Policy Optimization with Linear Temporal Logic Constraints

Cameron Voloshin
Caltech

Hoang M. Le
Argo AI

Swarat Chaudhuri
UT Austin

Yisong Yue
Argo AI
Caltech

Abstract

We study the problem of policy optimization (PO) with linear temporal logic (LTL) constraints. The language of LTL allows flexible description of tasks that may be unnatural to encode as a scalar cost function. We consider LTL-constrained PO as a systematic framework, decoupling task specification from policy selection, and as an alternative to the standard of cost shaping. With access to a generative model, we develop a model-based approach that enjoys a sample complexity analysis for guaranteeing both task satisfaction and cost optimality (through a reduction to a reachability problem). Empirically, our algorithm can achieve strong performance even in low-sample regimes.

1 Introduction

The standard reinforcement learning (RL) framework aims to find a policy that minimizes a cost function. The premise is that this scalar cost function can completely capture the task specification (known as the "reward hypothesis" [55, 53]). To date, almost all theoretical understanding of RL is focused on this cost minimization setting (e.g., [62, 32, 31, 57, 45, 9, 24, 19, 10, 3, 4, 40, 47, 48]).

However, capturing real-world task specifications using scalar costs can be challenging. For one, real-world tasks often consist of objectives that are required, as well as those that are merely desirable. By combining these objectives into scalar costs, one erases the distinction between these two categories of tasks. Also, there is recent theoretical evidence that certain tasks are simply not reducible to scalar costs [1] (see Section 2). In practice, one circumvents these challenges using heuristics such as adding "breadcrumbs" [54]. However, such heuristics can lead to catastrophic failures in which the learning agent ends up exploiting the cost function in an unanticipated way [49, 61, 28, 68, 44].

In response to these limitations, recent work has studied alternative RL paradigms that use Linear Temporal Logic (LTL) to specify tasks (see Section 7). LTL is a modeling language that can express desired characteristics of future paths of the system [11]. The notation is precise enough to allow the specification of both the required and desired behaviors; the cost minimization is left only to discriminate between which LTL-satisfying policy is "best". This ensures that the main objective — e.g., time, energy, or effort — does not have any relation to the task and is easily interpretable.

Existing work on RL with LTL constraints tends to make highly restrictive assumptions. Examples include (i) known mixing time of the optimal policy [23], (ii) the assumption that every policy satisfies the task eventually [64], or (iii) known optimal discount factor [26], all of which assist in task satisfaction verification. These assumptions have complex interactions with the environment, making them impractical if not impossible to calculate. The situation is made more complex by recent theoretical results [66, 7] that show that there are LTL tasks that are not PAC-MDP-learnable.

In this paper, we address these limitations through a novel policy optimization framework for RL under LTL constraints. Our approach relies on two assumptions that are significantly less restrictive than those in prior work and circumvent the negative results on RL-modulo-LTL: the availability

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

of a generative model of the environment and a lower bound on the transition probabilities in the underlying MDP. Under these assumptions, we derive a learning algorithm based on a reduction to a reachability problem. The reduction in our method can be instantiated with several planning procedures that handle unknown dynamics [12, 46]. We show that our algorithm offers strong constraint satisfaction guarantees and give a rigorous sample complexity analysis of the algorithm.

In summary, the contributions of this paper are:

1. We provide a novel approach to LTL-constrained RL that requires significantly fewer assumptions, and offers stronger guarantees, than previous work.
2. We develop several new theoretical tools for our analysis. These may be of independent interest.
3. We empirically validate using both infinite- and indefinite-horizon problems, and with composite specifications such as collecting items while avoiding enemies. We find that our method enjoys strong performance, often requiring many fewer samples than our worst-case guarantees.

2   Motivating Examples

We examine two examples where standard cost engineering cannot capture the task (Figure 1). We consider the undiscounted setting here. See [41, 1] for difficult examples for the discounted setting.

Example 1 (Infinite Loop). A robot is given the task of perpetually walking between the coffee room and the office (Figure 1 (Left)). To achieve this behavior, both the policy and cost-function must be history-dependent. These can be made Markovian through proper state-space augmentation and has been studied in hierarchical reinforcement learning or learning with options [38, 56]. Options engineering is laborious and requires expertise. Nevertheless, without the appropriate augmentation, any cost-optimal policy of a Markovian cost function will fail at the task. We will see in Section 3 that any LTL expression comes with automatic state-space augmentation, requiring no expert input.

Example 2 (Safe Delivery). The goal is to maximize the probability of safely sending a packet from one computer to another (Figure 1 (Right)). Policy 1 leads to a hacker sniffing packets but passing them through, and is unsafe. Policy 2 leads to a hacker stealing packets with probability $p > 0$, and is safe with probability $1 - p$, and is the policy that satisfies the task. For cost engineering, let R and S be the recurring costs of the received and stolen states. For the two policies, the avg. costs are $g_1 = R$ and $g_2 = pS + (1 - p)R$. Strangely, we must set $R > S$ in order for $g_2 < g_1$. Fortunately, optimizing any cost function constrained to satisfying the LTL specification does not suffer from this counter intuitive behavior as only policy 2 has any chance of satisfying the LTL expression.

Figure 1: *(Left) Infinite Loop. The robot must perpetually walk between the coffee room and office. Without proper state-space augmentation, a markovian cost function cannot capture this task. (Right) Safe Delivery. The specification is to deliver a packet without being interfered. Policy 2 should be chosen. One would need to penalize receiving the packet significantly over having it stolen: $R > S$.*

3   Background and Problem Formulation

We now formulate the problem. An *atomic proposition* is a variable that takes on a truth value. An *alphabet* over a set of atomic propositions AP is given by $\Sigma = 2^{\text{AP}}$. For example, if AP $= \{a, b\}$ then $\Sigma = \{\{\}, \{a\}, \{b\}, \{a, b\}\}$. $\Delta(X)$ represents the set of probability distributions over a set X.

3.1   MDPs with Labelled State Spaces

We assume that the environment follows the finite Markov Decision Process (MDP) framework given by the tuple $\mathcal{M} = (\mathcal{S}^{\mathcal{M}}, \mathcal{A}^{\mathcal{M}}, P^{\mathcal{M}}, \mathcal{C}^{\mathcal{M}}, d_0^{\mathcal{M}}, L^{\mathcal{M}})$ consisting of a finite state space $\mathcal{S}^{\mathcal{M}}$, a finite

2

action space $\mathcal{A}^{\mathcal{M}}$, an ***unknown*** transition function $P^{\mathcal{M}} : \mathcal{S}^{\mathcal{M}} \times \mathcal{A}^{\mathcal{M}} \to \Delta(\mathcal{S}^{\mathcal{M}})$, a cost function $\mathcal{C} : \mathcal{S}^{\mathcal{M}} \times \mathcal{A}^{\mathcal{M}} \to \Delta([c_{\min}, c_{\max}])$, an initial state distribution $d_0 \in \Delta(\mathcal{S}^{\mathcal{M}})$, and a labelling function $L^{\mathcal{M}} : \mathcal{S}^{\mathcal{M}} \to \Sigma$. We take $\mathcal{A}^{\mathcal{M}}(s)$ to be the set of available actions in state s. Unlike traditional MDPs, $\mathcal{M}$ has a labeling function $L^{\mathcal{M}}$ which returns the atomic propositions that are true in that state. A **run** in $\mathcal{M}$ is a sequence of states $\tau = (s_0, s_1, \ldots)$ reached through successive transitions.

3.2 Linear Temporal Logic (LTL), Synchronization with MDPs, and Satisfaction

Now we give some basic background on LTL. For a more comprehensive overview, see [11].

Definition 3.1 (LTL Specification, φ). An LTL specification φ is the entire description of the task, including both desired and required behaviors, and is constructed from a composition of atomic propositions, including logical connectives: not ($\neg$), and ($\&$), and implies ($\to$); and temporal operators: next (X), repeatedly/always/globally (G), eventually (F), and until (U).

Examples. Consider again the examples in Section 2. For $AP = \{a, b\}$, some basic task specifications include safety ($G\neg a$), reachability (Fa), stability (FGa), response ($a \to Fb$), and progress ($a \, \& \, XFb$). For the Infinite Loop example (Figure 1 (Left)), $AP = \{o, c\}$ indicating the label of the grid location of our agent (office, coffee, or neither). The specification is "$GF(o \, \& \, XFc)$" meaning "go between office and coffee forever", and is a combination of safety, reachability, and progress. For the Safe Delivery example (Figure 1 (Right)), $AP = \{s\}$ indicating the safety of a state. The specification is "Gs" meaning "always be safe".

LTL Satisfaction: Synchronizing MDP with LTL. By synchronizing an MDP with an LTL formula, we can easily check if a run in the MDP satisfies a specification φ. In particular, it is possible to model the progression of satisfying φ through a specialized automaton, an LDBA $\mathcal{B}_\varphi$ [52], defined below. More details for constructing LDBAs are in [25, 11, 35]. We drop φ from $\mathcal{B}_\varphi$ for brevity.

Definition 3.2. (Limit Deterministic Büchi Automaton, LDBA [52]) An ***LDBA*** is a tuple $\mathcal{B} = (\mathcal{S}^{\mathcal{B}}, \Sigma \cup \mathcal{A}_{\mathcal{B}}, P^{\mathcal{B}}, \mathcal{S}^{\mathcal{B}*}, s_0^{\mathcal{B}})$ consisting of (i) a finite set of states $\mathcal{S}^{\mathcal{B}}$, (ii) a finite alphabet $\Sigma = 2^{\text{AP}}$, $\mathcal{A}_{\mathcal{B}}$ is a set of indexed jump transitions (iii) a transition function $P^{\mathcal{B}} : \mathcal{S}^{\mathcal{B}} \times (\Sigma \cup \mathcal{A}_{\mathcal{B}}) \to 2^{\mathcal{S}^{\mathcal{B}}}$, (iv) accepting states $\mathcal{S}^{\mathcal{B}*} \subseteq \mathcal{S}^{\mathcal{B}}$, and (v) initial state $s_0^{\mathcal{B}}$. There exists a mutually exclusive partitioning of $\mathcal{S}^{\mathcal{B}} = \mathcal{S}_D^{\mathcal{B}} \cup \mathcal{S}_N^{\mathcal{B}}$ such that $\mathcal{S}^{\mathcal{B}*} \subseteq \mathcal{S}_D^{\mathcal{B}}$, and for $s \in S_D^{\mathcal{B}}, a \in \Sigma$ then $P^{\mathcal{B}}(s, a) \subseteq \mathcal{S}_D^{\mathcal{B}}$ and $|P^{\mathcal{B}}(s, a)| = 1$, deterministic. $\mathcal{A}_{\mathcal{B}}(s)$ is only (possibly) non-empty for $s \in \mathcal{S}_D^{\mathcal{B}}$ and allows $\mathcal{B}$ to transition without reading an AP. A *path* $\sigma = (s_0, s_1, \ldots)$ is a sequence of states in $\mathcal{B}$ reached through successive transitions. $\mathcal{B}$ **accepts** a path σ if there exists some state $s \in \mathcal{S}^{\mathcal{B}*}$ in the path that is visited infinitely often.

