8. *Fix $M \in \mathcal{M}$, and let $1 \le t_1 < t_2 < \ldots < t_m \le T$ be the rounds for which $M_t = M$. Then, the expected total loss incurred in these rounds is bounded as:*

$$\bar{L}_M(\textsc{ForcedExp}) < (m\rho_0 + \rho_2 - \rho_3)\bar{L}_1 + (\rho_3 - \rho_0)\bar{L}_2 + \rho_1\bar{L}_3\,,$$

where $\bar{L}_1 := \sum_i \prod_{j<i}(1-\eta_{t_j})\eta_{t_i}$, $\bar{L}_2 := \sum_i \prod_{j<i}(1-\eta_{t_j})\eta_{t_i} \cdot i$, and $\bar{L}_3 := \sum_i \Big(\prod_{j<i}(1-\eta_{t_j})\eta_{t_i}\sum_{j>i}\eta_{t_j}\Big)$.

Proof. Let $\bar{L}_M(\textsc{ForcedExp})$ be the expected total loss incurred in the rounds t where $M_t = M$: $1 \le t_1 < t_2 < \cdots < t_m \le T$ for some $m \ge 0$. Let $I \in \{1, 2, \ldots, m, m+1\}$ be the random variable, so that M is first discovered in round t_I. That is,

$$A_{t_j} = \begin{cases} 0, & \text{if } j < I \\ 1, & \text{if } j = I\,. \end{cases}$$

Note that $I = m + 1$ means M is never discovered; such a notation is for convenience in the analysis below. The corresponding loss is given by

$$(I-1)\rho_3 + \rho_2 + \sum_{j>I}\left(\rho_0\mathbb{I}\left\{A_{t_j} = 0\right\} + \rho_1\mathbb{I}\left\{A_{t_j} = 1\right\}\right)\,,$$

whose expectation, conditioned on I, is at most

$$(I-1)\rho_3 + \rho_2 + \sum_{j>I}\left(\rho_0 + \rho_1\eta_{t_j}\right)\,.$$

Since ForcedExp chooses to explore in step t with probability, we have that

$$\mathbb{P}\left\{I = i\right\} = \prod_{j<i}(1-\eta_{t_j})\eta_{t_i}\,.$$

Therefore, $\bar{L}_M(\textsc{ForcedExp})$ can be bounded by

$$\begin{aligned} &\bar{L}_M(\textsc{ForcedExp}) \\ \le &\sum_{i=1}^{m+1}\mathbb{P}\left\{I = i\right\}\left((I-1)\rho_3 + \rho_2 + \sum_{j>I}\left(\rho_0 + \rho_1\eta_{t_j}\right)\right) \\ = &(m\rho_0 + \rho_2 - \rho_3)\bar{L}_1 + (\rho_3 - \rho_0)\bar{L}_2 + \rho_1\bar{L}_3,\,, \end{aligned}$$

where

$$\bar{L}_1 = \sum_i \prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \,,$$

$$\bar{L}_2 = \sum_i \prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \cdot i \,,$$

$$\bar{L}_3 = \sum_i \left(\prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \sum_{j>i} \eta_{t_j} \right) \,.$$

■

B.3 Proof for Theorem 2

For each $M \in \mathcal{M}$, Lemma 8 gives an upper bound of loss incurred in rounds t for which $M_t = M$:

$$\bar{L}_M(\text{FORCEDEXP}) \le (m\rho_0 + \rho_2 - \rho_3)\bar{L}_1 + (\rho_3 - \rho_0)\bar{L}_2 + \rho_1 \bar{L}_3 \,,$$

where

$$\bar{L}_1 := \sum_i \prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \,,$$

$$\bar{L}_2 := \sum_i \prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \cdot i \,,$$

$$\bar{L}_3 := \sum_i \left(\prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \sum_{j>i} \eta_{t_j} \right) \,.$$

We next bound the three terms of $\bar{L}_M(\text{FORCEDEXP})$, respectively.

