Tractable Offline Learning of Regular Decision Processes

Ahana Deb
Universitat Pompeu Fabra
ahana.deb@upf.edu

Roberto Cipollone
Sapienza University of Rome
cipollone@diag.uniroma1.it

Anders Jonsson
Universitat Pompeu Fabra
anders.jonsson@upf.edu

Alessandro Ronca
University of Oxford
alessandro.ronca@cs.ox.ac.uk

Mohammad Sadegh Talebi
University of Copenhagen
m.shahi@di.ku.dk

Abstract

This work studies offline Reinforcement Learning (RL) in a class of non-Markovian environments called Regular Decision Processes (RDPs). In RDPs, the unknown dependency of future observations and rewards from the past interactions can be captured by some hidden finite-state automaton. For this reason, many RDP algorithms first reconstruct this unknown dependency using automata learning techniques. In this paper, we show that it is possible to overcome two strong limitations of previous offline RL algorithms for RDPs, notably RegORL [14]. This can be accomplished via the introduction of two original techniques: the development of a new pseudometric based on formal languages, which removes a problematic dependency on L^{p}_{∞}-distinguishability parameters, and the adoption of Count-Min-Sketch (CMS), instead of naive counting. The former reduces the number of samples required in environments that are characterized by a low complexity in language-theoretic terms. The latter alleviates the memory requirements for long planning horizons. We derive the PAC sample complexity bounds associated to each of these techniques, and we validate the approach experimentally.

1 Introduction

The Markov assumption is fundamental for most Reinforcement Learning (RL) algorithms, requiring that the immediate reward and transition only depend on the last observation and action. Thanks to this property, the computation of (near-)optimal policies involves only functions over observations and actions. However, in complex environments, observations may not be complete representations of the internal environment state. In this work, we consider RL in Non-Markovian Decision Processes (NMDPs). In these very expressive models, the probability of future observations and rewards may depend on the entire history, which is the past interaction sequence composed of observations and actions. The unrestricted dynamics of the NMDP formulation is not tractable for optimization. Therefore, previous work in non-Markovian RL focus on tractable subclasses of decision processes. In this work, we focus on Regular Decision Processes (RDPs) [11, 12]. In RDPs, the distribution of the next observation and reward is allowed to vary according to regular properties evaluated on the history. Thus, these dependencies can be captured by a Deterministic Finite-state Automaton (DFA). RDPs are expressive models that can represent complex dependencies, which may be based on events that occurred arbitrarily back in time. For example, we could model that an agent may only enter a restricted area if it has previously asked for permission and the access was granted.

Due to the properties above, RL algorithms for RDPs are very general and applicable. However, provably correct and sample efficient algorithms for RDPs are still missing. On one hand, local

17th European Workshop on Reinforcement Learning (EWRL 2024).

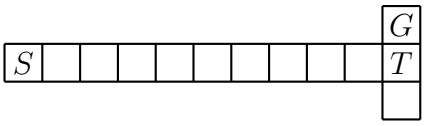


Figure 1: T-maze [3] with corridor length $N = 10$. The observation produced at the initial position S indicates the position of the goal G at the end of the corridor for the current episode.

optimization approaches are generally more efficient, but lack correctness guarantees. In this group, we recall Abadi and Brafman [1], Toro Icarte et al. [64] and all RL algorithms with policy networks that can operate on sequences. On the other hand, algorithms with formal guarantees do not provide a practical implementation [14, 55] or can only be applied effectively in small environments [56]. In this work, we propose a new offline RL algorithm for RDPs with a Probably Approximately Correct (PAC) sample complexity guarantee. Our algorithm improves over previous work in three ways. First, we overcome a limitation of previous sample complexity bounds, which is the dependence on a parameter called L_∞^{p}-distinguishability. This is desirable because there exist simple non-Markovian domains in which this parameter decays exponentially with respect to the number of RDP states; an example is the T-maze by Bakker [3], discussed below. Second, a careful treatment of the individual features that compose the trace and each observation allows us to further improve the efficiency. Third, inspired by the automaton-learning algorithm FlexFringe [7], we use a data structure called Count-Min-Sketch (CMS) [16] to compactly represent probability distributions on large sets.

Example 1 (T-maze)**.** As a running example, we consider the T-Maze domain [3], a deterministic non-Markovian grid-world environment. An agent has to reach a goal G from an initial position S in a corridor of length N that terminates with a T-junction as shown in Figure 1. The agent can move one cell at a time, taking actions $North$, $South$, $East$, or $West$. In each episode, the rewarding goal G can be in the cell above or below the T-junction. Depending on the position of the goal, the observation in state S is 011 or 110. In the corridor the observation is 101, and at the T-junction the observation is 010. This means that, when crossing the corridor, the agent cannot observe the current location or the goal position. This yields a history-dependent dynamics that cannot be modeled with an MDP or any k-order MDP. As we show later, this domain can be expressed as an RDP.

Contributions We study offline RL for Regular Decision Processes (RDPs) and address limitations of previous algorithms. To do so, we develop a novel language metric $L_{\mathcal{X}}$, based on the theory of formal languages, and define a hierarchy of language families that allows for a fine-grained characterization of the RDP complexity. Unlike previous algorithms, the RL algorithm we develop based on this language hierarchy does not depend on L_∞^{p}-distinguishability and is exponentially more sample efficient in domains having low complexity in language-theoretic terms. In addition, we reduce the space complexity of the algorithm and modify REGORL with the use of Count-Min-Sketch. To validate our claims, we provide a theoretical analysis for both variants, when the $L_{\mathcal{X}}$-distinguishability or CMS is used. Finally, we provide an experimental analysis of both approaches.

1.1 Related work

RL in RDPs The first online RL algorithm for RDPs is provided in Abadi and Brafman [1]. Later, Ronca and De Giacomo [55] and Ronca et al. [56] developed the first online RL algorithms with sample complexity guarantees. The algorithm and the sample complexity bound provided in Ronca and De Giacomo [55] adapts analogous results from automata learning literature [4, 5, 6, 15, 48, 54]. RegORL from Cipollone et al. [14] is an RL algorithm for RDPs with sample complexity guarantees for the offline setting. In this work, we study the same setting and improve on two significant weaknesses of RegORL, regarding the sample compexity and the space requirements. The details of these improvements are discussed in the following sections. Lastly, the online RL algorithm in Toro Icarte et al. [64] can also be applied to RDPs, but it is not proven to be sample efficient.

Non-Markovian RL Apart from RDP algorithms, one can apply RL methods to more general decision processes. Indeed, the automaton state of an RDP can be seen as an information state, as defined in [60]. As shown in [11], any RDP can also be expressed as a POMDP whose hidden dynamics evolves according to its finite-state automaton. Therefore, any RL algorithm for generic POMDPs can also be applied to RDPs. Unfortunately, planning and learning in POMDPs is intractable

[32, 49]. More efficient learning algorithms, with more favorable complexity bounds, have been obtained for subclasses of POMDPs, such as undercomplete POMDPs [21, 29], few-step reachability [22], ergodicity [2], few-step decodability [18, 32], or weakly-revealing [37]. However, none of these assumptions exhaustively capture the entire RDP class, and they cannot be applied in general RDPs.

Regarding more general non-Markovian dynamics, Predictive State Representations (PSRs) [9, 28, 33, 59] are general descriptions of dynamical systems that capture POMDPs and therefore RDPs. There exist polynomial PAC bounds for online RL in PSRs [73]. However, these bounds involve parameters that are specific to PSRs and do not immediately apply to RDPs. Feature MDPs and state representations both share the idea of having a map from histories to a state space. This is analogous to the map determined by the transition function of the automaton underlying an RDP. Algorithmic solutions for feature MDPs are based on suffix trees, and they cannot yield optimal performance in our setting [27, 68]. The automaton of an RDP can be seen as providing one kind of state representation [39, 40, 41, 45, 47]. The existing bounds for state representations show a linear dependency on the number of candidate representations, which is exponential in the number of states in our case. A similar dependency is also observed in [34]. With respect to temporal dependencies for rewards, Reward Machines consider RL with non-Markovian rewards [8, 17, 20, 26, 63, 70]. However, this dependency is usually assumed to be known, which makes the Markovian state computable. Lastly, non-Markovianity is also introduced by the logical specifications that the agent is required to satisfy [10, 19, 23, 24, 25]; however, it is resolved a priori from the known specification.

Offline RL in MDPs. There is a rich and growing literature on offline RL, and provably sample efficient algorithms have been proposed for various settings of MDPs [13, 30, 35, 52, 53, 65, 67, 69, 71, 72]. For example, in the case of episodic MDPs, it is established that the optimal sample size in offline RL depends on the size of state-space, episode length, as well as some notion of concentrability, reflecting the distribution mismatch between the behavior and optimal policies. A closely related problem is off-policy learning [31, 38, 62, 66].

2 Preliminaries

Notation Given a set $\mathcal{Y}$, $\Delta(\mathcal{Y})$ denotes the set of probability distributions over $\mathcal{Y}$. For a function $f : \mathcal{X} \to \Delta(\mathcal{Y})$, $f(y \mid x)$ is the probability of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$. Further, we write $y \sim f(x)$ to abbreviate $y \sim f(\cdot \mid x)$. Given an event E, $\mathbb{I}(E)$ denotes the indicator function of E, which equals 1 if E is true, and 0 otherwise. For any pair of integers m and n such that $0 \leq m \leq n$, we let $[m, n] := \{m, \ldots, n\}$ and $[n] := \{1, \ldots, n\}$. The notation $\widetilde{\mathcal{O}}(\cdot)$ hides poly-logarithmic terms.

Count-Min-Sketch Count-Min-Sketch, or CMS [16], is a data structure that compactly represents a large non-negative vector $v = [v_1, \ldots, v_m]$. CMS takes two parameters δ_c and ε as input, and constructs a matrix C with $d = \lceil \log \frac{1}{\delta_c} \rceil$ rows and $w = \lceil \frac{e}{\varepsilon} \rceil$ columns. For each row $j \in [d]$, CMS picks a hash function $h_j : [m] \to [w]$ uniformly at random from a pairwise independent family [44]. Initially, all elements of v and C equal 0. An update (i, c) consists in incrementing the element v_i by $c > 0$. CMS approximates an update (i, c) by incrementing $C(j, h_j(i))$ by c for each $j \in [d]$. At any moment, a point query $\widetilde{v}_i$ returns an estimate of v_i by taking the minimum of the row estimates, i.e. $\widetilde{v}_i = \min_j C(j, h_j(i))$. It is easy to see that $\widetilde{v}_i \geq v_i$, i.e. CMS never underestimates v_i.