We can now construct a synchronized product MDP from the interaction of $\mathcal{M}$ and $\mathcal{B}$.

Definition 3.3. (Product MDP) The product MDP $\mathcal{X}_{\mathcal{M},\mathcal{B}} = (\mathcal{S}, \mathcal{A}, P, \mathcal{C}, d_0, L, \mathcal{S}^*)$ is an MDP with $\mathcal{S} = \mathcal{S}^{\mathcal{M}} \times \mathcal{S}^{\mathcal{B}}$, $\mathcal{A} = \mathcal{A}^{\mathcal{M}} \cup \mathcal{A}^{\mathcal{B}}$, $\mathcal{C}((m, b), a) = \mathcal{C}^{\mathcal{M}}(m, a)$ if $a \in A^{\mathcal{M}}(m)$ otherwise 0, $d_0 = \{(m, b) | m \in d_0^{\mathcal{M}}, b \in P^{\mathcal{B}}(s_0^{\mathcal{B}}, L^{\mathcal{M}}(m))\}$, $L((m, b)) = L^{\mathcal{M}}(m)$, $S^* = \{(\cdot, b) \in \mathcal{S} | b \in \mathcal{S}^{\mathcal{B}*}\}$ accepting states, and $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ taking the form:

$$
P((m, b), a, (m', b')) = \begin{cases} P^{\mathcal{M}}(m, a, m') & a \in A^{\mathcal{M}}(m), b' \in P^{\mathcal{B}}(b, L(m')) \\ 1, & a \in A^{\mathcal{B}}(b), b' \in P^{\mathcal{B}}(b, a), m = m' \\ 0, & \text{otherwise} \end{cases}
$$

A run $\tau = (s_0, s_1, \ldots) = ((m_0, b_0), (m_1, b_1), \ldots)$ in $\mathcal{X}$ is accepting (accepted) if $(b_0, b_1, \ldots)$, the projection onto $\mathcal{B}$, is accepted. Equivalently, some $s \in \mathcal{S}^*$ in $\mathcal{X}$ is visited infinitely often. This leads us to the following definition of LTL satisfaction:

Definition 3.4 (Satisfaction, $\tau \models \varphi$). A run τ in $\mathcal{X}$ *satisfies* φ, denoted $\tau \models \varphi$, if it is accepted.

Definition 3.5. (Satisfaction, $\pi \models \varphi$) A policy $\pi \in \Pi$ *satisfies* φ with probability $\mathbb{P}[\pi \models \varphi] = \mathbb{E}_{\tau \sim \text{T}_\pi^P}[\mathbf{1}_{\tau \models \varphi}]$. Here, $\mathbf{1}_X$ is an indicator variable which is 1 when X is true, otherwise 0. T_π^P is the set of trajectories induced by π in $\mathcal{X}$ with transition function P.

3.3 Problem Formulation

Our goal is to find a policy that simultaneously satisfies a given LTL specification φ with highest probability (probability-optimal) and is also optimal w.r.t. the cost function of the MDP. We consider

3

(stochastic) Markovian policies Π, and define the set of all probability-optimal policies as $\Pi_{\max} = \{\arg\max_{\pi'\in\Pi} \mathbb{P}[\pi' \models \varphi]\}$. We first define the gain g (average-cost) and transient cost J:

$$ g_\pi^P \equiv \mathbb{E}_{\tau\sim\mathrm{T}_\pi^P} \left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathcal{C}(s_t, \pi(s_t)) \Big| \tau \models \varphi \right], \; J_\pi^P \equiv \mathbb{E}_{\tau\sim\mathrm{T}_\pi^P} \left[\sum_{t=0}^{\kappa_\tau} \mathcal{C}(s_t, \pi(s_t)) \Big| \tau \models \varphi \right] \quad (1) $$

where κ_τ is the first (hitting) time the trajectory τ leaves the transient states induced by π. When P is clear from context, we abbreviate g_π^P and J_π^P by g_π and J_π, respectively.

Gain optimality for infinite horizon problems has a long history in RL [12, 46]. Complementary to gain optimality, we consider a hybrid objective including the transient cost. For any $\lambda \geq 0$, we define the optimal policy as the probability-optimal policy with minimum combined cost:

$$ \pi_\lambda^* \equiv \arg\min_{\pi\in\Pi_{\max}} J_\pi + \lambda g_\pi = \arg\min_{\pi\in\Pi_{\max}} (J_\pi + \lambda g_\pi)\mathbb{P}[\pi \models \varphi] \quad (\equiv V_{\pi,\lambda}^P). \quad \text{(OPT)} $$

In other words, probability-optimal policies are those that satisfy the entirety of the task, both desired and required behaviors, where $V_{\pi,\lambda}^P \equiv (J_\pi + \lambda g_\pi)\mathbb{P}[\pi \models \varphi]$ is the normalized value function[1], corresponding to a notion of energy or effort required, with λ representing the tradeoff between gain and transient cost. We will often omit the dependence of V on P and λ for brevity.

Example. Consider the Safe Delivery example (Figure 1 (Right)). For policy 1, $\mathbb{P}[1 \models \varphi] = 0$ and so $1 \notin \Pi_{\max}$. Let policy 2 be a cost 1 timestep before stolen or receipt, then $g_2 = R$ is the (conditional) gain, $J_2 = 1$ is the (conditional) transient costs, $\mathbb{P}[2 \models \varphi] = 1 - p$, and $V_2 = (1 + \lambda R)(1 - p)$.

Problem 1 (Planning with Generative Model/Simulator). Suppose access to a generative model of the true dynamics P from which we can sample transitions $s' \sim P(s, a)$ for any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$.[2] With probability $1 - \delta$, for some errors $\epsilon_\varphi, \epsilon_V > 0$, find a policy $\pi \in \Pi$ that simultaneously has the following properties: (i) $|\mathbb{P}[\pi \models \varphi] - \mathbb{P}[\pi^* \models \varphi]| < \epsilon_\varphi$ (ii) $|V_\pi - V_{\pi^*}| < \epsilon_V$.

4 Approach

4.1 End Components & Accepting Maximal End Components

Our analysis relies on the idea of an end component: a recurrent, inescapable set of states when restricted to a certain action set. It is a sub-MDP of a larger MDP that is probabilistically closed.

Definition 4.1. (End Component, EC/MEC/AMEC [11]) Consider MDP $(\mathcal{S}, \mathcal{A}, P, \mathcal{C}, d_0, L, \mathcal{S}^*)$. An end component $(E, \mathcal{A}_E)$ is a set of states $E \subseteq \mathcal{S}$ and acceptable actions $\mathcal{A}_E(s) \subseteq \mathcal{A}(s)$ (where $s \in E$) such that $\forall(s, a) \in E \times \mathcal{A}_E$ then $Post(s, a) = \{s'|P(s, a, s') > 0\} \subseteq E$. Furthermore, $(E, \mathcal{A}_E)$ is strongly connected: any two states in E is reachable from one another by means of actions in $\mathcal{A}_E$. We say an end component $(E, \mathcal{A}_E)$ is *maximal* (MEC) if it is not contained within a larger end component $(E', \mathcal{A}_{E'})$, ie. $\nexists(E', \mathcal{A}_{E'})$ EC where $E \subseteq E', \mathcal{A}_E(s) \subseteq \mathcal{A}_{E'}(s)$ for each $s \in A$. A MEC $(E, \mathcal{A}_E)$ is an *accepting* MEC (AMEC) if it contains an accepting state, $\exists s \in E$ s.t. $s \in \mathcal{S}^*$.

4.2 High-Level Intuition

The description of our approach, LTL Constrained Planning (LCP), in Section 4.4 is rather technical in order to yield theoretical guarantees. We thus first summarize the high-level intuitions.

Solution Decomposition. Consider the accepting states s_1^*, s_2^* in Figure 2 (Left), which are the states we need to visit infinitely often to satisfy the specification. First, let us identify the accepting maximal end components (AMECs) of s_1^* and s_2^*: the state sets A_1 and A_2 (resp.) and their corresponding action sets $\mathcal{A}_{A_1}$ and $\mathcal{A}_{A_2}$ (the blue arrows in A_1 and A_2). Note that these AMECs do not include the yellow action in Figure 2 (Left), which has a chance of leaving A_1 and getting stuck in A_3.

Our solution first runs a *transient* policy until reaching A_1 or A_2, and then switches to a (probability-optimal) *recurrent* policy that stays within A_1 or A_2 (resp.) while visiting s_1^* or s_2^* (resp.) infinitely often. A probability-optimal *recurrent* policy will select actions in $\mathcal{A}_{A_1}$ and $\mathcal{A}_{A_2}$ to visit s_1^*, s_2^*

---

[1]Normalized objectives are not unusual in RL, e.g. in discounted settings, multiplication by $(1 - \gamma)$

[2]The use of a generative model is increasingly common in RL [24, 40, 3, 58], and is applicable in many settings where such a generative model is readily available as a simulator (e.g., [21]).