To bound $\bar{L}_1$, we define a random variable I, taking values in $\{1, 2, \dots, m, m+1\}$, whose probability mass function is given by

$$\mathbb{P}\{I = i\} = \begin{cases} \prod_{j<i} \left(1-\eta_{t_j}\right) \eta_{t_i}, & \text{if } i \le m \\ \prod_{j \le m} \left(1-\eta_{t_j}\right), & \text{if } i = m+1. \end{cases} \tag{3}$$

Therefore, I is like a geometrically distributed random variable, except that the parameter for the ith draw is not the same and is η_{t_i}. Consequently,

$$\bar{L}_1 = \sum_i \mathbb{P}\{I = i\} \le 1 \,.$$

To bound $\bar{L}_2$, we use the same random variable I:

$$\begin{aligned}
\bar{L}_2 &= \sum_{i=1}^{m} \mathbb{P}\{I = i\} \cdot i \\
&\le \sum_{i=1}^{m} \mathbb{P}\{I \ge i\} \quad \text{(Corollary of Theorem 3.2.1 of [6])} \\
&= \sum_{i=1}^{m} \prod_{j<i} (1-\eta_{t_j}) \quad \text{(By definition of } I \text{ in Equation 3)} \\
&\le \sum_{i=1}^{m} \prod_{j<i} (1-\eta_{t_T}) \quad \text{(By assumption that } \eta_1 \ge \cdots \ge \eta_T) \\
&= \frac{1}{\eta_T}\left(1-(1-\eta_T)^m\right) \le \frac{1}{\eta_T} \,.
\end{aligned}$$

To bound $\bar{L}_3$, we have

$$\bar{L}_3 \le \sum_{i=1}^{m} \prod_{j<i} (1-\eta_{t_j})\eta_{t_i} \sum_{j=1}^{m} \eta_{t_j} = \bar{L}_1 \sum_{j=1}^{m} \eta_{t_j} \le \sum_{j=1}^{m} \eta_{t_j} \,.$$

Putting all three bounds above, we have

$$\bar{L}_M(\textsc{ForcedExp}) \le m\rho_0 + \frac{\rho_3 - \rho_0}{\eta_T} + \rho_1 \sum_{j=1}^{m} \eta_{t_j} \,.$$

Now sum up all $\bar{L}_M(\textsc{ForcedExp})$ over all $M \in \mathcal{M}$, and we have

$$\begin{aligned}\mathbb{E}[L(\textsc{ForcedExp}, T)] &= \sum_{M\in\mathcal{M}} \bar{L}_M(\textsc{ForcedExp}) \\ &\le \rho_0 T + \frac{C\rho_3}{\eta_T} + \rho_1 \sum_{i=1}^{T} \eta_{t_i} \,.\end{aligned}$$

B.4 Proof for Corollary 3

For polynomial exploration rates $\eta_t = t^{-\alpha}$, we have

$$\begin{aligned}\sum_{t=1}^{T} \eta_t &= 1 + \sum_{t=2}^{T} t^{-\alpha} \\ &\le 1 + \int_1^T t^{-\alpha} dt \\ &= 1 + \frac{1}{1-\alpha}\, t^{1-\alpha}\Big|_{t=1}^{T} \\ &= 1 + \frac{1}{1-\alpha}\left(T^{1-\alpha} - 1\right) \\ &\le \frac{T^{1-\alpha}}{1-\alpha} \,.\end{aligned}$$

The total regret follows immediately from Theorem 2. Furthermore, if one sets $\alpha = 1/2$, the regret bound becomes $O(C\rho_3 + 2\rho_1)\sqrt{T} = O(\sqrt{T})$.