2.1 Languages and operators

An *alphabet* Γ is a finite non-empty set of elements called *letters*. A *string* over Γ is a concatenation $a_1 \cdots a_\ell$ of letters from Γ, and we call ℓ its length. In particular, the string containing no letters, having length zero, is a valid string called the *empty string*, and is denoted by λ. Given two strings $x_1 = a_1 \cdots a_\ell$ and $x_2 = b_1 \cdots b_m$, their concatenation $x_1 x_2$ is the string $a_1 \cdots a_\ell b_1 \cdots b_m$. In particular, for any given string x, we have $x\lambda = \lambda x = x$. The set of all strings over alphabet Γ is written as Γ^*, and the set of all strings of length ℓ is written as Γ^ℓ. Thus, $\Gamma^* = \cup_{\ell \in \mathbb{N}} \Gamma^\ell$. A *language* is a subset of Γ^*. Given two languages X_1 and X_2, their concatenation is the language defined by $X_1 X_2 = \{x_1 x_2 \mid x_1 \in X_1, x_2 \in X_2\}$. When concatenating with a language $\{a\}$ containing a single string consisting of a letter $a \in \Gamma$, we simply write Xa instead of $X\{a\}$. Concatenation is associative and hence we can write the concatenation $X_1 X_2 \cdots X_k$ of an arbitrary number of languages.

Given the fundamental definitions above, we introduce two operators to construct sets of languages. The first operator C_k^ℓ is defined for any two non-negative integers ℓ and $k \in \{1, \ldots, \ell\}$, it takes a set of languages $\mathcal{G}$, and constructs a new set of languages as follows:

$$\mathrm{C}_k^\ell(\mathcal{G}) = \{\Gamma^\ell \cap S_1 G_1 S_2 G_2 \cdots S_k G_k S_{k+1} \mid G_1, \ldots, G_k \in \mathcal{G},\ S_1, \ldots, S_{k+1} \in \mathcal{S}\},$$

where the set $\mathcal{S} = \{\Gamma^*, \{\lambda\}\}$ consists of the language Γ^* of all strings over the considered alphabet and the singleton language $\{\lambda\}$ consisting of the empty string only. In the definition, each S_i can be Γ^* or λ. Choosing $S_i = \Gamma^*$ allows arbitrary letters between a string from G_i and the next string from G_{i+1}, whereas choosing $S_i = \{\lambda\}$ enforces that a string from G_i is immediately followed by a string from G_{i+1}. Also, the intersection with Γ^ℓ amounts to restricting to strings of length ℓ in the languages given by the concatenation.

The second operator we define, B, takes a set of languages $\mathcal{X}$, and it constructs a new set of languages $\mathrm{B}(\mathcal{X}) = \mathcal{X}^\sqcup \cup \mathcal{X}^\sqcap$ by taking combinations as follows:

$$\mathcal{X}^\sqcup = \{X_1 \cup X_2 \mid X_1, X_2 \in \mathcal{X}\}, \qquad \mathcal{X}^\sqcap = \{X_1 \cap X_2 \mid X_1, X_2 \in \mathcal{X}\}.$$

The set $\mathcal{X}^\sqcup$ consists of the pair-wise unions of the languages in $\mathcal{X}$. For example, when $\mathcal{X} = \{\{ac, ad\}, \{ac, bc\}\}$, we have $\mathcal{X}^\sqcup = \mathcal{X} \cup \{\{ac, ad, bc\}\}$. Similarly, the set $\mathcal{X}^\sqcap$ consists of the pair-wise intersections of the languages in $\mathcal{X}$. For example, when $\mathcal{X} = \{\{ac, ad\}, \{ac, bc\}\}$, we have $\mathcal{X}^\sqcap = \mathcal{X} \cup \{\{ac\}\}$. From the previous example we can observe that $\mathcal{X} \subseteq \mathcal{X}^\sqcup, \mathcal{X}^\sqcap$, which holds in general since $X = \{X \cap X \mid X \in \mathcal{X}\} \subseteq \mathcal{X}^\sqcap$ and similarly $X = \{X \cup X \mid X \in \mathcal{X}\} \subseteq \mathcal{X}^\sqcup$ Therefore $\mathcal{X} \subseteq \mathrm{B}(\mathcal{X})$. The operators will later be used to define relevant patterns on episode traces. They are inspired by classes of languages in the first level of the *dot-depth hierarchy*, a well-known hierarchy of star-free regular languages [50, 58, 61].

2.2 Episodic regular decision processes

We first introduce generic episodic decision processes. An episodic decision process is a tuple $\mathbf{P} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R}, H \rangle$, where $\mathcal{O}$ is a finite set of observations, $\mathcal{A}$ is a finite set of actions, $\mathcal{R} \subset [0, 1]$ is a finite set of rewards, and $H \geq 1$ is a finite horizon. We frequently consider the concatenation $\mathcal{A}\mathcal{O}$ of the sets $\mathcal{A}$ and $\mathcal{O}$. Let $\mathcal{H}_t = (\mathcal{A}\mathcal{O})^{t+1}$ be the set of histories of length $t+1$, and let $e_{m:n} \in \mathcal{H}_{n-m}$ denote a history from time m to time n, both included. Each action-observation pair $ao \in \mathcal{A}\mathcal{O}$ in a history has an associated reward label $r \in \mathcal{R}$, which we write $ao/r \in \mathcal{A}\mathcal{O}/\mathcal{R}$. A *trajectory* $e_{0:T}$ is the full history generated until (and including) time T.

We assume that a trajectory $e_{0:T}$ can be partitioned into *episodes* $e_{\ell:\ell+H} \in \mathcal{H}_H$ of length $H + 1$, In each episode $e_{0:H}$, $a_0 = a_\perp$ is a dummy action used to initialize the distribution on $\mathcal{H}_0$. The transition function $\bar{T} : \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{O})$ and the reward function $\bar{R} : \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{R})$ depend on the current history in $\mathcal{H} = \cup_{t=0}^{H} \mathcal{H}_t$. Given $\mathbf{P}$, a generic policy is a function $\pi : (\mathcal{A}\mathcal{O})^* \to \Delta(\mathcal{A})$ that maps trajectories to distributions over actions. The value function $V^\pi : [0, H] \times \mathcal{H} \to \mathbb{R}$ of a policy π is a mapping that assigns real values to histories. For $h \in \mathcal{H}$, it is defined as $V^\pi(H, h) \coloneqq 0$ and

$$V^\pi(t, h) \coloneqq \mathbb{E}\left[\sum_{i=t+1}^{H} r_i \,\middle|\, h, \pi\right], \quad \forall t < H, \forall h \in \mathcal{H}_t. \tag{1}$$

For brevity, we write $V_t^\pi(h) := V^\pi(t, h)$. The optimal value function V^* is defined as $V_t^*(h) \coloneqq \sup_\pi V_t^\pi(h), \forall t \in [H], \forall h \in \mathcal{H}_t$, where sup is taken over all policies $\pi : (\mathcal{A}\mathcal{O})^* \to \Delta(\mathcal{A})$. Any policy achieving V^* is called optimal, which we denote by π^*; namely $V^{\pi^*} = V^*$. Solving $\mathbf{P}$ amounts to finding π^*. In what follows, we consider simpler policies of the form $\pi : \mathcal{H} \to \Delta(\mathcal{A})$ mapping finite histories to distributions over actions. Let $\Pi_\mathcal{H}$ denote the set of such policies. It can be shown that $\Pi_\mathcal{H}$ always contains an optimal policy, i.e. $V_t^*(h) \coloneqq \max_{\pi \in \Pi_\mathcal{H}} V_t^\pi(h), \forall t \in [H], \forall h \in \mathcal{H}_t$. An episodic MDP is an episodic decision process whose dynamics at each timestep t only depends on the last observation and action [51].

Episodic RDPs An episodic Regular Decision Process (RDP) [1, 11, 12] is an episodic decision process $\mathbf{R} = \langle \mathcal{O}, \mathcal{A}, \mathcal{R}, \bar{T}, \bar{R}, H \rangle$ described by a *finite transducer* (Moore machine) $\langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$, where $\mathcal{Q}$ is a finite set of states, $\Sigma = \mathcal{A}\mathcal{O}$ is a finite input alphabet composed of actions and observations, Ω is a finite output alphabet, $\tau : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ is a transition function, $\theta : \mathcal{Q} \to \Omega$ is an output function, and $q_0 \in \mathcal{Q}$ is a fixed initial state [43, 57]. The output space $\Omega = \Omega_\mathsf{o} \times \Omega_\mathsf{r}$

consists of a finite set of functions that compute the conditional probabilities of observations and rewards, on the form $\Omega_{\mathsf{o}} \subset \mathcal{A} \to \Delta(\mathcal{O})$ and $\Omega_{\mathsf{r}} \subset \mathcal{A} \to \Delta(\mathcal{R})$. For simplicity, we use two output functions, $\theta_{\mathsf{o}} : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{O})$ and $\theta_{\mathsf{r}} : \mathcal{Q} \times \mathcal{A} \to \Delta(\mathcal{R})$, to denote the individual conditional probabilities. Let τ^{-1} denote the inverse of τ, i.e. $\tau^{-1}(q) \subseteq \mathcal{Q} \times \mathcal{A}\mathcal{O}$ is the subset of state-symbol pairs that map to $q \in \mathcal{Q}$. An RDP $\mathbf{R}$ implicitly represents a function $\bar{\tau} : \mathcal{H} \to \mathcal{Q}$ from histories in $\mathcal{H}$ to states in $\mathcal{Q}$, recursively defined as $\bar{\tau}(h_0) \coloneqq \tau(q_0, a_0 o_0)$, where a_0 is some fixed starting action, and $\bar{\tau}(h_t) \coloneqq \tau(\bar{\tau}(h_{t-1}), a_t o_t)$. The dynamics and of $\mathbf{R}$ are defined as $\bar{T}(o \mid h, a) = \theta_{\mathsf{o}}(o \mid \bar{\tau}(h), a)$ and $\bar{R}(r \mid h, a) = \theta_{\mathsf{r}}(r \mid \bar{\tau}(h), a), \forall h \in \mathcal{H}, \forall ao/r \in \mathcal{A}\mathcal{O}/\mathcal{R}$. As in previous work [14], we assume that any episodic RDP generates a designated termination observation $o_\perp \in \mathcal{O}$ after exactly H transitions. This ensures that any episodic RDP is acyclic, i.e. the states can be partitioned as $\mathcal{Q} = \mathcal{Q}_0 \cup \cdots \cup \mathcal{Q}_{H+1}$, where each $\mathcal{Q}_{t+1}$ is the set of states generated by the histories in $\mathcal{H}_t$ for each $t \in [0, H]$. An RDP is minimal if its Moore machine is minimal. We use A, R, O, Q to denote the cardinality of $\mathcal{A}, \mathcal{R}, \mathcal{O}, \mathcal{Q}$, respectively, and assume $A \geq 2$ and $O \geq 2$.