4

(a) Abstract Diagram

(b) Example, Infinite Loop

(c) Example, Safe Delivery
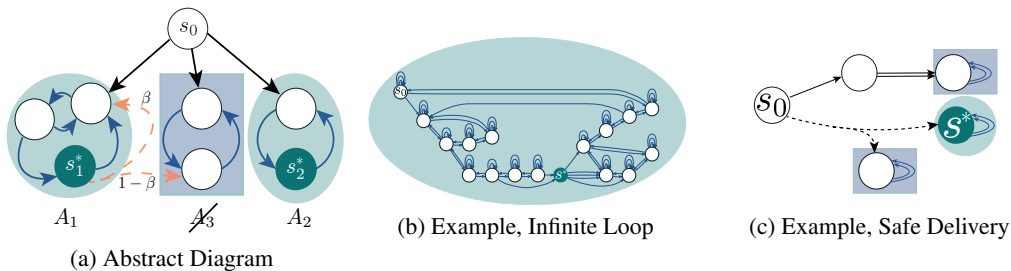
Figure 2: Product MDP diagrams. (Left) The goal of LTL Constrained Policy Optimization can be reduced to a reachability problem. We want to reach A_1 or A_2 from s_0 and then follow the blue arrows with some distribution. A_3 with the blue arrows is a rejecting end component because it does not contain an accepting state s^*. For $\beta < 1$, the yellow action is not in the allowable action set of A_1 because there is a risk of entering A_3, strictly decreasing our probability of LTL satisfaction. (Center) Example for Infinite Loop, Figure 1 Left. (Right) Example for Safe Delivery, Figure 1 Right.

infinitely often (e.g., the uniform policies with the AMECs $(A_1, \mathcal{A}_{A_1})$ and $(A_2, \mathcal{A}_{A_2})$). Finding a *transient* policy from s_0 to A_1, A_2 can be viewed as a reachability problem, which we can solve via a Stochastic Shortest Path (SSP) problem and leverage recent literature [58, 34].

Cost Optimality. As stated in OPT, the goal is to find a cost-optimal policy within the set of probability-optimal policies. For instance, the uniform policy over $\mathcal{A}_{A_1}$ and $\mathcal{A}_{A_2}$ (the blue arrows in Figure 2 (Left) is probability optimal, but may not be cost optimal. Similarly, the unconstrained cost-optimal policy may not be probability optimal. Consider just A_1 for the moment. Suppose the cost of the arrows between the white nodes is 4 while the other costs are 7. Then the uniform (probability-optimal) policy in A_1 over $\mathcal{A}_{A_1}$ has cost $\frac{1}{2}\left(\frac{4+4}{2}\right) + \frac{1}{2}\left(\frac{7+7+4}{3}\right) = 5$. The gain-optimal policy that deterministically selects the actions between the white nodes $\tilde{\pi}$ has cost $\left(\frac{4+4}{2}\right) = 4$, but is not probability optimal. If we perturb $\tilde{\pi}$ to make it even slightly stochastic (but still mostly deterministic, i.e η-greedy with $\eta \approx 0$), then it will be arbitrarily close to gain optimality and also recover probability optimality. This is a preferable probability-optimal policy over the uniform policy.

Overall Procedure. The high-level procedure is: (i) identify the AMECs (e.g. $(A_1, \mathcal{A}_{A_1}), (A_2, \mathcal{A}_{2})$) by filtering out bad actions like the yellow arrow; (ii) find a cost-optimal (optimal gain cost) recurrent policy in each AMEC that visits some s^* infinitely often; (iii) instantiate an SSP problem that finds a cost-optimal (optimal transient cost) transient policy from s_0 to $A_1 \cup A_2$ and avoids A_3; (iv) return a policy that stitches together the policies from (ii) and (iii). See Section 4.4 for the algorithmic details. We show in Section 5 that this solution gives the optimal solution to OPT.

4.3 Additional Assumptions and Definitions

Perhaps surprisingly, when planning with a simulator (i.e., generative model), even infinite data is insufficient to verify an LTL formula without having a known lower-bound on the lowest nonzero probability of the transition function P [41]. Without this assumption, LTL constrained policy learning is not learnable [66]. We thus begin by assuming a known lower bound on entries in P.[3]

Assumption 1 (Lower Bound). *We assume we have access to a lower bound $\beta > 0$ on the lowest non-zero probability of the transition function P (Sec. 3.1):*

$$0 < \beta \leq \min_{s,a,s' \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}} \{P(s,a,s') | P(s,a,s') > 0\}. \tag{2}$$

We assume that all the costs are strictly positive, avoiding zero-cost (or negative-cost) cycles that trap a policy. Leveraging cost-perturbations and prior work [58] can remove the assumption.

Assumption 2 (Bounds on cost function). *The minimum cost $c_{\min} > 0$ (Sec. 3.1) is strictly positive.*

Let $D = \{(s,a,s')\}$ be all the collected samples (s,a,s') while running the algorithm. At any point, $\widehat{P}(s,a,s') = \frac{|\{(s,a,s') \in D\}|}{|\{(s,a) \in D\}|}$ is the empirical frequency of visiting s' from (s,a). We introduce

---

[3]Our assumptions are consistent with the minimal requirements studied by [41]

5

an event $\mathcal{E}$ and error $\psi(n)$ to quantify uncertainty on $\widehat{P}(s, a, s')$ based on current data: $n(s, a) = |\{(s, a) \in D\}|$. $\mathcal{E}$ is based on empirical Bernstein bounds [42], and holds w.p. $1 - \delta$ (Lemma B.1).

Definition 4.2 (High Probability Event). A high probability event $\mathcal{E}$:

$$\mathcal{E} = \{\forall s, a, s' \in S \times A \times S, \forall n(s, a) > 1 : |(P(s, a, s') - \widehat{P}(s, a, s'))| \leq \psi_{sas'}(n) \leq \psi(n)\},$$

where $\psi_{sas'}(n) \equiv \sqrt{2\widehat{P}(s, a, s')(1 - \widehat{P}(s, a, s')))\xi(n)} + \frac{7}{3}\xi(n)$, $\psi(n) \equiv \sqrt{\frac{1}{2}\xi(n)} + \frac{7}{3}\xi(n)$, and $\xi(n) \equiv \log(\frac{4n^2|\mathcal{S}|^2|\mathcal{A}|}{\delta})/(n - 1)$.

Remark 4.1. *For some $\rho > 0$, if we require $|P(s, a, s') - \widehat{P}(s, a, s')| \leq \rho$ then we need $n(s, a) = \psi^{-1}(\rho)$ samples for state-action pair (s, a). See Lemma B.2 for the quantity $\psi^{-1}(\rho)$.*

Definition 4.3 (Plausible Transition Function). The set of plausible transition functions is given by

$$\mathcal{P} = \{\tilde{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})| \begin{cases} \tilde{P}(s, a, s') = \widehat{P}(s, a, s'), & \widehat{P}(s, a, s') \in \{0, 1\} \\ \tilde{P}(s, a, s') \in \widehat{P}(s, a, s') \pm \psi_{sas'} \cap [\beta, 1 - \beta], & \text{otherwise} \end{cases}\} \quad (3)$$

Let $\mathcal{P}(s, a) \equiv \{P(s, a, \cdot)|P \in \mathcal{P}\}$ be the possible transition distributions for state-action pair (s, a). We denote $P_\pi(s, s') = \mathbb{E}_{a \sim \pi}[P(s, a, s')]$ as the Markov chain given dynamics P with policy π, and can be thought of as a $|\mathcal{S}| \times |\mathcal{S}|$ matrix $P_\pi = \{p_{ij}\}_{i,j=1}^{|\mathcal{S}|}$.

4.4 Main Algorithm: LTL Constrained Planning (LCP)

Algorithm 1 LTL Constrained Planning (LCP)

Param: Error $\epsilon_V > 0$, Error $\epsilon_\varphi > 0$, Tolerance $\delta > 0$, Lower bound $\beta > 0$ (see Assumption 1)
1: Globally, track $\widehat{P}(s, a, s') = \frac{|\{(s, a, s') \in D\}|}{|\{(s, a) \in D\}|}$ // Empirical estimate of P
2: $((A_1, \mathcal{A}_{A_1}), \ldots, (A_m, \mathcal{A}_{A_k})) \leftarrow$ FindAMEC$((\mathcal{S}, \mathcal{A}, \widehat{P}))$
3: **for** $i = 1, \ldots, k$ **do**
4:     Set $\pi_i, g_i \leftarrow$ PlanRecurrent$((A_i, \mathcal{A}_{A_i}), \frac{\epsilon_V}{7\lambda})$ // Plan gain-optimal policy π_i for A_i
5: Set $\pi_0 \leftarrow$ PlanTransient$(((A_1, g_1), \ldots, (A_k, g_k)), \frac{2\epsilon_V}{9})$ // Plan shortest paths policy π_0 to $\cup_{i=1}^k A_i$
6: **return** $\pi = \cup_{i=0}^k \pi_i$

Our approach, LTL Constrained Planning (LCP), has three components, as shown in Algorithm 1 and described below. Recall from Problem 1 that the policy optimization problem OPT is instantiated over a product MDP (Def. 3.3), and that we are given a generative model of the true dynamics P from which we can sample transitions $s' \sim P(s, a)$ for any state/action pair.

Finding AMECs (FindAMEC). After sampling each state-action pair $\phi_{\text{FindAMEC}} = O(\frac{1}{\beta})$ times (see Prop. B.4), by Assumption 1, we can verify the support of P. We can compute all of the MECs using Algorithm 47 from [11]. Among these MECs, we keep the AMECs, which amounts to checking if the MEC $(A_i, \mathcal{A}_{A_i})$ contains an accepting state $s^* \in \mathcal{S}^*$ from the given product MDP.