For constant exploration rates $\eta_t \equiv \eta$, the regret follows directly from Theorem 2:

$$\frac{C\rho_3}{\eta} + \rho_1 \sum_{t=1}^{\eta} = \frac{C\rho_3}{\eta} + \eta\rho_1 T \,.$$

The optimal η value to minimize the right-hand side above is $\eta = \sqrt{\frac{C\rho_3}{\rho_1 T}}$, which yields the final regret bound of $2\sqrt{C\rho_1\rho_3 T} = O(\sqrt{T})$.

Optimizing η in constant exploration when T is unknown. The optimal η value above depends on T. If T is unknown, the following standard doubling trick may be used to yield the same $O(\sqrt{T})$ regret bound. The idea is to guess the value of T, and then double the guess when the number of steps exceeds the current guess. Concretely, let m be the integer such that

$$2^m - 1 = \sum_{\tau=0}^{m-1} 2^\tau < T \le \sum_{\tau=0}^{m} 2^\tau = 2^{m+1} - 1 \,.$$

Let phase τ consist of steps t between $2^{\tau-1}$ (exclusively) and 2^τ (inclusively); that is, $t \in (2^{\tau-1}, 2^\tau]$. In phase τ, the guess of T is $T_\tau = 2^\tau$, and the learning rate can be set at $\eta_\tau = 1/\sqrt{T_\tau} = \sqrt{2^{-\tau}}$ in those steps. Theorem 2 implies the total regret in phase τ is at most

$$\frac{C\rho_3}{\eta_\tau} + \rho_1\eta_\tau(2^\tau - 2^{\tau-1}) = (C\rho_3 + \rho_1/2)\sqrt{2^\tau}\,.$$

Therefore, the total regret across all T steps is at most

$$\begin{aligned}\sum_{\tau=0}^{m}(C\rho_3 + \rho_1/2)\sqrt{2^\tau} &\leq (2+\sqrt{2})(C\rho_3 + \frac{\rho_1}{2})2^{m/2} \\ &\leq (2+\sqrt{2})(C\rho_3 + \frac{\rho_1}{2})\sqrt{T} = O(\sqrt{T})\,,\end{aligned}$$

where we have used the condition that $2^m - 1 < T$ in the second inequality.

B.5 Proof for Theorem 4

We construct a stochastic online discovery problem with $\mathcal{M} = \{1, 2, \ldots, C\}$ and distribution μ so that

$$\mu(M) = \begin{cases}\mu_m, & \text{if } M < C \\ 1 - C\mu_m, & \text{if } M = C\,,\end{cases}$$

where $\mu_m = 1/\sqrt{T} \ll 1$. For every $M \in \mathcal{M}$, define $T_M \in \{1, \ldots, T, \infty\}$ as the first time M is discovered; that is

$$T_M = \min\{t \mid M_t = M, A_t = 1\}\,.$$

Furthermore, let $1 \leq t_1 < t_2 < \cdots < t_E \leq T$ be the rounds in which exploration happens (that is, $A_t = 1$); denote by $\mathcal{E}$ the set $\{t_1, t_2, \ldots, t_E\}$. Since the two random variables M_t and A_t are independent, conditioned on H_t, we have for any $i \in \{1, 2, \ldots, E\}$ and any $M \in \mathcal{M}$ that

$$\mathbb{P}\{M_{t_i} = M\} = \mu(M)\,.$$

Conditioning on $\mathcal{E}$ being the rounds of exploration, we want to lower bound the number E of exploration rounds so that the probability of discovering *all* items in $\mathcal{M}$ is at most δ (which is necessary for the expected regret to be $O(\sqrt{T})$). First, note that the events $\{T_M < \infty\}_{M\in\mathcal{M}}$ are negatively correlated, since discovering some M_1 in $\mathcal{E}$ can only decrease the probability of discovering $M_2 \neq M_1$ in $\mathcal{E}$. Therefore, we have

$$\mathbb{P}\{\forall M, T_M < \infty\} \leq \prod_{M\in\mathcal{M}} \mathbb{P}\{T_M < \infty\} \leq (1 - (1-\mu_m)^E)^{C-1}\,.$$