Since the conditional probabilities of observations and rewards are fully determined by the current state-action pair (q, a), an RDP $\mathbf{R}$ adheres to the Markov property over its states, but *not over the observations*. Given a state $q_t \in \mathcal{Q}$ and an action $a_t \in \mathcal{A}$, the probability of the next transition is

$$\mathbb{P}(r_t, o_t, q_{t+1} \mid q_t, a_t, \mathbf{R}) = \theta_{\mathsf{r}}(r_t \mid q_t, a_t)\, \theta_{\mathsf{o}}(o_t \mid q_t, a_t)\, \mathbb{I}(q_{t+1} = \tau(q_t, a_t o_t)).$$

Since RDPs are Markovian in the unobservable states $\mathcal{Q}$, there is an important class of policies that is called *regular*. Given an RDP $\mathbf{R}$, a policy $\pi : \mathcal{H} \to \Delta(\mathcal{A})$ is called *regular* if $\pi(h_1) = \pi(h_2)$ whenever $\bar{\tau}(h_1) = \bar{\tau}(h_2)$, for all $h_1, h_2 \in \mathcal{H}$. Hence, we can compactly define a regular policy as a function of the RDP state, i.e. $\pi : \mathcal{Q} \to \Delta(\mathcal{A})$. Let $\Pi_{\mathbf{R}}$ denote the set of regular policies for $\mathbf{R}$. Regular policies exhibit powerful properties. First, under a regular policy, suffixes have the same probability of being generated for histories that map to the same RDP state. Second, there exists at least one optimal policy that is regular. Finally, in the special case where an RDP is Markovian in both observations and rewards, it reduces to a nonstationary episodic MDP.

Example 2 (RDP for T-maze)**.** Consider the T-maze described in Example 1. This can be modeled as an episodic RDP $\langle \mathcal{Q}, \Sigma, \Omega, \tau, \theta, q_0 \rangle$ with states $\mathcal{Q} \coloneqq q_0 \cup (\{q_{\top 1}, \ldots, q_{\top 13}\} \cup \{q_{\perp 1}, \ldots, q_{\perp 13}\}) \times \{1, \ldots, H\}$, which include the initial state q_0 and two parallel components, $\top$ and $\perp$, for the 13 cells of the grid world. In addition, each state also includes a counter for the time step. Within each component $\{(q_{\top i}, t)\}_i$ and $\{(q_{\perp i}, t)\}$, the transition function τ mimics the grid world dynamics of the maze and increments the counter t. From the initial state and the start action a_0, $\tau(q_0, a_0 o_0)$ equals $q_{1,\top}$ if $o_0 = 110$, and $q_{1,\perp}$ if $o_0 = 011$. Observations are deterministic as described in Example 1, except for $\theta_{\mathsf{o}}(q_0, a_0) = \mathrm{unif}\{110, 011\}$. The rewards are null, except for a 1 in the top right or bottom right cell, depending if the current state is in the component $\top$ or $\perp$, respectively.

Occupancy and distinguishability Given a regular policy $\pi : \mathcal{Q} \to \Delta(\mathcal{A})$ and a time step $t \in [0, H]$, let $d_t^\pi \in \Delta(\mathcal{Q}_t \times \mathcal{A}\,\mathcal{O})$ be the induced *occupancy*, i.e. a probability distribution over the states in $\mathcal{Q}_t$ and the input symbols in $\mathcal{A}\,\mathcal{O}$, recursively defined as $d_0^\pi(q_0, a_0 o_0) = \theta_{\mathsf{o}}(o_0 \mid q_0, a_0)$ and

$$d_t^\pi(q_t, a_t o_t) = \sum_{(q, ao) \in \tau^{-1}(q_t)} d_{t-1}^\pi(q, ao) \cdot \pi(a_t \mid q_t) \cdot \theta_{\mathsf{o}}(o_t \mid q_t, a_t), \quad t > 0.$$

Of particular interest is the occupancy distribution $d_t^* \coloneqq d_t^{\pi^*}$ associated with an optimal policy π^*. Let us assume that π^* is unique, which further implies that d_t^* is uniquely defined[1].

Consider a minimal RDP $\mathbf{R}$ with states $\mathcal{Q} = \cup_{t \in [0, H+1]} \mathcal{Q}_t$. Given a regular policy $\pi \in \Pi_{\mathbf{R}}$ and a time step $t \in [0, H]$, each RDP state $q \in \mathcal{Q}_t$ defines a unique probability distribution $\mathbb{P}(\cdot \mid q_t = q, \pi)$ on episode suffixes in $\mathcal{E}_{H-t} = (\mathcal{A}\mathcal{O}/\mathcal{R})^{H-t+1}$. The states in $\mathcal{Q}_t$ can be compared in terms of the probability distributions they induce over $\mathcal{E}_{H-t}$. Consider any $L = \{L_\ell\}_{\ell=0}^{H}$, where each L_ℓ is a metric over $\Delta(\mathcal{E}_\ell)$. We define the *L-distinguishability* of $\mathbf{R}$ and π as the maximum $\mu_0 \geq 0$ such that, for any $t \in [0, H]$ and any two distinct $q, q' \in \mathcal{Q}_t$, the probability distributions over suffix traces $e_{t:H} \in \mathcal{E}_\ell$ from the two states satisfy

$$L_{H-t}(\mathbb{P}(e_{t:H} \mid q_t = q, \pi), \mathbb{P}(e_{t:H} \mid q_t = q', \pi)) \geq \mu_0\,.$$

[1]This assumption is imposed to ease the definition of the concentrability coefficient that follows, and is also considered in the offline reinforcement learning literature (e.g., [46]). In the general case, one may consider the set $\mathcal{D}^*$ of occupancy distributions, collecting occupancy distributions of all optimal policies.

We will often omit the remaining episode length $\ell = H - t$ from L_ℓ and simply write L. We consider the L^{p}_∞-distinguishability, instantiating the definition above with the metric $L^{\mathsf{p}}_\infty(p_1, p_2) = \max_{u\in[0,\ell], e\in\mathcal{E}_u} |p_1(e*) - p_2(e*)|$, where $p_i(e*)$ represents the probability of the trace prefix $e \in \mathcal{E}_u$, followed by any trace $e' \in \mathcal{E}_{\ell-u-1}$. The L^{p}_1-distinguishability is defined analogously using $L^{\mathsf{p}}_1(p_1, p_2) = \sum_{u\in[0,\ell], e\in\mathcal{E}_u} |p_1(e*) - p_2(e*)|$.

3 Learning RDPs with state-merging algorithms from offline data

Here we describe an algorithm called AdaCT-H [14] for learning episodic RDPs from a dataset of episodes. The algorithm starts with a set $\mathcal{D}$ of episodes generated using a regular behavior policy π^{b}, where the k-th episode is of the form $e^k_{0:H} = a^k_0 o^k_0 / r^k_0 \cdots a^k_H o^k_H / r^k_H$ and where, for each $t \in [0, H]$,

$$q^k_0 = q_0, \quad a^k_t \sim \pi^{\mathsf{b}}(q^k_t), \quad o^k_t \sim \theta_{\mathsf{o}}(q^k_t, a^k_t), \quad r^k_t \sim \theta_{\mathsf{r}}(q^k_t, a^k_t), \quad q^k_{t+1} = \tau(q^k_t, a^k_t o^k_t).$$

Note that the behavior policy π^{b} and underlying RDP states q^k_t are unknown to the learner. The algorithm is an instance of the PAC learning framework and takes as input an accuracy $\varepsilon \in (0, H]$ and a failure probability $\delta \in (0, 1)$. The aim is to find an ε-optimal policy $\widehat{\pi}$ satisfying $V^*_0(h) - V^{\widehat{\pi}}_0(h) \le \varepsilon$ for each $h \in \mathcal{H}_0$ with probability at least $1 - \delta$, using the smallest dataset $\mathcal{D}$ possible.

Since AdaCT-H performs offline learning, it is necessary to control the mismatch in occupancy between the behavior policy π^{b} and the optimal policy π^*. Concretely, the single-policy RDP concentrability coefficient associated with RDP $\mathbf{R}$ and behavior policy π^{b} is defined as [14]:

$$C^*_{\mathbf{R}} = \max_{t\in[H], q\in\mathcal{Q}_t, ao\in\mathcal{AO}} \frac{d^*_t(q, ao)}{d^{\mathsf{b}}_t(q, ao)}\,. \tag{2}$$

As [14], we also assume that the concentrability is bounded away from infinity, i.e. that $C^*_{\mathbf{R}} < \infty$.

AdaCT-H is a state-merging algorithm that iteratively constructs the set of RDP states $\mathcal{Q}_1, \ldots, \mathcal{Q}_H$ and the associated transition function τ. For each $t \in [0, H]$, AdaCT-H maintains a set of candidate states $qao \in \mathcal{Q}_{t-1} \times \mathcal{AO}$. Each candidate state qao has an associated multiset of suffixes $\mathcal{Z}(qao) = \{e^k_{t:H} : e^k \in \mathcal{D}, \bar{\tau}(e^k_{0:t-1}) = q, a^k_{t-1} o^k_{t-1} = ao\}$, i.e. episode suffixes whose history is consistent with qao. To determine whether or not the candidate qao should be promoted to $\mathcal{Q}_t$ or merged with an existing RDP state q_t, AdaCT-H compares the empirical probability distributions on suffixes using the prefix distance L^{p}_∞ defined earlier. For reference, we include the pseudocode of AdaCT-H$(\mathcal{D}, \delta)$ in Appendix A. Cipollone et al. [14] prove that AdaCT-H$(\mathcal{D}, \delta)$ constructs a minimal RDP $\mathbf{R}$ with probability at least $1 - 4AOQ\delta$.

In practice, the main bottleneck of AdaCT-H is the statistical test on the last line,

$$L^{\mathsf{p}}_\infty(\mathcal{Z}_1, \mathcal{Z}_2) \ge \sqrt{2\log(8(ARO)^{H-t}/\delta) / \min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)},$$

since the number of episode suffixes in $\mathcal{E}_\ell$ is exponential in the current horizon ℓ. The purpose of the present paper is to develop tractable methods for implementing the statistical test. These tractable methods can be directly applied to any algorithm that performs such statistical tests, e.g. the approximation algorithm AdaCT-H-A [14].

4 Tractable offline learning of RDPs

The lower bound derived in Cipollone et al. [14] shows that sample complexity of learning RDPs is inversely proportional to the L^{p}_1-distinguishability. When testing candidate states of the unknown RDP, L^{p}_1 is the metric that allows maximum separation between distinct distributions over traces. Unfortunately, accurate estimates of L^{p}_1 are impractical to obtain for distributions over large supports—in our case the number of episode suffixes which is exponential in the horizon. Accurate estimates of L^{p}_∞ are much more practical to obtain. However, there are instances for which states can be well separated in the L^{p}_1-norm, but have an L^{p}_∞-distance that is exponentially small. To address these issues, in this section we develop two improvements over the previous learning algorithms for RDPs.