PlanRecurrent (PR). To plan in each AMEC $(A, \mathcal{A}_A)$ (i.e., find the optimal recurrent policy), we use Alg. 2 with (extended) relative value iteration (VI, Alg. 4 in appendix) using the optimistic Bellman operator $\mathcal{L}_{\text{PR}}^\alpha$ (see Table 1, we discuss α in next paragraph). Let π_v denote the greedy policy w.r.t. the fixed point $v = \mathcal{L}_{\text{PR}}^\alpha v$ (v is the optimistic value estimate). Using the η-greedy policy, $\pi \equiv (1 - \eta)\pi_v + \eta$Unif$(\mathcal{A}_A)$ (Alg. 2, Line 7), together with P_π, makes A recurrent: $s^* \in A$ is visited infinitely often and $\mathbb{P}[\pi \models \varphi|s_0 \in A] = 1$. Since η can be arbitrarily small (Lemma B.7), then $g_\pi \approx g_{\pi_v}$ and π is both cost and probability optimal. As intuited in Section 4.2, π has full support over $\mathcal{A}_A$ but is nearly deterministic.[4]

Algorithm 2 PlanRecurrent (PR)

Param: AMEC $(A, \mathcal{A}_A)$, error $\epsilon_{\text{PR}} > 0$
1: Set $\rho \leftarrow 2\psi(\phi_{\text{FindAMEC}}(\beta))$ // $\rho \sim \|P - \tilde{P}\|_1^{-1}$
2: **repeat**
3:     Set $\rho \leftarrow \frac{\rho}{2}$
4:     Sample $\psi^{-1}(\rho)$ times $\forall(s, a) \in A \times \mathcal{A}_A$
5:     $v', v, \tilde{P} \leftarrow$ VI$(\mathcal{L}_{\text{PR}}^\alpha, d_{\text{PR}}, \epsilon_{\text{PR}}^{\mathcal{L}})$ // $v' = \mathcal{L}_{\text{PR}}^\alpha v$
6: **until** $\rho > \frac{\epsilon_{\text{PR}}(1 - \Delta(\tilde{P}))}{3|A|c_{\max}}$ // $\|P - \tilde{P}\|_1$ *small*
7: Set policy $\pi \leftarrow \eta$-greedy policy w.r.t. v'
8: Set gain $g_\pi \leftarrow \frac{1}{2}(\max(v' - v) + \min(v' - v))$
9: **return** π, g_π

---
[4]Typically, RL settings admit a fully deterministic optimal policy, but for LTL constrained policy optimization the optimal policy may not be deterministic (although can be very nearly so). See Cost Optimality in Section 4.2

6

Table 1: Subroutine Operators and Parameters for Value Iteration

Op/Param	Description
$\mathcal{L}_{\text{PR}}^{\alpha}v(s)$	$\min_{a \in \mathcal{A}_A(s)} \left(\mathcal{C}(s,a) + \alpha \min_{p \in \mathcal{P}(s,a)} p^T v \right) + (1-\alpha)v(s) \quad \forall s \in A$
$d_{\text{PR}}(v_{n+1}, v_n) < \epsilon_{\text{PR}}^{\mathcal{L}}$	$\max_{s \in A}(v_{n+1}(s) - v_n(s)) - \min_{s \in A}(v_{n+1}(s) - v_n(s)) < \frac{2\epsilon_{\text{PR}}}{3}$
$\mathcal{L}_{\text{PT}}v(s)$	$\begin{cases} \min\left\{ \min_{a \in \mathcal{A}_A(s)} \left(\mathcal{C}(s,a) + \min_{p \in \mathcal{P}(s,a)} p^T v \right), \bar{V}/\epsilon_{\varphi} \right\}, & s \in \mathcal{S} \setminus \cup_{i=1}^{k} A_i \\ \lambda g_i, & s \in A_i \end{cases}$
$d_{\text{PT}}(v_{n+1}, v_n) < \epsilon_{\text{PT}}^{\mathcal{L}}$	$\|v_{n+1} - v_n\|_1 < c_{\min}\epsilon_{\text{PT}}\epsilon_{\varphi}/(4\bar{V})$

`VI` in Line 5 of Alg. 2 is an iterative procedure (Alg. 4 in appendix), and terminates via $d_{\text{PR}} < \epsilon_{\text{PR}}^{\mathcal{L}}$ (Table 1). Convergence of extended `VI` is guaranteed [46, 29, 22], so long as the dynamics, $\tilde{P} = \arg\min_{p \in \mathcal{P}(s,a)} p^T v$, achieving the inner minimization of $\mathcal{L}_{\text{PR}}^{\alpha}$ are aperiodic – hence the aperiodicity transform $\alpha \in (0,1)$ in $\mathcal{L}_{\text{PR}}^{\alpha}$ [46]. Computing $\tilde{P}$ can be done efficiently [29] (Alg. 5 in appendix). For stability, we shift each entry of v_n by the value of the first entry $v_n(0)$ [12].

Alg. 2 returns the average gain cost g_{π} of policy π when we have enough samples for each state-action pair in $(A, \mathcal{A}_A)$ to verify that $n > \psi^{-1}\left(\frac{\epsilon_{\text{PR}}(1-\Delta(\tilde{P}_{\pi}))}{3|A|c_{\max}} \right)$ where $\Delta(\tilde{P}_{\pi}) = \frac{1}{2}\max_{ij}\sum_k |\tilde{p}_{ik} - \tilde{p}_{jk}|$. Here, $\Delta(\tilde{P}_{\pi})$ is an easily computable measure on the ergodicity of the Markov chain $\tilde{P}_{\pi}$ [18]. We track $\psi(n)$ (recall Def. 4.2) via a variable ρ and sample $\psi^{-1}(\rho) \approx \frac{1}{\rho^2}$ (see Lemma B.2) samples from each state-action pair in $(A, \mathcal{A}_A)$ (Alg. 2, Line 4). We halve ρ each iteration (Alg. 2, Line 3) and convergence is guaranteed because ρ will never fall below some unknown constant $\frac{\epsilon_{\text{PR}}(1-\bar{\Delta}_A)}{6|A|c_{\max}}$ (see Lemma B.8); the halving trick is required because $\bar{\Delta}_A$ is unknown a priori.

Proposition 4.2 (PR Convergence & Correctness, Informal). *Let π_A be the gain-optimal policy in AMEC $(A, \mathcal{A})$. Algorithm 2 terminates after at most $\log_2\left(\frac{6|A|c_{\max}}{\epsilon_{\text{PR}}(1-\bar{\Delta}_A)} \right)$ repeats, and collects at most $n = \tilde{\mathcal{O}}(\frac{|A|^2 c_{\max}^2}{\epsilon_{\text{PR}}^2(1-\bar{\Delta}_A)^2})$ samples for each $(s,a) \in (A, \mathcal{A}_A)$. The η-greedy policy π w.r.t. v' (Alg. 2, Line 5) is gain optimal and probability optimal: $|g_{\pi} - g_{\pi_A}| < \epsilon_{PR}$, $\mathbb{P}[\pi \models \varphi | s_0 \in A] = 1$.*

Algorithm 3 `PlanTransient (PT)`

Param: States & gains: $\{(A_i, g_i)\}_{i=1}^{k}$, err. $\epsilon_{\text{PT}} > 0$
1: Set $V_T(s) = \lambda g_i$ for $s \in A_i$ // *Terminal costs*
2: Sample ϕ_{PT} times $\forall (s,a) \in (\mathcal{S} \setminus \cup A_i) \times \mathcal{A}$
3: $v', v, \tilde{P} \leftarrow \text{VI}(\mathcal{L}_{\text{PT}}, d_{\text{PT}}, \epsilon_{\text{PT}}^{\mathcal{L}}, V_T)$ // $v' = \mathcal{L}_{\text{PT}}v$
4: Set $\pi \leftarrow$ greedy policy w.r.t v'
5: **return** π

PlanTransient (PT). This is the stochastic shortest path (SSP) reduction step that finds a policy from the initial state s_0 to the AMECs (Alg. 3). The main algorithmic tool used by `PlanTransient` is similar to that of `PlanRecurrent`: it also uses extended value iteration (`VI`, Alg. 4 in appendix) but with a different optimistic Bellman operator $\mathcal{L}_{\text{PT}}$ (Table 1), and then returns a (fully deterministic) greedy policy w.r.t. the resulting optimistic value v (Alg. 3, Line 4). $\mathcal{L}_{\text{PT}}$ is used to calculate the highest probability, lowest cost path to the AMECs (Alg. 3, Line 3).

Since rejecting end components might exist (see A_3 from Figure 2 (Left)), a trajectory may end up stuck and accumulate cost indefinitely, and so we must bound $\|v\|_{\infty} < \bar{V}/\epsilon_{\varphi}$ to prevent blow up. In Prop. B.13, we show how to select $\bar{V}$ such that π will reach the target states (in this case, the AMECs), first with high prob and then with lowest cost. The existence of such a bound on $\|v\|_{\infty}$ was shown to exist, without construction, in [34]. In practice, choosing a large $\bar{V}$ is enough.

The terminal costs V_T (Alg. 3, Line 1) together with Bellman equation $\mathcal{L}_{\text{PT}}$ has value function $\tilde{V}_{\pi} \approx p(J_{\pi} + \frac{1}{p}\sum_{i=1}^{k} p_i g_{\pi_i}) + (1-p)\bar{V}/\epsilon_{\varphi} \approx V_{\pi}$, relating to V_{π} (OPT), see Section A.1. Here, $p_i = \mathbb{P}[\pi \text{ reaches } A_i] \equiv \mathbb{E}_{\tau \sim \text{T}_{\pi}^{P}}[1_{\exists s \in \tau \text{ s.t } s \in A_i}]$ and $p = \sum_{i=1}^{k} p_i$. VI converges when $d_{\text{PT}} < \epsilon_{\text{PT}}$ (see Table 1). Convergence of extended VI for SSP is guaranteed [58, 34]. The number of samples required for each state-action pair $(s,a) \in (\mathcal{S} \setminus \cup A_i) \times \mathcal{A}$ is $\phi_{\text{PT}} = \psi^{-1}\left(\frac{c_{\min}\epsilon_{\text{PT}}\epsilon_{\varphi}^2}{14|\mathcal{S}\setminus\cup_{i=1}^{k} A_i|\bar{V}^2} \right)$.