Making the last expression to be $1 - \delta$, we have

$$\begin{aligned}E &= \frac{\ln\left(1 - (1-\delta)^{\frac{1}{C-1}}\right)}{\ln(1-\mu_m)} \\ &= \Omega\left(\frac{\ln\left(1 - (1 - \frac{\delta}{C})\right)}{-\mu_m}\right) \\ &= \Omega\left(\frac{1}{\mu_m}\ln\frac{C}{\delta}\right)\,,\end{aligned}$$

for sufficiently small μ_m and δ.

For simplicity, assume $\rho_0 = 0$ without loss of generality; otherwise, we can just define a related problem with $\rho_i' := \rho_i - \rho_0$, where the loss is just shifted by a constant and the regret remains unchanged. With this assumption, the optimal expected loss is $C\rho_2$ (*c.f.*, $L^*(T)$).

The expected loss of the is now at least $(E - C)\rho_1 + C\rho_2 + \delta(T - E)\mu_m\rho_3$, where the first two terms are for the loss incurred during the E exploration rounds; and the last term for the δ-probability event that some item is not discovered in the exploration rounds, which leads to ρ_3 loss when it is encountered in any of the remaining $T - E$ rounds.

Finally, the regret can be readily computed as

$$(E - C)\rho_1 + \delta(T - E)\mu_m\rho_3 = \Omega\left(\rho_3 T\mu_m + \frac{\rho_2}{\mu_m}\ln\frac{C}{\delta}\right)\,,$$

completing the proof with the fact that $\mu_m = 1/\sqrt{T}$.

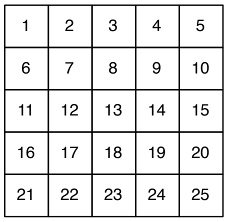


Figure 2: The gridworld domain used in experiments. See text for further description.

C Experiment Details

All four MDPs had the same 25-cell square grid layout and 4 actions (up, down, left, right), as shown in Figure 2. Actions succeed in their intended direction with probability 0.85 and with probability 0.05 go in the other directions (unless halted by a wall). For all actions corner states $s5$,$s20$, and $s25$ stay in the same state with probability 0.95 or transition back to the start state (for all actions). The start state is at the center of the square grid ($s13$). The dynamics of all MDPs are identical. All rewards are sampled from binomial distributions. All rewards have parameter 0.0 unless otherwise noted. In MDP 1, corner state $s20$ has a reward parameter of 0.75. In MDP 2, corner state $s5$ has a reward parameter of 0.75. In MDP 3, corner state $s25$ has a reward parameter of 0.75. In MDP 4, corner state $s25$ has a reward parameter of 0.75 and corner state $s1$ has a reward parameter of 0.99.

We also compare to a Bayesian hierarchical multi-task learning algorithm, or HMTL, of [26]. HMTL learns a Bayesian multi-task posterior distribution across tasks, and leverages as a prior when interacting with each new task. HMTL selects the MAP estimate of the model parameters, computes a policy for this models, executes for a fixed number of steps, updates the posterior for the current task, and repeats. No formal guarantees are provided, although empirically HMTL did well on a challenging real-time strategy game.

For HMTL, we set the interval between recomputing the MAP model parameters at 20 steps: this choice was made after informal experimentation suggested this value improved performance compared to longer intervals.