4.1 Testing in structured languages

The language of traces Under a regular policy, any RDP state $\mathcal{Q}_t$ uniquely determines a distribution over the remaining trace $a_t o_t / r_t \ldots a_H o_H / r_H \in (\mathcal{AO}/\mathcal{R})^{H-t+1}$. Existing work [14, 55, 56] treats

each element $a_i o_i / r_i$ as an independent atomic symbol. This approach neglects the internal structure of each tuple and of the observations, which are often composed of features. As a result, common conditions such as the presence of a reward become unpractical to express. In this work, we allow observations to be composed of internal features. Let $\mathcal{O} = \mathcal{O}^{(1)} \times \cdots \times \mathcal{O}^{(m)}$ be an observation space. We choose to see it as the language $\mathcal{O} = \mathcal{O}^{(1)} \cdots \mathcal{O}^{(m)}$ given by the concatenation of the features. Then, instead of representing an observation $(o^{(1)}, \ldots, o^{(m)})$ as an atomic symbol, we consider it as the word $o^{(1)} o^{(2)} \cdots o^{(m)}$. This results into traces $\mathcal{E}_i = (\mathcal{A}\mathcal{O}^{(1)} \cdots \mathcal{O}^{(m)} / \mathcal{R})^{i+1}$, and each regular policy and RDP state uniquely determine a distribution over strings from $\mathcal{E}_{H-t+1}$. This fine-grained representation greatly simplifies the representation of most common conditions, such as the presence of specific rewards or features in the observation vector.

Testing in the language metric The metrics induced by the L_1 and L_∞ norms are completely generic and can be applied to any distribution. Although this is generally an advantage, this means that they do not exploit the internal structure of the sample space. In our application, a sample is a trace that, as discussed above, can be regarded as a string of a specific language. An important improvement, proposed by Balle [4], is to consider L_1^{p} and L_∞^{p}, which take into account the variable length and conditional probabilities of longer suffixes. This was the approach followed by the previous RDP learning algorithms. However, these two norms are strongly different and lead to dramatically different sample and space complexities.

In this section, we define a new formalism that unifies both metrics and will allow the development of new techniques for distinguishing distributions over traces. Specifically, instead of expressing the probability of single strings, we generalize the concepts above by considering the probability of *sets* of strings. A careful selection of the sets to consider, which are languages, will allow an accurate trade-off between generality and complexity.

Definition 1. Let $\ell \in \mathbb{N}$, let Γ be an alphabet, and let $\mathcal{X}$ be a set of languages consisting of strings in Γ^ℓ. The *language metric* in $\mathcal{X}$ is the function $L_\mathcal{X} : \Delta(\Gamma^\ell) \times \Delta(\Gamma^\ell) \to \mathbb{R}$, on pairs of probability distributions p, p' over Γ^ℓ, defined as

$$L_\mathcal{X}(p, p') \coloneqq \max_{X \in \mathcal{X}} |p(X) - p'(X)|, \tag{3}$$

where the probability of a language is $p(X) \coloneqq \sum_{x \in X} p(x)$.

This original notion unifies all the most common metrics. Considering distributions over Γ^ℓ, when $\mathcal{X} = \{\{x\} \mid x \in \Gamma^\ell\}$, the language metric $L_\mathcal{X}$ reduces to L_∞. When $\mathcal{X} = 2^{\Gamma^\ell}$, which is the set of all languages in Γ^ℓ, the language metric becomes the total variation distance, which is half the value of L_1. A similar reduction can be made for the prefix distances. The language metric captures L_∞^{p} when $\mathcal{X} = \{x\Gamma^{\ell-t} \mid t \in [0, \ell], x \in \Gamma^t\}$, and it captures L_1^{p} when $\mathcal{X} = \cup_{t \in [0,\ell]} 2^{\Gamma^t}$.

Testing in language classes The language metric can be applied directly to the language of traces and used for testing in RDP learning algorithms. In particular, it suffices to consider any set of languages $\mathcal{X}$ that satisfy $X \subseteq \mathcal{E}_{H-h} = (\mathcal{A}\mathcal{O}^{(1)} \cdots \mathcal{O}^{(m)} / \mathcal{R})^{H-t+1} \subseteq \Gamma^{H-t+1}$, for each $X \in \mathcal{X}$. However, as we have seen above, the selection of a specific set of languages $\mathcal{X}$ has a dramatic impact on the metric that is being captured. In this section, we study an appropriate trade-off between generality and sample efficiency, obtained through a suitable selection of $\mathcal{X}$. Intuitively, we seek to evaluate the distance between candidate states based on increasingly complex sets of languages. We present a way to construct a hierarchy of sets of languages of increasing complexity. As a first step, we define sets of basic patterns $\mathcal{G}_i$ of increasing complexity.

$$\mathcal{G}_1 = \left\{a\mathcal{O}/\mathcal{R} \mid a \in \mathcal{A}\right\} \cup \left\{\mathcal{A}\mathcal{O}/r \mid r \in \mathcal{R}\right\} \cup \left\{\mathcal{A}\mathcal{O}^{(1)} \cdots o^{(i)} \cdots \mathcal{O}^{(m)}/\mathcal{R} \mid i \in [m], o^{(i)} \in \mathcal{O}^{(i)}\right\},$$
$$\mathcal{G}_i = \mathcal{G}_{i-1} \cup \mathrm{B}(\mathcal{G}_{i-1}), \quad \forall i \in [2, m+2].$$

In particular, $\mathcal{G}_1$ focuses on single components, by matching an action a, a reward r, or a single observation feature $o^{(i)}$. At the opposite side of the spectrum, the set $\mathcal{G}_{m+2}$ considers every possible combination of actions, observation, reward; namely, it includes the set of singleton languages $\left\{\{ao^{(1)} \cdots o^{(m)}/r\} \mid ao^{(1)} \cdots o^{(m)}/r \in \mathcal{A}\mathcal{O}^{(1)} \cdots \mathcal{O}^{(m)}/\mathcal{R}\right\}$, which is the most fine-grained choice of patterns, but it grows exponentially with m. On the contrary, the cardinality of $\mathcal{G}_1$ is linear in m. Starting from the above hiearchy, we identify two more dimensions along which complexity can

grow. One dimension results from concatenating the basic patterns from $\mathcal{G}_i$, and it is obtained by applying the operator C_k^ℓ. The other dimension results from Boolean combinations of languages, and it is obtained by applying the operator B. Thus, letting $\ell = H - t + 1$, we define the following *three-dimensional hierarchy* of sets $\mathcal{X}_{i,j,k}$ of languages:

$$\mathcal{X}_{i,j,1} = \mathcal{X}_{i,j-1,1} \cup \mathrm{C}_j^\ell(\mathcal{G}_i), \qquad \forall i \in [1, m+2], \forall j \in [\ell],$$
$$\mathcal{X}_{i,j,k} = \mathcal{X}_{i,j,k-1} \cup \mathrm{B}(\mathcal{X}_{i,j,k-1}), \qquad \forall i \in [1, m+2], \forall j \in [\ell],\ \forall k \in [2, \ell].$$

The family $\mathcal{X}_{i,j,k}$ induces a family of language metrics $L_{\mathcal{X}_{i,j,k}}$, which are non-decreasing along the dimensions of the hiearchy:

$$L_{\mathcal{X}_{i,j,k}} \leq L_{\mathcal{X}_{i+1,j,k}}, L_{\mathcal{X}_{i,j+1,k}}, L_{\mathcal{X}_{i,j,k+1}}, \quad \forall i \in [1, m+2], \forall j \in [\ell],\ \forall k \in [\ell],$$

and it may actually be increasing since we include more and more languages as we move towards higher levels of the hierarchy. Moreover, the last levels $\mathcal{X}_{m+2,\ell,k}$ satisfy $L_{\mathcal{X}_{m+2,\ell,k}} \geq L_\infty^{\mathsf{p}}$ for every $k \in [\ell]$ since $\{x\mathcal{E}^{\ell-t} \mid t \in [0,\ell],\ x \in \mathcal{E}^t\} \subseteq \mathcal{X}_{m+2,\ell,k}$. Therefore, the metric $L_{\mathcal{X}_{m+2,\ell,k}}$ is at least as effective as L_∞^{p} in distinguishing candidate states. It can be much more effective as shown next.

Example 3 (Language metric in T-maze)**.** We now discuss the importance of our language metric for distinguishability, with respect to the T-maze described in Example 1 and the associated RDP in Example 2. Consider the two states $(q_{\top 6}, 6)$ and $(q_{\bot 6}, 6)$ in the middle of the corridor and assume that the behavior policy is uniform. In these two states, the probability of each future sequence of actions, observations, and rewards is identical, except for the sequences that reach S or the top/bottom cells. These differ in the starting observation or the reward, respectively. Thus, the maximizer in the definition of L_∞^{p}-distinguishability is the length $u = 6$ and any string x that contains the initial observation. Since the behavior policy is a random walk, the probability of x is 0.5^6 from one state, and 0 from the other, depending on the observation. Therefore, L_∞^{p}-distinguishability decreases exponentially with the length of the corridor. Consider instead the language $\mathcal{X}_{1,1,1}$ and its associated $L_{\mathcal{X}_{1,1,1}}$. This set of languages includes $\Gamma^*\mathcal{A}\text{“}110\text{”}\mathcal{R}\Gamma^* \cap \Gamma^\ell$, which represents the language of any episode suffix containing the observation 110. Since it does not depend on specific paths, its probability is 0 for $(q_{\bot 6}, 6)$ and it equals the probability of visiting the leftmost cell for $(q_{\top 6}, 6)$. For sufficiently long episodes, this probability approaches 1. Thus, in some domains, the language metric improves exponentially over L_∞^{p}.

Assumption 1. The behavior policy π^{b} has an $L_{\mathcal{X}_{i,j,k}}$-distinguishability of at least $\mu_0 > 0$, where $\mathcal{X}_{i,j,k}$ is constructed as above and is an input to the algorithm.

Given $i \in [0, H]$, let $p, p' \in \Delta(\mathcal{E}_i)$ be two distributions over traces. To have accurate estimates for the language metric over some $\mathcal{X}_{i,j,k}$, we instantiate two estimators, $\hat{p}$ and $\hat{p}'$, respectively built using datasets of episodes $\mathcal{C}$ and $\mathcal{C}'$, defined as the fraction of samples that belong to the language; that is, $\hat{p} \coloneqq \sum_{e \in \mathcal{C}} \mathbb{I}(e \in \mathcal{X}_{i,j,k})/|\mathcal{C}|$ and $\hat{p}' \coloneqq \sum_{e \in \mathcal{C}'} \mathbb{I}(e \in \mathcal{X}_{i,j,k})/|\mathcal{C}'|$.

4.2 Analysis

In this section, we show how to derive high-probability sample complexity bounds for ADACT-H when Count-Min-Sketch (CMS) or the language metric $L_\mathcal{X}$ is used. Intuitively, it is sufficient to show that the statistical test is correct with high probability; the remaining proof is identical to that of Cipollone et al. [14].

Theorem 1. ADACT-H$(\mathcal{D}, \delta)$ *returns a minimal RDP* $\mathbf{R}$ *with probability at least* $1 - 4AOQ\delta$ *when CMS is used to store the empirical probability distributions of episode suffixes, the statistical test is*

$$L_\infty^{\mathsf{p}}(\mathcal{Z}_1, \mathcal{Z}_2) \geq \sqrt{8 \log(16(ARO)^{H-t}/\delta) / \min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)},$$

and the size of the dataset is at least $|\mathcal{D}| \geq \widetilde{\mathcal{O}}(\sqrt{H}/d_{\min}^{\mathsf{b}}\mu_0)$*, where* $d_{\min}^{\mathsf{b}} = \min_{t,q,ao} d_t^{\mathsf{b}}(q, ao)$.