Proposition 4.3 (`PlanTransient` Convergence & Correctness, Informal). *Denote the cost- and prob-optimal policy as π'. After collecting at most $n = \tilde{\mathcal{O}}(\frac{|\mathcal{S}\setminus\cup_{i=1}^{k} A_i|^2 \bar{V}^4}{c_{\min}^2 \epsilon_{PT}^2 \epsilon_{\varphi}^4})$ samples for each $(s,a) \in (\mathcal{S} \setminus \cup_{i=1}^{k} A_i) \times \mathcal{A}$, the greedy policy π w.r.t. v' (Alg. 3, Line 3) is both cost and probability optimal:*

$$\|\tilde{V}_{\pi} - \tilde{V}_{\pi'}\| < \epsilon_{PT}, \quad |\mathbb{P}[\pi \text{ reaches } \cup_{i=1}^{k} A_i] - \mathbb{P}[\pi' \text{ reaches } \cup_{i=1}^{k} A_i]| \leq \epsilon_{\varphi}.$$

7

5 End-To-End Guarantees

The number of samples necessary to guarantee an $(\epsilon_V, \epsilon_\varphi, \delta)$-PAC approximation to the cost-optimal and probability-optimal policy relies factors: β (lower bound on the min. non-zero transition probability of P), $\{c_{\min}, c_{\max}\}$ (bounds on the cost function $\mathcal{C}$), $\bar{\Delta}_{A_i}$ (worst-case coefficient of ergodicity for EC $(A_i, \mathcal{A}_{A_i})$), V (upper bound on the value function), and λ (tradeoff factor).

Theorem 5.1 (Sample Complexity). *Under the event $\mathcal{E}$, Assumption 1 and 2, after*

$$n = \tilde{\mathcal{O}}\left(\frac{1}{\beta} + \frac{1}{\epsilon_V^2}\left(\frac{|\mathcal{S}|^2 \bar{V}^4}{c_{\min}^2 \epsilon_\varphi^4} + \lambda^2 \sum_{i=1}^{k} \frac{|A_i|^2 c_{\max}^2}{(1 - \bar{\Delta}_{A_i})^2} \right) \right)$$

samples[5] are collected from each state-action pair, the policy π returned by Algorithm 1 is, with probability $1 - \delta$, simultaneously ϵ_V-cost optimal and ϵ_φ-probability optimal, satisfying:

$$(i) \ \ |\mathbb{P}[\pi \models \varphi] - \mathbb{P}[\pi^* \models \varphi]| \leq \epsilon_\varphi \quad (ii) \ \ \|V_\pi - V_{\pi^*}\|_\infty < \epsilon_V. \tag{4}$$

With a sufficiently large λ (which may not be verifiable in practice), π is also gain optimal.

Corollary 5.2 (Gain (Average Cost) Optimality). *There exists $\lambda^* > 0$ s.t. for $\lambda > \lambda^*$, the policy π returned by Alg. 1 satisfies (4), $g_\pi = \arg\min_{\pi' \in \Pi_{\max}} g_{\pi'}$, and is probability and gain optimal.*

The high-level structure of our analysis follows the algorithm structure in Section 4.4, via composing the constituent guarantees. To complete the analysis, we develop some technical tools which may be of independent interest, including a gain simulation Lemma B.8 and an η-greedy optimality Lemma B.7. For ease of exposition, we also ignore paths between AMECs (see Appendix D.2).

6 Empirical Analysis

We perform experiments in two domains: (1) Pacman domain where an agent finds food and indefinitely avoids a ghost; (2) discretized version of mountain car (MC) [14] where the agent must reach the flag. Our goal is to understand whether: (i) our LCP approach (Alg.1) produces competitive polices; (ii) LCP can work in continuous state spaces through discretization; (iii) LCP can enjoy efficient sample complexity in practice. For a baseline, we use Logically Constrained RL (LCRL, [26]), which is a Q-learning approach to LTL-constrained PO in unknown MDPs. We also do heavy cost shaping to LCRL as another baseline. See App E for more details, experiments, and figures.

6.1 Results

Competitiveness of the policy in full LTL specs? The probability of LCP satisfying the LTL spec in Figure 3 (Left) approaches 1 much faster than the two baselines. The returned policy collects the food quickly and then stays close, but avoids, the ghost. Any policy that avoids the ghost is equally good, as we have not incentivized it to stay far away. LCRL redefines cost as 1 if the LTL is solved and 0 otherwise, which is too sparse and learning suffers. Indeed, shaped LCRL performs better than straight LCRL.

Figure 3: *Results. (Left Column) Pacman. φ is to eventually collect food and always avoid the ghost. We let the system run for a maximum of 100 timesteps. (Right Column) Discretized Mountain Car (MC). φ is to eventually reach the flag.*

Performance in continuous state space? Similarly, the probability of satisfying the LTL spec in Figure 3 (Right) goes up to 1. However, here the LCRL (shaped) baseline performs relatively well as it is being given "breadcrumbs" for how to solve the task. Our algorithm performs well without needing any cost shaping. Standard LCRL fails to learn. This experiment demonstrates that our method can be used even in discretized continuous settings.

---

[5]The lower bound relating to β from [41] is $\Omega(\frac{\log(2\delta)}{\log(1-\beta)})$ whereas ours is $\tilde{O}(\frac{1}{\beta})$. We conjecture that $\tilde{\Omega}(\frac{1}{\beta})$ samples is required. See Appendix Section C.

8

Sample Complexity? Our theory is quite conservative w.r.t. empirical performance. In Pacman (Figure 3, Left), Thm. 5.1 suggests ≈ 350 samples per (s, a) pair just to calculate the AMECs. Empirically, LCP finds a good policy after 11 samples per (s, a) pair ($\sim 66k/6k$ samples/pair).

Other Considerations. One of the strengths and potential drawbacks of LTL is its specificity. If a φ, for a truly infinite horizon problem, is to "eventually" do something, then accomplishing the task quickly is not required. As a finite horizon problem, in MC (Fig. 3, Right) SSP finds the fastest path to the goal. In contrast, since any stochastic policy with full support will "eventually" work, the policy returned by LCP for Fig 1 (Left) (Fig. 2 Center, & App Fig. 7) may take exponential time to complete a single loop. Two straightforward ways to address this issue are: (a) including explicit time constraints in φ; and (b) cost shaping to prefer policies reaching some s^* quickly and repeatedly. Unlike standard cost-shaping, φ satisfaction is still guaranteed since the cost is decoupled from φ.

7 Related Work

Constrained Policy Optimization. One attempt at simplifying cost functions is to split the desired behaviors from the required behaviors. The desired behaviors remain as part of the cost function while the required behaviors are treated as constraints. Recent interest in constrained policy optimization within the RL community has been related to the constrained Markov Decision Process (CMDP) framework [6, 39, 2, 43]. This framework enables clean methods and guarantees, but enforces expected constraint violations rather than absolute constraint violations. Setting and interpreting constraint thresholds can be very challenging, and inappropriate in safety-critical problems [38].

LTL + RL. Recently, LTL-constrained policy optimization has been developed as an alternative to CMDPs [41]. Unlike CMPDs, the entire task is encoded into an LTL expression and is treated as the constraint. Q-learning variants when dynamics are unknown and Linear Programming methods when dynamics are known are common solution concepts [50, 26, 13, 16, 20]. The Q-learning approaches rely on proper, unknowable tuning of discount factor for their guarantees. Theoretically oriented works include [23, 64]. While providing PAC-style guarantees, the assumptions made in these works rely on unknowable policy-environment interaction properties. We make no such assumptions here.

Another solution technique is employing reward machines [60, 17, 63] or high-level specifications that can be translated into reward machines [30]. These works are generally empirical and handle finite or repeated finite problems (episodic problems at test time); they can only handle a smaller set of LTL expressions, specifically regular expressions. Our work handles ω-regular expressions, subsuming regular expressions and requires a nontrivial leap, algorithmically and theoretically, to access the broader set of allowable expressions. Many problems are ω-regular problems, but not regular, such as liveness (something good will happen eventually) and safety (nothing bad will happen forever). The works that attempt to handle full LTL expressibility redefine reward as 1 if the LTL is solved and 0 otherwise; the cost function of the MDP is entirely ignored.

Verification and Planning. As an alternative to our approach, one might consider LTL satisfaction verification and extend it to an optimization technique by checking every policy (which will naively take an exponential amount of samples to verify a single policy [15, 8]). Many verification approaches exist [36, 11, 5, 67, 37, 27] and among the ones that do not assume known dynamics, the verification guarantees rely on quantities as difficult to calculate as the original verification problem itself [8].

8 Discussion

We have presented a novel algorithm, LCP, for policy optimization under LTL constraints in an unknown environment. We formally guarantee that the policy returned by LCP simultaneously has minimal cost with respect to the MDP cost function and maximal probability of LTL satisfaction. Our experiments verify that our policies are competitive and our sample estimates conservative.

The assumptions we make are strong, but to the best of our knowledge, are the most relaxed amongst tractable model-based algorithms proposed for this space. Model-free algorithms (Q-learning) have less stringent assumptions but do not come with the kind of guarantees that our work has and largely ignore the cost function, solving only part of the problem. An interesting future direction would be to extend our work to continuous state and action spaces and settings with function approximation.

9

Acknowledgements. Cameron Voloshin is funded partly by an NSF Graduate Fellowship and a Kortschak Fellowship. This work is also supported in part by NSF #1918865, ONR #N00014-20-1-2115, and NSF #2033851.

References

[1] David Abel, Will Dabney, Anna Harutyunyan, Mark K Ho, Michael Littman, Doina Precup, and Satinder Singh. On the expressivity of markov reward. In *Advances in Neural Information Processing Systems*, 2021.

[2] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[3] Alekh Agarwal, Sham Kakade, and Lin F Yang. Model-based reinforcement learning with a generative model is minimax optimal. In *Conference on Learning Theory*, 2020.

[4] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, 2020.

[5] Gul Agha and Karl Palmskog. A survey of statistical model checking. *ACM Trans. Model. Comput. Simul.*, 28(1), jan 2018.

[6] Eitan Altman. *Constrained Markov Decision Processes: Stochastic Modeling*. Routledge, Boca Raton, 1 edition, December 2021.

[7] Rajeev Alur, Suguman Bansal, Osbert Bastani, and Kishor Jothimurugan. A framework for transforming specifications in reinforcement learning. *arXiv preprint arXiv:2111.00272*, 2021.

[8] Pranav Ashok, Jan Křetínský, and Maximilian Weininger. Pac statistical model checking for markov decision processes and stochastic games. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification*, page 497–519, Cham, 2019. Springer International Publishing.

[9] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, 2020.

[10] Kamyar Azizzadenesheli, Alessandro Lazaric, and Animashree Anandkumar. Reinforcement learning of pomdps using spectral methods. In *Conference on Learning Theory*, 2016.

[11] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. The MIT Press, Cambridge, Mass, 2008.