In all cases, EXPFIRST is given an upper bound on the number of MDPs (4) and the minimum probability of any of the MDPs across the 100 tasks. HMTL is also provided with an upper bound on the number of MDPs, though HMTL is also capable of learning this directly, and HMTL is used only with a two-level hierarchy (e.g. a class consists of a single MDP). For FORCEDEXP, we compare two variants. The approach labeled "FE" in the figures uses a polynomially decaying exploration rate, t^{α} with $\alpha = 0.5$, for all experiments. Performance does vary with the choice of α but $\alpha = 0.5$ gave good results in our preliminary investigations. Interestingly, this is consistent with the theoretical result that $\alpha = 0.5$ minimizes dependence on T for polynomially decaying exploration rates (*c.f.*, Corollary 3). We also evaluated the FORCEDEXP algorithm using a constant exploration rate $\frac{2}{\sqrt{T}}$ for some earlier experiments: as expected performance was similar but slightly worse generally than using a decaying exploration rate, and so we focus the comparison on the decaying exploration rate variant.

D Proofs for Section 3

D.1 Proof of Lemma 6

Proof. The proof follows closely to that of [8]. Consider the beginning of an episode, and let K be the set of known state–action pairs which have been visited by the agent at least m_e times. For each $(s, a) \in k$, the ℓ_1 distance between the empirical estimate and the actual next-state distribution is at most [11] ([Lemma 8.5.5]): $\alpha = \sqrt{\frac{8N}{m_e} \log \frac{2N}{\delta}}$. Let M_K be the known-state MDP, which is identical to $\hat{M}_K$ except that the transition probabilities are replaced by the true ones for known state–action pairs. Following the same line of reasoning as [8], one may lower-bound the probability that an unknown state is reached within the episode by $p_e \geq 1/6 - 3\alpha D$. Therefore, p_e is bounded by $1/12$ as long as $\alpha D \leq 1/36$. The latter is guaranteed if $m_e \geq m_0 = O\left(ND^2 \log \frac{N}{\delta}\right)$. The rest of the proof is the same as [8], invoking Lemma 56 of [16] to get an upper bound of H, as stated in the lemma as $H_0(m)$. ■

D.2 Proof of Lemma 7

Proof. For task M_t, let $\mathcal{E}_t$ be the event that all state–action pairs become known after H steps; Lemma 6 with a union bound shows all events $\{\mathcal{E}_t\}_{t\in\{1,2,\dots,T\}}$ hold with probability at least $1-\delta$. For every fixed t, under event $\mathcal{E}_t$, every state–action pair has at least m samples to estimate its transition probabilities and average reward after H steps. Applying Lemma 8.5.5 of [11] on the transition distribution, we can upper bound, with probability at least $1-\frac{\delta}{2SAT}$, the ℓ_1 error of the transition probability estimates by:

$$\epsilon_T = \sqrt{\frac{8N}{m}\log\frac{4SAT}{\delta}} \le \frac{\Gamma}{3}\,.$$

Similarly, an application of Hoeffding's inequality gives the following upper bound, with probability at least $1-\frac{\delta}{2SAT}$, on the reward estimate:

$$\epsilon_R = \sqrt{\frac{2}{m}\log\frac{4SAT}{\delta}} \le \frac{\Gamma}{6\sqrt{N}}\,.$$

Applying a union bound over all states, actions, and tasks, the above concentration results hold with probability at least $1-\delta$ for an agent running on T tasks. The rest of the proof is to show that task identification succeeds when the above concentration inequalities hold.

To do this, consider the following two mutually exclusive cases:

1. If M_t is new, then, by assumption, for every $M' \in \hat{\mathcal{M}}$, there exists some (s,a) for which the two models differ by at least Γ in ℓ_2 distance; that is, $\|\theta_{M_t}(\cdot|s,a)-\theta_{M'}(\cdot|s,a)\|_2 \ge \Gamma$. It follows from the equality,

$$\begin{aligned}&\|\theta_{M_t}(\cdot|s,a)-\theta_{M'}(\cdot|s,a)\|_2^2\\ &= \sum_{1\le s'\le S}\left(\theta_{M_t}(s'|s,a)-\theta_{M'}(s'|s,a)\right)^2 \quad \text{[error in transition probability estimates]}\\ &\quad + \left(\theta_{M_t}(S+1|s,a)-\theta_{M'}(S+1|s,a)\right)^2\,, \quad \text{[error in reward estimate]}\end{aligned}$$