The proof of Theorem 1 appears in Appendix C.

Theorem 2. ADACT-H$(\mathcal{D}, \delta)$ *returns a minimal RDP* $\mathbf{R}$ *with probability at least* $1 - 4AOQ\delta$ *when the statistical test is implemented using the language metric* $L_\mathcal{X}$ *and equals*

$$L_\mathcal{X}(\mathcal{Z}_1, \mathcal{Z}_2) \geq \sqrt{2 \log(4|\mathcal{X}|/\delta) / \min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)},$$

and the size of the dataset is at least $|\mathcal{D}| \geq \widetilde{\mathcal{O}}(1/d_{\min}^{\mathsf{b}}\mu_0)$.

Table 1: Summary of the experiments. For each domain, H is the horizon, and for each algorithm, Q is the number of states of the learned automaton, r is the average reward of the derived policy, and time is the running time in seconds of automaton learning.

		FlexFringe			CMS			Restricted Languages		
Name	H	Q	r	time	Q	r	time	Q	r	time
Corridor	5	11	1.0	0.03	11	1.0	0.3	11	1.0	0.01
T-maze	5	29	0.0	0.11	104	4.0	10.1	18	4.0	0.26
Cookie	9	220	1.0	0.36	116	1.0	6.05	91	1.0	0.08
Cheese	6	669	$0.69 \pm .04$	19.28	1158	$0.4 \pm .05$	207.4	326	$0.81 \pm .04$	2.23
Mini-hall	15	897	$0.33 \pm .04$	25.79	-	-	-	5134	$0.91 \pm .03$	23.9

The proof of Theorem 2 also appears in Appendix C. Note that by definition, μ_0 is the L-distinguishability of $\mathbf{R}$ for the chosen language set $\mathcal{X}$, which has to satisfy $\mu_0 > 0$ for ADACT-H to successfully learn a minimal RDP. Even though our sample complexity results are similar to those of previous work up to a factor $\sqrt{H}$, as discussed earlier, μ_0 may be exponentially smaller for $L_{\mathcal{X}}$ than for L_∞^{p}. We remark that the analysis does not hold if CMS is used to store the empirical probability distribution of the language metric $L_{\mathcal{X}}$, since the languages in a set $\mathcal{X}$ may overlap, causing the probabilities to sum to a value significantly greater than 1.

5 Experimental Results

In this section we present the results of experiments with two versions of ADACT-H: one that uses CMS to compactly store probability distributions on suffixes, and one that uses the restricted language family $\mathcal{X}_{1,1,1}$. We compare against FlexFringe [7], a state-of-the-art algorithm for learning probabilistic deterministic finite automata, which includes RDPs as a special case. To approximate episodic traces, we add a termination symbol to the end of each trace, but FlexFringe sometimes learns RDPs with cycles. Moreover, FlexFringe uses a number of different heuristics that optimize performance, but these heuristics no longer preserve the high-probability guarantees. Hence the automata output by FlexFringe are not always directly comparable to the RDPs output by ADACT-H.

We perform experiments in five domains from the literature on POMDPs and RDPs: Corridor [55], T-maze [3], Cookie [64], Cheese [42] and Mini-hall [36]. Appendix D contains a detailed description of each domain, as well as example automata learned by the different algorithms. Table 1 summarizes the results of the three algorithms in the five domains. We see that ADACT-H with CMS is significantly slower than FlexFringe, which makes sense since FlexFringe has been optimized for performance. CMS compactly represents the probability distributions over suffixes, but ADACT-H still has to iterate over all suffixes, which is exponential in H for the L_∞^{p} distance. As a result, CMS suffers from slow running times, and exceeds the alloted time budget of 1800 seconds in the Mini-hall domain.

On the other hand, ADACT-H with the restricted language family $\mathcal{X}_{1,1,1}$ is faster than FlexFringe in all domains except T-maze, and outputs smaller automata than both FlexFringe and CMS in all domains except Mini-hall. The number of languages of $\mathcal{X}_{1,1,1}$ is linear in the number of actions, observation symbols, and reward values, so the algorithm does not have to iterate over all suffixes, and the resulting RDPs better exploit the underlying structure of the domains. In Corridor and Cookie, all algorithms learn automata that admit an optimal policy. However, in T-maze, Cheese and Mini-hall, the RDPs learned by ADACT-H with the restricted language family $\mathcal{X}_{1,1,1}$ admit a policy that outperforms those of FlexFringe and CMS. As mentioned, the heuristics used by FlexFringe are not optimized to preserve reward, which is likely the reason why the derived policies perform worse.

6 Conclusion

In this paper, we propose two new approaches to offline RL for Regular Decision Processes and provide their respective theoretical analysis. We also improve upon existing algorithms for RDP learning, and propose a modified algorithm using Count-Min-Sketch with reduced memory complexity. We define a hierarchy of language families and introduce a language-restricted approach, removing the dependency on L_∞^{p}-distinguishability parameters and compare the performance of our algorithms

to FlexFringe, a state-of-the-art algorithm for learning probabilistic deterministic finite automata. Although CMS suffers from a large running time, the language-restricted approach offers smaller automata and optimal (or near optimal) policies, even on domains requiring long-term dependencies. Finally, as a future work, we plan to expand our approach to the online RDP learning setting.

Acknowledgments and Disclosure of Funding

Anders Jonsson is partially supported by AGAUR SGR and Spanish grant PID2019-108141GB-I00. Alessandro Ronca is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 852769, ARiAT). Mohammad Sadegh Talebi acknowledges partial support by the Independent Research Fund Denmark, grant number 1026-00397B.

References

[1] E. Abadi and R. I. Brafman. Learning and solving Regular Decision Processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1948–1954, 2020.

[2] K. Azizzadenesheli, A. Lazaric, and A. Anandkumar. Reinforcement learning of POMDPs using spectral methods. In *Conference on Learning Theory (COLT)*, pages 193–256, 2016.

[3] B. Bakker. Reinforcement learning with long short-term memory. In *Neural Information Processing Systems (NeurIPS)*, pages 1475–1482, 2001.

[4] B. Balle. *Learning Finite-State Machines: Statistical and Algorithmic Aspects*. PhD thesis, Universitat Politècnica de Catalunya, 2013.

[5] B. Balle, J. Castro, and R. Gavaldà. Learning probabilistic automata: A study in state distinguishability. *Theor. Comput. Sci.*, 473:46–60, 2013.

[6] B. Balle, J. Castro, and R. Gavaldà. Adaptively learning probabilistic deterministic automata from data streams. *Machine Learning*, 96(1):99–127, 2014.

[7] R. Baumgartner and S. Verwer. Learning state machines from data streams: A generic strategy and an improved heuristic. In *International Conference on Graphics and Interaction (ICGI)*, pages 117–141, 2023.

[8] H. Bourel, A. Jonsson, O.-A. Maillard, and M. S. Talebi. Exploration in reward machines with low regret. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4114–4146, 2023.

[9] M. H. Bowling, P. McCracken, M. James, J. Neufeld, and D. F. Wilkinson. Learning predictive state representations using non-blind policies. In *International Conference on Machine Learning (ICML)*, pages 129–136, 2006.

[10] A. K. Bozkurt, Y. Wang, M. M. Zavlanos, and M. Pajic. Control synthesis from linear temporal logic specifications using model-free reinforcement learning. In *International Conference on Robotics and Automation (ICRA)*, pages 10349–10355, 2020.

[11] R. I. Brafman and G. De Giacomo. Regular Decision Processes: A Model for Non-Markovian Domains. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5516–5522, 2019.

[12] R. I. Brafman and G. De Giacomo. Regular decision processes. *Artificial Intelligence*, 331: 104113, 2024.

[13] J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 1042–1051, 2019.

[14] R. Cipollone, A. Jonsson, A. Ronca, and M. S. Talebi. Provably efficient offline reinforcement learning in regular decision processes. In *Neural Information Processing Systems (NeurIPS)*, 2023.

[15] A. Clark and F. Thollard. PAC-learnability of probabilistic deterministic finite state automata. *J. Mach. Learn. Res.*, 5:473–497, 2004.

[16] G. Cormode and S. Muthukrishnan. An improved data stream summary: the Count-Min Sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.

[17] G. De Giacomo, L. Iocchi, M. Favorito, and F. Patrizi. Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 128–136, 2019.

[18] Y. Efroni, C. Jin, A. Krishnamurthy, and S. Miryoosefi. Provable reinforcement learning with a short-term memory. In *International Conference on Machine Learning (ICML)*, pages 5832–5850, 2022.

[19] J. Fu and U. Topcu. Probably approximately correct MDP learning and control with temporal logic constraints. In *Robotics: Science and Systems (RSS)*, 2014.

[20] G. D. Giacomo, M. Favorito, L. Iocchi, F. Patrizi, and A. Ronca. Temporal logic monitoring rewards via transducers. In *Knowledge Representation and Reasoning (KR)*, pages 860–870, 2020.

[21] H. Guo, Q. Cai, Y. Zhang, Z. Yang, and Z. Wang. Provably efficient offline reinforcement learning for partially observable Markov decision processes. In *International Conference on Machine Learning (ICML)*, pages 8016–8038, 2022.

[22] Z. D. Guo, S. Doroudi, and E. Brunskill. A PAC RL algorithm for episodic POMDPs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 510–518, 2016.

[23] E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Omega-regular objectives in model-free reinforcement learning. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 395–412, 2019.

[24] L. Hammond, A. Abate, J. Gutierrez, and M. J. Wooldridge. Multi-agent reinforcement learning with temporal logic specifications. In *Autonomous Agents and Multiagent Systems (AAMAS)*, pages 583–592, 2021.

[25] M. Hasanbeig, A. Abate, and D. Kroening. Cautious reinforcement learning with logical constraints. In *Autonomous Agents and Multiagent Systems (AAMAS)*, pages 483–491, 2020.

[26] M. Hasanbeig, N. Y. Jeppu, A. Abate, T. Melham, and D. Kroening. DeepSynth: automata synthesis for automatic task segmentation in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 7647–7656, 2021.

[27] M. Hutter. Feature reinforcement learning: Part I. Unstructured MDPs. *J. Artif. Gen. Intell.*, 1 (1):3–24, 2009.

[28] M. R. James and S. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In *International Conference on Machine Learning (ICML)*, 2004.

[29] C. Jin, S. M. Kakade, A. Krishnamurthy, and Q. Liu. Sample-efficient reinforcement learning of undercomplete POMDPs. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[30] Y. Jin, Z. Yang, and Z. Wang. Is pessimism provably efficient for offline RL? In *International Conference on Machine Learning (ICML)*, pages 5084–5096, 2021.