[12] Dimitri P Bertsekas et al. Dynamic programming and optimal control 3rd edition, volume ii. *Belmont, MA: Athena Scientific*, 2011.

[13] Alper Kamil Bozkurt, Yu Wang, Michael M. Zavlanos, and Miroslav Pajic. Control synthesis from linear temporal logic specifications using model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10349–10355, 2020.

[14] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[15] Tomáš Brázdil, Krishnendu Chatterjee, Martin Chmelík, Vojtěch Forejt, Jan Křetínský, Marta Kwiatkowska, David Parker, and Mateusz Ujma. Verification of markov decision processes using learning algorithms. In Franck Cassez and Jean-François Raskin, editors, *Automated Technology for Verification and Analysis*, page 98–114, Cham, 2014. Springer International Publishing.

[16] Mingyu Cai, Shaoping Xiao, Zhijun Li, and Zhen Kan. Optimal probabilistic motion planning with potential infeasible ltl constraints. *IEEE Transactions on Automatic Control*, pages 1–1, 2021.

[17] Alberto Camacho, Rodrigo Toro Icarte, Toryn Q. Klassen, Richard Valenzano, and Sheila A. McIlraith. Ltl and beyond: Formal languages for reward function specification in reinforcement learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6065–6073. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[18] Grace E. Cho and Carl D. Meyer. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra and its Applications*, 335(1):137–150, 2001.

[19] Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. *Advances in Neural Information Processing Systems*, 28, 2015.

10

[20] Xuchu Ding, Stephen L. Smith, Calin Belta, and Daniela Rus. Optimal control of markov decision processes with linear temporal logic constraints. *IEEE Transactions on Automatic Control*, 59(5):1244–1257, May 2014.

[21] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, 2017.

[22] Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Improved analysis of ucrl2 with empirical bernstein inequality. *arXiv preprint arXiv:2007.05456*, 2020.

[23] Jie Fu and Ufuk Topcu. Probably approximately correct MDP learning and control with temporal logic constraints. In Dieter Fox, Lydia E. Kavraki, and Hanna Kurniawati, editors, *Robotics: Science and Systems X, University of California, Berkeley, USA, July 12-16, 2014*, 2014.

[24] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91(3):325–349, 2013.

[25] Ernst Moritz Hahn, Guangyuan Li, Sven Schewe, Andrea Turrini, and Lijun Zhang. Lazy probabilistic model checking without determinisation. *arXiv preprint arXiv:1311.2928*, 2013.

[26] Mohammadhosein Hasanbeig, Alessandro Abate, and Daniel Kroening. Logically-constrained reinforcement learning, 2018.

[27] Thomas Hérault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation*, page 73–84, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[28] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

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

[30] Kishor Jothimurugan, Rajeev Alur, and Osbert Bastani. A composable specification language for reinforcement learning tasks. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Sham Kakade. On the sample complexity of reinforcement learning. *PhD thesis, Gatsby Computational Neuroscience Unit, University College London*, 2003.

[32] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer, 2002.

[33] Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi Yadkori, and Benjamin Van Roy. Conservative contextual linear bandits. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017.

[34] Andrey Kolobov, Mausam, and Daniel S. Weld. A theory of goal-oriented mdps with dead ends. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, page 438–447, Arlington, Virginia, USA, 2012. AUAI Press.

[35] Jan Křetínský, Tobias Meggendorfer, and Salomon Sickert. Owl: a library for ω-words, automata, and ltl. In *International Symposium on Automated Technology for Verification and Analysis*, pages 543–550. Springer, 2018.

[36] M. Kwiatkowska, G. Norman, and D. Parker. Prism 4.0: Verification of probabilistic real-time systems. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, page 585–591. Springer, 2011.

[37] Richard Lassaigne and Sylvain Peyronnet. Probabilistic verification and approximation. *Electron. Notes Theor. Comput. Sci.*, 143:101–114, jan 2006.

[38] Hoang M Le, Nan Jiang, Alekh Agarwal, Miro Dudík, Yisong Yue, and Hal Daumé III. Hierarchical imitation and reinforcement learning. In *International Conference on Machine Learning (ICML)*, Jul 2018.

[39] Hoang M Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning (ICML)*, 2019.

11

[40] Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. *Advances in neural information processing systems*, 33:12861–12872, 2020.

[41] Michael L. Littman, Ufuk Topcu, Jie Fu, Charles Isbell, Min Wen, and James MacGlashan. Environment-independent task specifications via gltl, 2017.

[42] Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization, 2009.

[43] Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. Reinforcement learning with convex constraints. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[44] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

[45] Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

[46] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[47] Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and q-learning. In *Conference on Learning Theory*, 2020.

[48] Guannan Qu, Chenkai Yu, Steven Low, and Adam Wierman. Exploiting linear models for model-free nonlinear control: A provably convergent policy gradient approach. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6539–6546. IEEE, 2021.

[49] Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *ICML*, 1998.

[50] Dorsa Sadigh, Eric S. Kim, Samuel Coogan, S. Shankar Sastry, and Sanjit A. Seshia. A learning based approach to control synthesis of markov decision processes for linear temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 1091–1096, 2014.

[51] E. Seneta. Perturbation of the stationary distribution measured by ergodicity coefficients. *Advances in Applied Probability*, 20(1):228–230, 1988.

[52] Salomon Sickert, Javier Esparza, Stefan Jaax, and Jan Křetínský. Limit-deterministic büchi automata for linear temporal logic. In Swarat Chaudhuri and Azadeh Farzan, editors, *Computer Aided Verification*, page 312–332, Cham, 2016. Springer International Publishing.

[53] David Silver, Satinder Singh, Doina Precup, and Richard S Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021.

[54] Jonathan Sorg. *The Optimal Reward Problem: Designing Effective Reward for Bounded Agents*. PhD thesis, University of Michigan, USA, 2011.

[55] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[56] R.S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[57] Csaba Szepesvári. Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103, 2010.

[58] Jean Tarbouriech, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Sample complexity bounds for stochastic shortest path with a generative model. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, Mar 2021.

[59] Jean Tarbouriech, Runlong Zhou, Simon Shaolei Du, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[60] Rodrigo Toro Icarte, Toryn Q. Klassen, Richard Valenzano, and Sheila A. McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *J. Artif. Int. Res.*, 73, may 2022.

12

[61] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. Is deep reinforcement learning really superhuman on atari? leveling the playing field. *arXiv preprint arXiv:1908.04683*, 2019.

[62] John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

[63] Pashootan Vaezipoor, Andrew C. Li, Rodrigo Toro Icarte, and Sheila A. McIlraith. Ltl2action: Generalizing LTL instructions for multi-task RL. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 10497–10508, 2021.

[64] Eric M. Wolff, Ufuk Topcu, and Richard M. Murray. Robust control of uncertain markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3372–3379, 2012.

[65] Yihong Wu and Pengkun Yang. Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, 47(2):857 – 883, 2019.

[66] Cambridge Yang, Michael L. Littman, and Michael Carbin. Reinforcement learning for general LTL objectives is intractable. *CoRR*, abs/2111.12679, 2021.

[67] Håkan L. S. Younes, Edmund M. Clarke, and Paolo Zuliani. Statistical verification of probabilistic properties with unbounded until. In Jim Davies, Leila Silva, and Adenilso Simao, editors, *Formal Methods: Foundations and Applications*, page 144–160, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[68] Baohe Zhang, Raghu Rajan, Luis Pineda, Nathan Lambert, André Biedenkapp, Kurtland Chua, Frank Hutter, and Roberto Calandra. On the importance of hyperparameter optimization for model-based reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 2021.

13

Contents

1	**Introduction**	**1**
2	**Motivating Examples**	**2**
3	**Background and Problem Formulation**	**2**
	3.1 MDPs with Labelled State Spaces	2
	3.2 Linear Temporal Logic (LTL), Synchronization with MDPs, and Satisfaction . . .	3
	3.3 Problem Formulation	3
4	**Approach**	**4**
	4.1 End Components & Accepting Maximal End Components	4
	4.2 High-Level Intuition	4
	4.3 Additional Assumptions and Definitions	5
	4.4 Main Algorithm: LTL Constrained Planning (LCP)	6
5	**End-To-End Guarantees**	**8**
6	**Empirical Analysis**	**8**
	6.1 Results	8
7	**Related Work**	**9**
8	**Discussion**	**9**
A	**Notation and Overview**	**16**
	A.1 Overview	17
B	**Analysis: Statements with Proof**	**19**
	B.1 Sample Complexity Guarantee	19
	B.2 High Probability Event and Sample Requirement	22
	B.3 `FindAMEC` proofs	23
	B.4 `PlanRecurrent` proofs	24
	B.5 `PlanTransient` proofs	28
C	**Conjecture on Sample Complexity**	**30**
D	**Additional Algorithms**	**31**
	D.1 Value Iteration	31
	D.2 Modified Algorithm handling Blocking Failure in Algorithm 1	32
E	**Experiments**	**34**
	E.1 Environments and Details	34
	E.2 Hyperparameters	35