   that at least one of two terms on the right-hand side above is at least $\Gamma^2/2$.
   If the first term is larger than $\Gamma^2/2$, then the ℓ_1 distance between the two next-state transition distributions is at least $\Gamma/\sqrt{2}$, which is larger than $2\epsilon_T = 2\Gamma/3$. It implies that the ℓ_1-balls of transition probability estimates for (s,a) between M_t and M' do not overlap, and we will identify M_t as a new MDP. Similarly, if the second term is larger than $\Gamma^2/2$, then using ϵ_R we can still identify M_t as a new MDP.
2. If M_t is not new, we claim that the algorithm will correctly identify it as some previously solved MDP, say $M'' \in \hat{\mathcal{M}}$. In particular, confidence intervals of its estimated model in every state–action pair must overlap with M'', since both models' confidence intervals contain the true model parameters. On the other hand, for any $M' \in \hat{\mathcal{M}} \setminus \{M''\}$, its model estimate's confidence intervals do not have overlap with that of M_t's in at least one state–action pair, as shown in case 1. Therefore, the algorithm can find the unique and correct $M'' \in \hat{\mathcal{M}}$ that is the same as M_t.

Finally, the lemma is proved with a union bound over all tasks, states and actions, and with the probability that E_t fails to hold for some t. ■

D.3 Proof of Theorem 5

Proof. We consider each possible case when solving the tth task, M_t. As shown in previous lemma, with probability $1-\delta$, the following event $\mathcal{E}_t$ hold for all $t\in[T]$: after PAC-EXPLORE is run on M_t, Algorithm 2 will discover the identity of M_t correctly. That is, if M_t is a new MDP, it will be added to $\hat{\mathcal{M}}$; otherwise, $\hat{\mathcal{M}}$ remains unchanged. In the following, we assume $\mathcal{E}_t$ holds for every t, and consider the following cases:

(a) Exploitation in discovered tasks: we choose to exploit (line 12 in Alg 2) and M_t has been already discovered. In this case, Finite-Model-RL is used to do model elimination (within $\hat{\mathcal{M}}$) and to transfer samples from previous tasks that correspond to the same MDP as the current task M_t. Therefore, with a similar analysis, we can get a per-task sample complexity of at most $O(CDm) = \tilde{O}(\frac{CD}{\Gamma^2}) = \rho_0$.

(b) Exploitation in undiscovered tasks: we choose to exploit and M_t has *not* been discovered. Running Finite-Model-RL in this case can end up with an arbitrarily poor policy which follows a non-ϵ-optimal policy in every step. Therefore, the sample complexity can be as large as $H = \rho_3$.

(c) Exploration: we choose to explore using PAC-EXPLORE (lines 6—9 in Alg 2). In this case, with high probability, it takes at most $H_0(m)$ steps to make every state known, so that the model parameters can be estimated to within accuracy $O(\Gamma)$. After that, we can reliably decide whether M_t is a new MDP or not. With sample transfer, the additional steps where ϵ-sub-optimal policies are taken in the MDP corresponding to M_t (accumulated across all tasks in the T-sequence) is at most ζ_s, the single-task sample complexity. The total sample complexity for tasks corresponding to this MDP is therefore at most $H_0(m)T(M_t) + \zeta_s = \rho_2 T(M_t) + \zeta_s$, where $T(M_t)$ is the number of times this MDP occurs in the T-sequence.

Finally, when Algorithm 2 is run on a sequence of T tasks, the total sample complexity—the number of steps in all tasks for which the agent does not follow an ϵ-optimal policy—is given by one of the three cases above. The expected sample complexity can therefore be upper bounded by evoking Theorem 2, completing the proof with an application of union bound that takes care of error probabilities. ■