[31] N. Kallus and M. Uehara. Double reinforcement learning for efficient off-policy evaluation in Markov decision processes. *Journal of Machine Learning Research*, 21(1):6742–6804, 2020.

[32] A. Krishnamurthy, A. Agarwal, and J. Langford. PAC reinforcement learning with rich observations. In *Neural Information Processing Systems (NeurIPS)*, pages 1840–1848, 2016.

[33] A. Kulesza, N. Jiang, and S. Singh. Spectral learning of predictive state representations with insufficient statistics. In *AAAI Conference on Artificial Intelligence*, pages 2715–2721, 2015.

[34] T. Lattimore, M. Hutter, and P. Sunehag. The sample-complexity of general reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2013.

[35] G. Li, L. Shi, Y. Chen, Y. Chi, and Y. Wei. Settling the sample complexity of model-based offline reinforcement learning. *arXiv preprint arXiv:2204.05275*, 2022.

[36] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning (ICML)*, pages 362–370, 1995.

[37] Q. Liu, A. Chung, C. Szepesvári, and C. Jin. When is partially observable reinforcement learning not scary? In *Conference on Learning Theory (COLT)*, pages 5175–5220, 2022.

[38] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton. Toward off-policy learning control with function approximation. In *International Conference on Machine Learning (ICML)*, pages 719–726, 2010.

[39] M. M. H. Mahmud. Constructing states for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 727–734, 2010.

[40] O. Maillard, R. Munos, and D. Ryabko. Selecting the state-representation in reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, pages 2627–2635, 2011.

[41] O. Maillard, P. Nguyen, R. Ortner, and D. Ryabko. Optimal regret bounds for selecting the state representation in reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 543–551, 2013.

[42] A. McCallum and D. H. Ballard. Reinforcement learning with selective perception and hidden state. 1996. URL `https://api.semanticscholar.org/CorpusID:60716402`.

[43] E. F. Moore. Gedanken-experiments on sequential machines. *Automata studies*, 34:129–153, 1956.

[44] R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge university press, 1995.

[45] P. Nguyen, O. Maillard, D. Ryabko, and R. Ortner. Competing with an infinite set of models in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 463–471, 2013.

[46] T. Nguyen-Tang, M. Yin, S. Gupta, S. Venkatesh, and R. Arora. On instance-dependent bounds for offline reinforcement learning with linear function approximation. In *AAAI Conference on Artificial Intelligence*, pages 9310–9318, 2023.

[47] R. Ortner, M. Pirotta, A. Lazaric, R. Fruit, and O. Maillard. Regret bounds for learning state representations in reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, pages 12717–12727, 2019.

[48] N. Palmer and P. W. Goldberg. PAC-learnability of probabilistic deterministic finite state automata in terms of variation distance. *Theor. Comput. Sci.*, 387(1):18–31, 2007.

[49] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[50] J.-É. Pin. The dot-depth hierarchy, 45 years later. *THE ROLE OF THEORY IN COMPUTER SCIENCE: Essays Dedicated to Janusz Brzozowski*, pages 177–201, 2017.

[51] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994.

[52] P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In *Neural Information Processing Systems (NeurIPS)*, pages 11702–11716, 2021.

[53] T. Ren, J. Li, B. Dai, S. S. Du, and S. Sanghavi. Nearly horizon-free offline reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, pages 15621–15634, 2021.

[54] D. Ron, Y. Singer, and N. Tishby. On the learnability and usage of acyclic probabilistic finite automata. *J. Comput. Syst. Sci.*, 56(2):133–152, 1998.

[55] A. Ronca and G. De Giacomo. Efficient PAC reinforcement learning in regular decision processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2026–2032, 2021.

[56] A. Ronca, G. P. Licks, and G. De Giacomo. Markov abstractions for PAC reinforcement learning in non-markov decision processes. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3408–3415, 2022.

[57] J. Shallit. *A second course in formal languages and automata theory*. Cambridge University Press, 2008.

[58] I. Simon. *Hierarchies of events with dot-depth one*. PhD thesis, University of Waterloo, 1972.

[59] S. Singh, M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone. Learning predictive state representations. In *International Conference on Machine Learning (ICML)*, pages 712–719, 2003.

[60] J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *Journal of Machine Learning Research*, 23(1):483–565, 2022.

[61] D. Thérien. Imre simon: an exceptional graduate student. *RAIRO Theor. Informatics Appl.*, 39 (1):297–304, 2005.

[62] P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 2139–2148, 2016.

[63] R. Toro Icarte, T. Q. Klassen, R. A. Valenzano, and S. A. McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 2112–2121, 2018.

[64] R. Toro Icarte, E. Waldie, T. Q. Klassen, R. A. Valenzano, M. P. Castro, and S. A. McIlraith. Learning reward machines for partially observable reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, pages 15497–15508, 2019.

[65] M. Uehara and W. Sun. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations (ICLR)*, 2022.

[66] M. Uehara, C. Shi, and N. Kallus. A review of off-policy evaluation in reinforcement learning. *arXiv preprint arXiv:2212.06355*, 2022.

[67] M. Uehara, X. Zhang, and W. Sun. Representation learning for online and offline RL in low-rank MDPs. In *International Conference on Learning Representations (ICLR)*, 2022.

[68] J. Veness, K. S. Ng, M. Hutter, W. T. B. Uther, and D. Silver. A Monte-Carlo AIXI approximation. *J. Artif. Intell. Res.*, 40:95–142, 2011.

[69] T. Xie, N. Jiang, H. Wang, C. Xiong, and Y. Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. In *Neural Information Processing Systems (NeurIPS)*, pages 27395–27407, 2021.

[70] Z. Xu, I. Gavran, Y. Ahmad, R. Majumdar, D. Neider, U. Topcu, and B. Wu. Joint inference of reward machines and policies for reinforcement learning. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 590–598, 2020.

[71] M. Yin and Y.-X. Wang. Towards instance-optimal offline reinforcement learning with pessimism. In *Neural Information Processing Systems (NeurIPS)*, pages 4065–4078, 2021.

[72] W. Zhan, B. Huang, A. Huang, N. Jiang, and J. Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory (COLT)*, pages 2730–2775, 2022.

[73] W. Zhan, M. Uehara, W. Sun, and J. D. Lee. PAC reinforcement learning for predictive state representations. In *International Conference on Learning Representations (ICLR)*, 2023.

A Pseudocode of ADACT-H

FUNCTION(ADACT–H)
Input: Dataset $\mathcal{D}$ containing N traces in $\mathcal{E}_H$, failure probability $0 < \delta < 1$
Output: Set $\mathcal{Q}$ of RDP states, transition function $\tau : \mathcal{Q} \times \mathcal{AO} \to \mathcal{Q}$

$\mathcal{Q}_0 \leftarrow \{q_0\}$, $\mathcal{Z}(q_0) \leftarrow \mathcal{D}$ // initial state
for $t = 0, \ldots, H$ **do**
  $\mathcal{Q}_{c,t+1} \leftarrow \{qao \mid q \in \mathcal{Q}_t, ao \in \mathcal{AO}\}$ // get candidate states
  foreach $qao \in \mathcal{Q}_{c,t+1}$ **do** $\mathcal{Z}(qao) \leftarrow \{e_{t+1:H} \mid aroe_{t+1:H} \in \mathcal{Z}(q)\}$ // compute suffixes
  $q_m a_m o_m \leftarrow \arg\max_{qao \in \mathcal{Q}_{c,t+1}} |\mathcal{Z}(qao)|$ // most common candidate
  $\mathcal{Q}_{t+1} \leftarrow \{q_m a_m o_m\}, \tau(q_m, a_m o_m) = q_m a_m o_m$ // promote candidate
  $\mathcal{Q}_{c,t+1} \leftarrow \mathcal{Q}_{c,t+1} \setminus \{q_m a_m o_m\}$ // remove from candidate states
  for $qao \in \mathcal{Q}_{c,t+1}$ **do**
    Similar $\leftarrow \{q' \in \mathcal{Q}_{t+1} \mid$ not TESTDISTINCT$(t, \mathcal{Z}(qao), \mathcal{Z}(q'), \delta)\}$ // confidence test
    if *Similar* $= \emptyset$ **then** $\mathcal{Q}_{t+1} \leftarrow \mathcal{Q}_{t+1} \cup \{qao\}, \tau(q, ao) = qao$ // promote candidate
    else $q' \leftarrow$ element in *Similar*, $\tau(q, ao) = q'$, $\mathcal{Z}(q') \leftarrow \mathcal{Z}(q') \cup \mathcal{Z}(qao)$ // merge states
  end
end
return $\mathcal{Q}_0 \cup \cdots \cup \mathcal{Q}_{H+1}$, τ

FUNCTION(TESTDISTINCT)
Input: Time t, two multisets $\mathcal{Z}_1$ and $\mathcal{Z}_2$ of traces in $(\mathcal{AO} \times \mathcal{R})^{H-t+1}$, failure probability $0 < \delta < 1$
Output: True if $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are regarded as distinct, False otherwise

return $L_\infty^{\mathrm{p}}(\mathcal{Z}_1, \mathcal{Z}_2) \geq \sqrt{2\log(8(ARO)^{H-t}/\delta)/\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}$

B Pseudometrics

We provide the definition of pseudometric spaces.

Definition 2 (Pseudometrics)**.** Given a set $\mathcal{K}$ and a non-negative function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, we call $(\mathcal{X}, d)$ a pseudometric space with pseudometric d, if for all $x, y, z \in \mathcal{X}$:

(i) $d(x, x) = 0$,

(ii) $d(x, y) = d(y, x)$,

(iii) $d(x, y) + d(y, z) \geq d(x, z)$.

If it also holds that $d(x, y) = 0$ iff $x = y$, then d is a metric.

C Proof of theorems

In this appendix we prove Theorems 1 and 2 using a series of lemmas. We first describe how to implement ADACT-H using CMS.

Given a finite set $\mathcal{Z}$ and a probability distribution $q \in \Delta(\mathcal{Z})$, let $\widehat{q} \in \Delta(\mathcal{Z})$ be an empirical estimate of q computed using n samples. CMS can store an estimate $\widetilde{q}$ of the empirical distribution $\widehat{q}$. In this setting, the vector v contains the empirical counts of each element of $\mathcal{Z}$, which implies $\|v\|_1 = n$ and $\widehat{q}(z_i) = v_i/n$ for each $z_i \in \mathcal{Z}$. The following lemma shows how to bound the error between $\widetilde{q}$ and $\widehat{q}$.