14

E.3   Additonal Results   35

E.4   Policies   36

15

A Notation and Overview

Table 2: Glossary of terms

Acronym	Term
RL	Reinforcement Learning
PO	Policy Optimization
LTL	Linear Temporal Logic
MDP	Markov Decision Process
LDBA	Limit Determinisitic Buchi Automaton
AMEC/MEC/EC	Accepting MEC, Maximal EC, End Component
LCP	LTL Constrained Planning, Algo 1
FindAMEC	Subroutine to assist in finding AMECs
PlanRecurrent, PR	Subroutine to plan in AMECs, Algo 2
PlanTransient, PT	Subroutine to plan to AMECs, Algo 3
NoBlockPlanTransient, NB-PT	Subroutine to plan to AMECs, Algo 6
VI	Value Iteration Subroutine, Algo 4
AP	Atomic Proposition
Σ	Alphabet $\Sigma = 2^{AP}$
$\mathcal{S}$	State Space
$\mathcal{A}$	Action Space. $\mathcal{A}(s)$ allowable actions in state s.
$\mathcal{A}_A$	Restricted Action Space. $\mathcal{A}_A(s) \subseteq \mathcal{A}(s)$ allowable actions in state $s \in A \subseteq \mathcal{S}$.
P	Transition Function
P_π	Markov Chain induced by π in P
$\mathcal{C}$	Cost Function
$\mathcal{X}$	Product-MDP
τ	Run or trajectory in an MDP
π	Policy
η-greedy w.r.t π in $A \subseteq \mathcal{S}$	$= (1 - \eta)\pi + \eta\texttt{Unif}(\mathcal{A}_A)$ with $0 \leq \eta \leq 1$
φ	LTL Specification/Formula/Task
$\mathbb{P}[\pi \models \varphi]$	Probability that a policy satisfies the task
$\Pi, \Pi_{\max}$	Class of stochastic policies, $\pi \in \Pi$ with maximal $\mathbb{P}[\pi \models \varphi]$
β	Lower bound on minimum, nonzero transition probability
D	Dataset tracking all tuples (s, a, s') simulated
$\widehat{P}$	Empirical estimate of P from data in D
$\tilde{P}$	Optimistic dynamics returned by VI
$\mathcal{P}$	Plausible transition functions consistent with all the information gathered in D
$\mathcal{E}$	High probability event
$n(s, a)$	Number of samples accumulated in (s, a). Also denoted n
$\psi(n)$	Error bound on $\max_{s' \in \mathcal{S}} |\widehat{P}(s, a, s') - P(s, a, s')|$
$\psi^{-1}(\rho)$	Number of samples $n(s, a)$ necessary to achieve $\max_{s' \in \mathcal{S}} |\widehat{P}(s, a, s') - P(s, a, s')| < \rho$
ϵ_V	Cost-optimality tolerance wrt. main problem (1)
ϵ_φ	Prob-optimality tolerance wrt. main problem (1)
ϵ_{PR}	error input into PR, $\epsilon_{\text{PR}} = \frac{\epsilon_V}{7\lambda}$
ϵ_{PT}	error input into PT, $\epsilon_{\text{PT}} = \frac{2\epsilon_V}{9}$
$\epsilon_{\text{PR}}^{\mathcal{L}}$	Convergence condition for VI in PR, $\epsilon_{\text{PR}}^{\mathcal{L}} = \frac{2\epsilon_{\text{PR}}}{3}$
$\epsilon_{\text{PT}}^{\mathcal{L}}$	Convergence condition for VI in PT, $\epsilon_{\text{PT}}^{\mathcal{L}} = \frac{c_{\min}\epsilon_{\text{PT}}\epsilon_\varphi}{4V}$
α	Aperiodicity Coefficient $\alpha \in (0, 1)$. $P_{\alpha,\pi} = \alpha P_\pi + (1 - \alpha)I$
$\mathcal{L}_{\text{PR}}^{\alpha}$	$\mathcal{L}_{\text{PR}}^{\alpha} v(s) = \min_{a \in \mathcal{A}_A(s)} \left(\mathcal{C}(s, a) + \alpha \min_{p \in \mathcal{P}(s,a)} p^T v\right) + (1 - \alpha)v(s) \quad \forall s \in A$
$\mathcal{L}_{\text{PT}}$	$\mathcal{L}_{\text{PT}} v(s) = \begin{cases} \min\left\{\min_{a \in \mathcal{A}_A(s)} \left(\mathcal{C}(s, a) + \min_{p \in \mathcal{P}(s,a)} p^T v\right), \bar{V}\right\}, & s \in \mathcal{S} \setminus \cup_{i=1}^{k} A_i \\ \lambda g_i, & s \in A_i \end{cases}$
d_{PR}	Convergence operator for PR, $d_{\text{PR}}(v', v) = \max_{s \in A}(v'(s) - v(s)) - \min_{s \in A}(v'(s) - v(s))$
d_{PT}	Convergence operator for PT, $d_{\text{PT}}(v_{n+1}, v_n) = \|v_{n+1} - v_n\|_1$
V_T	Terminal costs. $V_T = 0$ by default
λ	Tradeoff between g_π and J_π
J_π	Transient cost, conditioned on runs satisfying φ
g_π	Gain, Average-cost, conditioned on runs satisfying φ
$\bar{V}$	Upper bound on J_π for any $\pi \in \Pi$
$\Delta(M)$	Coefficient of Ergodicity of matrix M, $\Delta(M) = \frac{1}{2}\max_{ij}\sum_k |M_{ik} - M_{jk}|$
$\bar{\Delta}_{A_i}$	Worst-case coefficient of ergodicty in A_i

16

A.1 Overview

There is a lot of notation that we will be using to get through the analysis. It is important to distinguish the following:

Table 3: Policies and Probabilities

Acronym	Term
π^*	Optimal policy w.r.t (OPT)
π	Policy returned by LCP (Algo 1)
π_{A_i}	Gain and Prob-optimal policy in AMEC $(A_i, \mathcal{A}_{A_i})$ in dynamics P
$\tilde{\pi}_{A_i}$	Gain and Prob-optimal policy in AMEC $(A_i, \mathcal{A}_{A_i})$ in dynamics $\tilde{P}$
π_i	A policy in states A_i of an AMEC $(A_i, \mathcal{A}_{A_i})$, ignoring what π does outside of A_i
p^π, p^*	$\mathbb{P}[\pi \models \varphi]$ and $\mathbb{P}[\pi^* \models \varphi]$. Also denoted p^π and p^{π^*}
$p_i^\pi, p_i^{\pi^*}$	$\mathbb{P}[\pi \text{ reaches } A_i], \mathbb{P}[\pi^* \text{ reaches } A_i] \geq 0$ denoted p_i, p_i^* (resp), $\sum_{i=1}^k p_i = p$, $\sum_{i=1}^k p_i^* = p^*$

Table 4: Gains

Term	Description. (Subscript i or A_i denotes "in AMEC $(A_i, \mathcal{A}_{A_i})$")
$\widehat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}$	Approximated gain of (greedy) optimal policy $\tilde{\pi}_{A_i}$ under optimistic dynamics $\tilde{P}$
$g_{\pi_i}^{P}$	Actual gain of policy π_i (η-greedy version of policy from PR) under true dynamics P
$g_{\pi_{A_i}}^{P}$	Gain of optimal policy π_{A_i} under dynamics P

The relationship between these gains is subtle. `PlanRecurrent` (Algo 2) returns $\widehat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}$ as the estimate for how good the best greedy policy will be in AMEC $(A_i, \mathcal{A}_{A_i})$ under dynamics $\tilde{P}$. But, we don't use the greedy policy, we use the η-greedy policy π_i. With π_{A_i} being the true gain-optimal policy in dynamics P (in AMEC $(A_i, \mathcal{A}_{A_i})$), then we will find the following relations:

$$\underbrace{\widehat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}}_{\text{Output from PR}} \underbrace{\approx}_{\frac{\epsilon_{\text{PR}}^{\mathcal{L}}}{2}[22]} g_{\tilde{\pi}_{A_i}}^{\tilde{P}} \underbrace{\approx}_{\text{Lem } B.7} g_{\pi_i}^{\tilde{P}} \underbrace{\approx}_{\text{Lem } B.8} g_{\pi_i}^{P} \underbrace{\approx}_{\text{Prop } B.5} \underbrace{g_{\pi_{A_i}}^{P}}_{\text{Actual}}$$

In general gains g are functions of state: $g(s)$. However, it is well known [46, 22] that in communicating MDPs (each state is reachable from one another by some policy) that the gain of the optimal policy (even if determinstic) is constant – independent of state. Since AMECs are communicating MDPs, then $\pi_{A_i}, \tilde{\pi}_{A_i}$ induce constant gains in $P, \tilde{P}$ respectively. Lastly, the stochastic policy π_i makes both $\tilde{P}$ and P recurrent, and so the gain is also constant. We will therefore only be considering the absolute difference between gains rather than L_∞ norms (as they coincide).

Table 5: Value Functions

Acronym	Term
V_π	Main objective, value function $V_{\pi,\lambda}^P = J_\pi + \lambda g_\pi$
v	Approximated value of policy π from PT (Algo 2) in dynamics $\tilde{P}$
$\tilde{V}_\pi^{\tilde{P}}$	Actual value of policy π from PT in dynamics $\tilde{P}$
$\tilde{V}_\pi^{P}$	Actual value of policy π from PT in dynamics P,
	also denoted $\tilde{V}_\pi = p(J_\pi + \sum_{i=1}^k \frac{p_i}{p} \widehat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}) + (1-p)\frac{\bar{V}}{\epsilon_\varphi}$

When superscripts are dropped in V, the dynamics are the true dynamics P of the product-MDP $\mathcal{X}$. Once again, the relationships between these value functions is subtle. `PlanTransient` (Algo 3) returns v as the estimate for how good π (the greedy policy wrt v) will be in reaching AMECs $\{(A_i, \mathcal{A}_{A_i})\}_{i=1}^k$, but is optimistic. v is an approximation to $\tilde{V}_\pi^{\tilde{P}}$. Roughly speaking, we will find that they are all similar/related:

$$\underbrace{v}_{\text{Ouput from PT}} \underbrace{\approx}_{\text{Lem } B.16} \tilde{V}_\pi^{\tilde{P}} \underbrace{\approx}_{\text{Lem } B.15} \tilde{V}_\pi^{P} \underbrace{\approx}_{\text{Prop } B.12/D.1} \underbrace{\tilde{V}_{\pi^*}^{P}}_{\text{An intermediate Value Func.}}$$

17

where the last approximation has 2 different propositions: the first allows the simplifying assumption made in the main paper regarding paths between AMECS, the second removes that assumption at the expense of increased computation. Finally,

$$\|\tilde{V}_\pi^P - \tilde{V}_{\pi^*}^P\| \underbrace{\approx}_{\text{Thm 5.1}} \|V_\pi^P - V_{\pi^*}^P\| \tag{5}$$

which involves swapping $\widehat{g}_{\pi_{A_i}}^{\tilde{P}}$ in $\tilde{V}$ for $g_{\pi_{A_i}}^P$.