Lemma 3. *Given a finite set $\mathcal{Z}$, a probability distribution $q \in \Delta(\mathcal{Z})$, and an empirical estimate $\widehat{q} \in \Delta(\mathcal{Z})$ of q obtained using n samples, let $\widetilde{q}$ be the estimate of $\widehat{q}$ output by CMS with parameters δ_c and ε. With probability at least $1 - |\mathcal{Z}|\delta_c$ it holds that $\|\widetilde{q} - \widehat{q}\|_\infty \leq \epsilon$.*

Proof. Cormode and Muthukrishnan [16] show that for a point query that returns an approximation $\widetilde{v_i}$ of v_i, with probability at least $1 - \delta_c$ it holds that

$$\widetilde{v_i} \leq v_i + \varepsilon\|v\|_1.$$

In our case, the estimated probability of an element $z_i \in \mathcal{Z}$ equals $\widetilde{q}(z_i) = \widetilde{v}_i/n$, where $\widetilde{v}_i$ is the point query for z_i. Using the result above, with probability at least $1 - \delta_c$ we have

$$\widetilde{q}(z_i) = \frac{\widetilde{v}_i}{n} \leq \frac{v_i}{n} + \frac{\varepsilon \|v\|_1}{n} = \widehat{q}(z_i) + \varepsilon.$$

Since CMS never underestimates a value, $\widehat{q}(z_i) \leq \widetilde{q}(z_i)$ trivially holds. Taking a union bound shows that the inequality above holds simultaneously for all $z_i \in \mathcal{Z}$ with probability $1 - |\mathcal{Z}|\delta_c$. □

The following two lemmas are analogous to Lemmas 13 and 14 of Cipollone et al. [14].

Lemma 4. *For $t \in [0, H]$, let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be multisets sampled from distributions p_1 and p_2 on $\Delta(\mathcal{E}_{H-t})$. Assume that we use CMS with parameters $\delta_c = \delta/8(AOR)^{H-t}$ and $\varepsilon = \sqrt{\log(2/\delta_c)/2|\mathcal{Z}_i|}$ to store an approximation $\widetilde{p}_i$ of the empirical estimate $\widehat{p}_i$ of p_i due to $\mathcal{Z}_i$, $i \in [2]$. If $p_1 = p_2$, with probability at least $1 - \delta$ the statistical test satisfies*

$$L_\infty^{\mathsf{p}}(\widetilde{p}_1, \widetilde{p}_2) \leq \sqrt{\frac{8 \log(16(AOR)^{H-t}/\delta)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}.$$

Proof. Hoeffding's inequality states that for each $i \in [2]$, with probability at least $1 - \delta_c$ it holds for each $e \in \mathcal{E}_u$, $u \leq H - t$, that

$$|\widehat{p}_i(e) - p_i(e)| \leq \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_i|}}.$$

We can now use Lemma 3 to upper bound the L_∞^{p} metric as

$$\begin{aligned}
L_{\mathcal{X}}(\widetilde{p}_1, \widetilde{p}_2) &= \max_{u \in [0, H-t], e \in \mathcal{E}_u} |\widetilde{p}_1(e) - \widetilde{p}_2(e)| \\
&\leq \max_{u \in [0, H-t], e \in \mathcal{E}_u} (|\widetilde{p}_1(e) - \widehat{p}_1(e)| + |\widehat{p}_1(e) - p_1(e)| + |p_1(e) - p_2(e)| \\
&\qquad + |p_2(e) - \widehat{p}_2(e)| + |\widehat{p}_2(e) - \widetilde{p}_2(e)|) \\
&\leq \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_1|}} + \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_1|}} + \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_2|}} + \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_2|}} \leq \sqrt{\frac{8 \log(2/\delta_c)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}.
\end{aligned}$$

Taking a union bound implies that this holds with probability at least $1 - 8(AOR)^{H-t}\delta_c = 1 - \delta$. □

Lemma 5. *For $t \in [0, H]$, let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be multisets sampled from distributions p_1 and p_2 on $\Delta(\mathcal{E}_{H-t})$. Assume that we use CMS with parameters $\delta_c = \delta/8(AOR)^{H-t}$ and $\varepsilon = \sqrt{\log(2/\delta_c)/2|\mathcal{Z}_i|}$ to store an approximation $\widetilde{p}_i$ of the empirical estimate $\widehat{p}_i$ of p_i due to $\mathcal{Z}_i$, $i \in [2]$. If $p_1 \neq p_2$ and $\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|) \geq 32 \log(2/\delta_c)/\mu_0^2$, with probability at least $1 - \delta$ the statistical test satisfies*

$$L_\infty^{\mathsf{p}}(\widetilde{p}_1, \widetilde{p}_2) \geq \sqrt{\frac{8 \log(16(AOR)^{H-t}/\delta)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}.$$

Proof. We can use Lemma 3 to lower bound the L_∞^{p} metric as

$$\begin{aligned}
L_{\mathcal{X}}(\widetilde{p}_1, \widetilde{p}_2) &= \max_{u \in [0, H-t], e \in \mathcal{E}_u} |\widetilde{p}_1(e) - \widetilde{p}_2(e)| \\
&\geq \max_{u \in [0, H-t], e \in \mathcal{E}_u} (|p_1(e) - p_2(e)| - |\widetilde{p}_1(e) - \widehat{p}_1(e)| - |\widehat{p}_1(e) - p_1(e)| \\
&\qquad - |p_2(e) - \widehat{p}_2(e)| - |\widehat{p}_2(e) - \widetilde{p}_2(e)|) \\
&\geq \mu_0 - \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_1|}} - \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_1|}} - \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_2|}} - \sqrt{\frac{\log(2/\delta_c)}{2|\mathcal{Z}_2|}} \\
&\geq \mu_0 - \sqrt{\frac{8 \log(2/\delta_c)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}} \geq \mu_0 - \frac{\mu_0}{2} \geq \frac{\mu_0}{2} \geq \sqrt{\frac{8 \log(2/\delta_c)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}.
\end{aligned}$$

Taking a union bound implies that this holds with probability at least $1 - 8(AOR)^{H-t}\delta_c = 1 - \delta$. □

The remainder of the proof of Theorem 1 follows exactly the same steps as in the proof of Cipollone et al. [14, Theorem 6]. For completeness, we repeat the steps here. The proof consists in choosing $N = |\mathcal{D}|$ and δ such that the condition on $\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)$ in Lemma 5 is true with high probability for each application of TESTDISTINCT. Consider an iteration $t \in [0, H]$ of ADACT-H. For a candidate state $qao \in \mathcal{Q}_{\mathsf{c},t+1}$, its associated probability is $d_t^{\mathsf{b}}(q, ao)$ with empirical estimate $\widehat{p}_t(qao) = |\mathcal{Z}(qao)|/N$, i.e. the proportion of episodes in $\mathcal{D}$ that are consistent with qao. We can apply an empirical Bernstein inequality to show that

$$\mathbb{P}\left(\left|\widehat{p}_t(qao) - d_t^{\mathsf{b}}(q, ao)\right| \geq \sqrt{\frac{2\widehat{p}_t(qao)\ell}{N}} + \frac{14\ell}{3N} = \frac{\sqrt{2M\ell} + 14\ell/3}{N}\right) \leq \delta,$$

where $M = |\mathcal{Z}(qao)|$, $\ell = \log(4/\delta)$, and δ is the failure probability of ADACT-H. To obtain a bound on M and N, assume that we can estimate $d_t^{\mathsf{b}}(q, ao)$ with accuracy $d_t^{\mathsf{b}}(q, ao)/2$, which yields

$$\frac{d_t^{\mathsf{b}}(q, ao)}{2} \geq \frac{\sqrt{2M\ell} + 14\ell/3}{N} \tag{4}$$

$$\widehat{p}_t(qao) \geq d_t^{\mathsf{b}}(q, ao) - \frac{\sqrt{2M\ell} + 14\ell/3}{N} \geq d_t^{\mathsf{b}}(q, ao) - \frac{d_t^{\mathsf{b}}(q, ao)}{2} = \frac{d_t^{\mathsf{b}}(q, ao)}{2}. \tag{5}$$

Combining these two results, we obtain

$$M = N\widehat{p}_t(qao) \geq N d_t^{\mathsf{b}}(q, ao)/2 \geq \frac{N}{2N}\left(\sqrt{2M\ell} + 14\ell/3\right) = \frac{1}{2}\left(\sqrt{2M\ell} + 14\ell/3\right). \tag{6}$$

Solving for M yields $M \geq 4\ell$, which is subsumed by the bound on M in Lemma 5 since $\mu_0 < 1$. Hence the bound on M in Lemma 5 is sufficient to ensure that we estimate $d_t^{\mathsf{b}}(q, ao)$ with accuracy $d_t^{\mathsf{b}}(q, ao)/2$. We can now insert the bound on M from Lemma 5 into (4) to obtain a bound on N:

$$N \geq \frac{2(\sqrt{2M\ell} + 14\ell/3)}{d_t^{\mathsf{b}}(q, ao)} \geq \frac{2\ell}{d_t^{\mathsf{b}}(q, ao)}\left(\frac{8}{\mu_0}\sqrt{\frac{(H-t)\log(4ARO)}{\ell} + 1} + \frac{14}{3}\right) \equiv N_1. \tag{7}$$

To simplify the bound, we can choose any value larger than N_1:

$$\begin{aligned} N_1 &\leq \frac{2\ell}{d_t^{\mathsf{b}}(q, ao)}\left(\frac{8}{\mu_0}\sqrt{H\log(4ARO) + H\log(4ARO)} + \frac{14}{3\mu_0}\sqrt{H\log(4ARO)}\right) \\ &< \frac{32\ell}{d_{\min}^{\mathsf{b}}\,\mu_0}\sqrt{H\log(4ARO)} \equiv N_0, \end{aligned} \tag{8}$$

where we have used $d_t^{\mathsf{b}}(q, ao) \geq d_{\min}^{\mathsf{b}}$, $\mu_0 < 1$, $\ell = \log 4 + \log(1/\delta) \geq 1$, $H\log(4ARO) \geq \log 4 \geq 1$ and $8\sqrt{2} + 14/3 < \frac{32}{2}$. Choosing $\delta = \delta_0/2QAO$, a union bound implies that accurately estimating $d_t^{\mathsf{b}}(q, ao)$ for each candidate state qao and accurately estimating $p(e_{0:u}*)$ for each prefix in the multiset $\mathcal{Z}(qao)$ associated with qao occurs with probability $1 - 2QAO\delta = 1 - \delta_0$, since there are at most QAO candidate states. Ignoring logarithmic terms, this equals the bound in the theorem.

It remains to show that the resulting RDP is minimal. We show the result by induction. The base case is given by the set $\mathcal{Q}_0$, which is clearly minimal since it only contains the initial state q_0. For $t \in [0, H]$, assume that the algorithm has learned a minimal RDP for sets $\mathcal{Q}_0, \ldots, \mathcal{Q}_t$. Let $\mathcal{Q}_{t+1}$ be the set of states at layer $t+1$ of a minimal RDP. Each pair of histories that map to a state $q_{t+1} \in \mathcal{Q}_{t+1}$ generate the same probability distribution over suffixes. Hence by Lemma 4, with high probability TESTDISTINCT$(t, \mathcal{Z}(qao), \mathcal{Z}(q'a'o'), \delta)$ returns false for each pair of candidate states qao and $q'a'o'$ that map to q_{t+1}. Consequently, the algorithm merges qao and $q'a'o'$. On the other hand, by assumption, each pair of histories that map to different states of $\mathcal{Q}_{t+1}$ have L_∞^{p}-distinguishability μ_0. Hence by Lemma 5, with high probability TESTDISTINCT$(t, \mathcal{Z}(qao), \mathcal{Z}(q'a'o'), \delta)$ returns true for each pair of candidate states qao and $q'a'o'$ that map to different states in $\mathcal{Q}_{t+1}$. Consequently, the algorithm does not merge qao and $q'a'o'$. It follows that with high probability, ADACT–H will generate exactly the set $\mathcal{Q}_{t+1}$, which is that of a minimal RDP.

The proof of Theorem 2 is achieved by proving two very similar lemmas.

Lemma 6. *For $t \in [0, H]$, let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be multisets sampled from distributions p_1 and p_2 on $\Delta(\mathcal{E}_{H-t})$, and let $\widehat{p}_1$ and $\widehat{p}_2$ be empirical estimates of p_1 and p_2 due to $\mathcal{Z}_1$ and $\mathcal{Z}_2$, respectively. If $p_1 = p_2$, with probability at least $1 - \delta$ the statistical test satisfies*

$$L_{\mathcal{X}}(\widehat{p}_1, \widehat{p}_2) \leq \sqrt{\frac{2\log(4|\mathcal{X}|/\delta)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}.$$

Proof. Let $\delta_\ell = \delta/2|\mathcal{X}|$. Hoeffding's inequality states that for each $X \in \mathcal{X}$ and each $i \in \{1, 2\}$, with probability at least $1 - \delta_\ell$ it holds that

$$|\widehat{p}_i(X) - p_i(X)| \leq \sqrt{\frac{\log(2/\delta_\ell)}{2|\mathcal{Z}_i|}}.$$

We can now upper bound the language metric as

$$\begin{aligned} L_{\mathcal{X}}(\widehat{p}_1, \widehat{p}_2) &= \max_{X \in \mathcal{X}} |\widehat{p}_1(X) - \widehat{p}_2(X)| \\ &\leq \max_{X \in \mathcal{X}} (|\widehat{p}_1(X) - p_1(X)| + |p_1(X) - p_2(X)| + |p_2(X) - \widehat{p}_2(X)|) \\ &\leq \sqrt{\frac{\log(2/\delta_\ell)}{2|\mathcal{Z}_1|}} + 0 + \sqrt{\frac{\log(2/\delta_\ell)}{2|\mathcal{Z}_2|}} \leq \sqrt{\frac{2\log(2/\delta_\ell)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}. \end{aligned}$$

Taking a union bound implies that this bound holds with probability at least $1 - 2|\mathcal{X}|\delta_\ell = 1 - \delta$. □

Lemma 7. *For $t \in [0, H]$, let $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be multisets sampled from distributions p_1 and p_2 on $\Delta(\mathcal{E}_{H-t})$, and let $\widehat{p}_1$ and $\widehat{p}_2$ be empirical estimates of p_1 and p_2 due to $\mathcal{Z}_1$ and $\mathcal{Z}_2$, respectively. If $p_1 \neq p_2$ and $\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|) \geq 8\log(4|\mathcal{X}|/\delta)/\mu_0^2$, with probability at least $1 - \delta$ the statistical test satisfies*

$$L_{\mathcal{X}}(\widehat{p}_1, \widehat{p}_2) \geq \sqrt{\frac{2\log(4|\mathcal{X}|/\delta)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}.$$

Proof. We can lower bound the language metric as

$$\begin{aligned} L_{\mathcal{X}}(\widehat{p}_1, \widehat{p}_2) &= \max_{X \in \mathcal{X}} |\widehat{p}_1(X) - \widehat{p}_2(X)| \\ &\geq \max_{X \in \mathcal{X}} (|p_1(X) - p_2(X)| - |\widehat{p}_1(X) - p_1(X)| - |p_2(X) - \widehat{p}_2(X)|) \\ &\geq \mu_0 - \sqrt{\frac{\log(2/\delta_\ell)}{2|\mathcal{Z}_1|}} - \sqrt{\frac{\log(2/\delta_\ell)}{2|\mathcal{Z}_2|}} \geq \mu_0 - \sqrt{\frac{2\log(4|\mathcal{X}|/\delta)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}} \\ &\geq \mu_0 - \frac{\mu_2}{2} \geq \frac{\mu_0}{2} \geq \sqrt{\frac{2\log(4|\mathcal{X}|/\delta)}{\min(|\mathcal{Z}_1|, |\mathcal{Z}_2|)}}. \end{aligned}$$

Taking a union bound implies that this bound holds with probability at least $1 - 2|\mathcal{X}|\delta_\ell = 1 - \delta$. □

The remainder of the proof of Theorem 2 is analogous to that of Theorem 1. However, we no longer get a term $\sqrt{H}$ from $\sqrt{\log((AOR)^H)}$ unless the number of languages in $\mathcal{X}$ is exponential in H.

D Details of the experiments

In this appendix we describe each domain in detail and include examples of RDPs learned by ADACT-H.

D.1 Corridor

This RDP example was introduced in Ronca and De Giacomo [55]. The environment consists of a $2 \times m$ grid, with only two actions a_0 and a_1 which moves the agent to states $(0, i+1)$ and $(1, i+1)$ respectively from state $(\cdot, i)$. The goal of the agent is to avoid an enemy which is present in position $(0, i)$ with probability p_i^0, and at $(1, i)$ with probability p_i^1. The agent receives a reward of $+1$ for avoiding the enemy at a particular column, and the probabilities p_i^0 and p_i^i are switched every time it encounters the enemy. When the agent reaches the last column, its position is reset to the first column. The observation space is given by (i, j, e), where i, j is the cell position of the agent and $e \in \{\textit{enemy}, \textit{clear}\}$ denotes the presence of the guard in the current cell. Fig 2 shows the minimal automaton obtained by all three algorithms for $H = 5$.

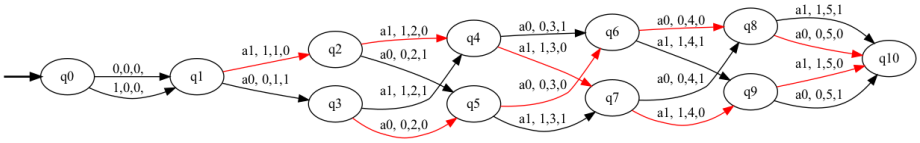


Figure 2: Automaton obtained from the corridor environment. The edges are labelled as [*action*, *observation*, *enemy*].

D.2 T-maze

The T-maze environment was introduced by Bakker [3] to capture long term dependencies with RL-LSTMs. As shown in Figure 1, at the initial position S, the agent receives an observation X, depending on the position of the goal state G in the last column. The agent can take four actions, *North*, *South*, *East* and *West*. The agent receives a reward of $+4$ on taking the correct action at the T-junction, and -1 otherwise, terminating the episode. The agent also recieves a -1 reward for standing still. At the initial state the agent receives observation 011 or 110, 101 throughout the corridor and 010 at the T-junction. Fig 1 shows the optimal automaton obtained when the available actions in the corridor are restricted to only $East$ (the automaton obtained without this restriction is shown in Fig. 5). Table 1 shows our results with the unrestricted action space. Both our approaches find the optimal policy in this case, unlike Flex-Fringe which fails to capture this long term dependency.

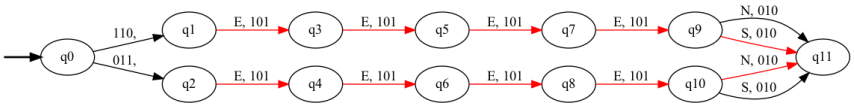


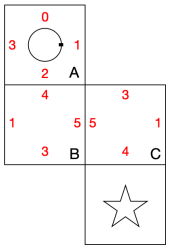
Figure 3: Automaton obtained from T-maze environment with restricted actions. The edges are labelled as [*action*, *observation*, *reward*].

D.3 Cookie domain

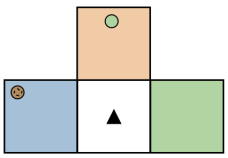
We modify the original *cookie domain* as described in Icarte et al. [64], to a simpler domain consisting of 4 rooms, *blue*, *white*, *green* and *red* as shown in Fig. 4b. If the agent presses the button in room *red*, a cookie appears in room *blue* or *green* with equal probability. The agent can move *left*, *right*, *up* or *down*, can *press* the button in room *red*, and *eat* the cookie to receive a reward 1, and then it may press the button again. There are 6 possible observations (4 for each room, and 2 for observing the *cookie* in the two rooms). We use the set $\mathcal{X}_{1,1,1}$ for distinguishability in the restricted language case. Our restricted language approach here finds the optimal policy and the smallest state space.

D.4 Cheese maze

Cheese maze [42] consists of 10 states, and 6 observations, and 4 actions. After reaching the goal state, the agent receives a reward of $+1$, and the position of the agent is reintialised to one of the non-goal states with equal probability. Our restricted language approach uses the set $\mathcal{X}_{1,1,1}$. For a



(a) Mini-hall [36] environment.



(b) Simplified cookie domain.

Figure 4: Environments.

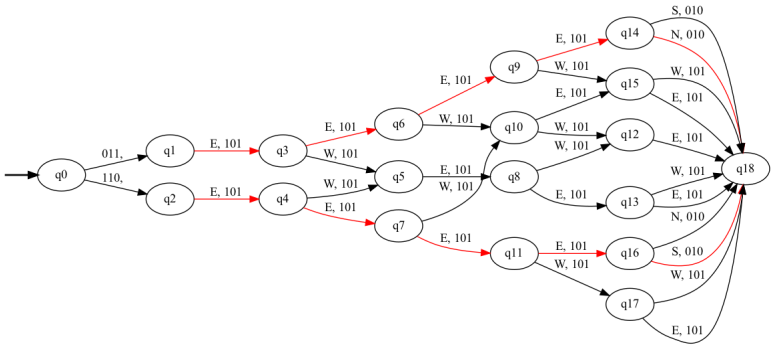


Figure 5: Automaton obtained from T-maze (partially restricted action space), restricted language.

horizon of 6, the results for the restricted language and Flex-Fringe are comparable, however upon further increasing the horizon, Flex-Fringe outperforms ADACT-H by learning cyclic RDPs which is not possible in our approach.

D.5 Mini-hall

The mini-hall environment [36] shown in Fig 4a has 12 states, 4 orientations in 3 rooms, a goal state given by a star associated with a reward of $+1$, 6 observation and 3 actions, and the position of the agent is reset after the goal is reached. This setting is much more complex than the others because 12 states are mapped into 6 observations; for example, starting from observation 3, 3 actions are required under the optimal policy to solve the problem if the starting underlying state was in Room B or C. We use the set $\mathcal{X}_{1,1,1}$ in our restricted language approach for distinguishability. Although we get a much larger state space, our algorithm gets closer to the optimal policy. However our CMS approach is not efficient in this case and exceeds the alloted time budget of 1800 seconds, as it requires to iterate over the entire length of trajectories.