18

B Analysis: Statements with Proof

B.1 Sample Complexity Guarantee

The number of samples necessary to guarantee an $(\epsilon_V, \epsilon_\varphi, \delta)$-PAC approximation to the cost-optimal and probability-optimal policy relies factors: β (lower bound on the mininum non-zero transition probability of P), $\{c_{\min}, c_{\max}\}$ (bounds on the cost function $\mathcal{C}$), $\bar{\Delta}_{A_i}$ (worst-case coefficient of ergodicity for EC $(A_i, \mathcal{A}_{A_i})$), V (upper bound on the value function), and λ (tradeoff factor). Recall that an event $\mathcal{E}$ captures the scenario where the empirical transition function $\widehat{P}$ is close to the true transition function P. $\mathcal{E}$ holds with probability at least $1 - \delta$, see Lem B.1.

Theorem 5.1 (Sample Complexity). *Under the event $\mathcal{E}$, Assumption 1 and 2, after*

$$n = \tilde{\mathcal{O}}\left(\frac{1}{\beta} + \frac{1}{\epsilon_V^2}\left(\frac{|\mathcal{S}|^2 \bar{V}^4}{c_{\min}^2 \epsilon_\varphi^4} + \lambda^2 \sum_{i=1}^{k} \frac{|A_i|^2 c_{\max}^2}{(1 - \bar{\Delta}_{A_i})^2} \right) \right)$$

samples[6] are collected from each state-action pair, the policy π returned by Algorithm 1 is, with probability $1 - \delta$, simultaneously ϵ_V-cost optimal and ϵ_φ-probability optimal, satisfying:

$$(i) \ \ |\mathbb{P}[\pi \models \varphi] - \mathbb{P}[\pi^* \models \varphi]| \leq \epsilon_\varphi \quad (ii) \ \ \|V_\pi - V_{\pi^*}\|_\infty < \epsilon_V. \tag{4}$$

Comparison To RL Literature. Before presenting the proofs, we briefly compare this guarantee with standard guarantees in model-based reinforcement learning under a generative model. It is important to note that while we show that our guarantee is a sum of 3 terms, a tighter bound would be a max over the 3 terms. To the best of our knowledge, the current state-of-the-art RL (with generative model) guarantee is $\tilde{\mathcal{O}}(\frac{1}{(1-\gamma)^3 \epsilon^2})$ [3], per state-action pair. Here, $H = \frac{1}{1-\gamma}$ represents the effective horizon in discounted settings. In other words, $c_{\max} H$ is the bound on (their) $\|V\|$. In our case, for the SSP reduction, the effective horizon is $H = \frac{\|\tilde{V}\|_\infty}{c_{\min}}$, as this is the expected goal-reaching time in the worst-case (since we do not have any discounting). We estimate $\|\tilde{V}\|_\infty$ with upper bound $\frac{\bar{V}}{\epsilon_\varphi}$. Suppose we set $\epsilon = \min(\epsilon_V, \epsilon_\varphi)$. Focusing just on the center term, we have guarantee taking the form, roughly, $\frac{|\mathcal{S}|^2 H^4}{\epsilon^2}$. Here, the $|\mathcal{S}|^2$ comes from a loose upper bound $\max_{s \in \mathcal{S}, a \in \mathcal{A}} \|\widehat{P}(s, a, \cdot)\|_1 = |\mathcal{S}|$. In fact, as noted in [58], when the MDP is not too chaotic $\max_{s \in \mathcal{S}, a \in \mathcal{A}} \|\widehat{P}(s, a, \cdot)\|_1 = \mathcal{O}(1)$. Further, by using careful variance-aware arguments from [58] we can decrease the dependency from H^4 to H^3. Hence, the SSP guarantee (our center term) and the standard RL guarantee are very similar. The first term $\frac{1}{\beta}$ does not appear in standard RL literature because there is no constraint verification needed, but in practice will be dominated by the other terms. The last term is also similar to the center term. $\frac{c_{\max}}{1 - \bar{\Delta}_{A_i}}$ can also be seen as an effective horizon, accumulating $c_{\max}$ cost until the accepting component sufficiently mixes. Here, $|A_i|^2 \leq |\mathcal{S}|^2$ and, again, comes from the loose upper bound $\max_{s \in A_i, a \in \mathcal{A}_{A_i}} \|\widehat{P}(s, a, \cdot)\|_1 = |A_i|$.

Proof *of Theorem 5.1.* We begin by examining the interaction of π^* with P. The Markov chain P_{π^*} has a number, say m, of recurrent classes $R_1, \ldots, R_m$, sets of states that are trapping and visited infinitely often once reached. Some of the recurrent classes R_i contain an accepting state $s \in \mathcal{S}^*$, making any trajectory entering R_i an accepting run, without loss of generality call these $R_1, \ldots, R_{m'}$ (we just relabel them). Let $\mathcal{A}_{\pi_i^*}(s) = \{a \in \mathcal{A} | \pi_i^*(s|a) > 0\}$ denote the support of actions taken by π_i^* in state $s \in R_i$. Let $\mathcal{A}_{\pi_i^*} = \{\mathcal{A}_{\pi_i^*}(s)\}_{s \in R_i}$ be the indexed action set in R_i. Then, by definition, $\{(R_i, \mathcal{A}_{\pi_i^*})\}_{i=1}^{m'}$ are accepting EC. By definition, each accepting EC $(R_i, \mathcal{A}_{\pi_i^*})$ must be contained within (or is itself) some AMEC $(A_i, \mathcal{A}_i)$.

Fix some accepting EC $(R_j, \mathcal{A}_{\pi_j^*})$. We claim, without loss of generality, $(R_j, \mathcal{A}_{\pi_j^*}) = (A_i, \mathcal{A}_{A_i})$ for some index $i \in 1, \ldots, k$. To show this, let π_{A_i} be the gain optimal, and probability-optimal policy in AMEC $(A_i, \mathcal{A}_{A_i})$: π_{A_i} is defined over all states $s \in A_i$ and actions $a \in \mathcal{A}_{A_i}$. Further, consider the modified optimal policy

$$\tilde{\pi}^*(s, a) = \begin{cases} \pi_{A_i}(s, a), & s \in A_i \\ \pi^*(s, a), & \text{otherwise.} \end{cases}$$

---

[6]The lower bound relating to β from [41] is $\Omega(\frac{\log(2\delta)}{\log(1-\beta)})$ whereas ours is $\tilde{O}(\frac{1}{\beta})$. We conjecture that $\tilde{\Omega}(\frac{1}{\beta})$ samples is required. See Appendix Section C.

19

Because π_{A_i} is prob-optimal (ie. $\mathbb{P}[\tilde{\pi}^* \models \varphi | s_0 \in A_i] = 1$) in A_i then the probability $\mathbb{P}[\tilde{\pi}^* \models \varphi] \geq \mathbb{P}[\pi^* \models \varphi]$. Further, $J_{\tilde{\pi}^*} \leq J_\pi$ because any τ that formerly passed through $R_j \setminus A_i$ now accumulates less cost. Further, $g_{\pi_{A_i}} \leq g_{\pi_i^*}$ by definition of optimality in AMEC $(A_i, \mathcal{A}_{A_i})$. Thus, $V_{\tilde{\pi}^*} \leq V_{\pi^*}$. Of course, by definition of optimality, the opposite signs hold: $V_{\tilde{\pi}^*} \geq V_{\pi^*}$ and $\mathbb{P}[\tilde{\pi}^* \models \varphi] \leq \mathbb{P}[\pi^* \models \varphi]$. Therefore $\tilde{\pi}^*$ and π^* are indistinguishable.

Repeating the above argument for each $(R_j, \mathcal{A}_{\pi_j^*})$ means the accepting EC of π^* are AMECS and, by definition, form some subset of all of the AMECs $\{(A_i, \mathcal{A}_{A_i}\}_{i=1}^k$. In other words, all accepting runs of π^* reach states $\cup_{i=1}^k A_i$. Furthermore, $g_{\pi^*} = \sum_{i=1}^k \frac{p_i^*}{p} g_{\pi_{A_i}}^P$ where $p_i^* \geq 0$ is the probability that π^* reaches A_i and $\sum_{i=1}^k p_i^* = p$.

Property (i) now follows as a direct consequence of Prop 4.3 and Prop 4.2. Recall by Prop 4.3 that $|\mathbb{P}[\pi \text{ reaches } \cup_{i=1}^k A_i] - \max_{\pi' \in \Pi_{\max}} \mathbb{P}[\pi' \text{ reaches } \cup_{i=1}^k A_i]| \leq \epsilon_\varphi$. Prop 4.2 implies that once a run enters some A_i, the run is accepted. Remaining runs cannot be accepted since they do not reach any AMEC, the only way to be accepted. Hence $\mathbb{P}[\pi \models \varphi] = \mathbb{P}[\pi \text{ reaches } \cup_{i=1}^k A_i]$. Since we just showed that all accepting runs of π^* reach some $(A_i, \mathcal{A}_i)$ then:

$$0 \leq \mathbb{P}[\pi^* \models \varphi] - \mathbb{P}[\pi \models \varphi] \leq \mathbb{P}[\pi^* \text{ reaches } \cup_{i=1}^k A_i] - \mathbb{P}[\pi \text{ reaches } \cup_{i=1}^k A_i] \leq \epsilon_\varphi.$$

To show **Property** (ii), first let us define p_i^π as the probability of π reaching AMEC $(A_i, \mathcal{A}_{A_i})$ and, by property (i), $\sum_{i=1}^k p_i^\pi = \sum_{i=1}^k p_i^* = p$. The value function given by the Bellman operator $\mathcal{L}_{\text{PT}}$ (Table 1) in Algorithm 3 takes the form

$$\tilde{V}_\pi(s) = p(J_\pi(s) + \lambda \sum_{i=1}^k \frac{p_i^\pi}{p} \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}} + (1-p)\frac{\bar{V}}{\epsilon_\varphi} \tag{6}$$

where $\hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}$ are the approximated gains for end component $(A_i, \mathcal{A}_{A_i})$ from Algorithm 2. To see this, there is probability p that $\pi \models \varphi$ and achieves (conditional) expected cost $J_\pi(s) + \lambda \sum_{i=1}^k \frac{p_i}{p} \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}$ and prob $1-p$ that $\pi \not\models \varphi$ where all cooresponding trajectories get stuck and accumulate $\frac{\bar{V}}{\epsilon_\varphi}$ cost. Let $\tilde{\pi}$ now represent the optimal solution to the value function $\tilde{V}_\pi$ (Algo 3). Therefore we claim:

$$0 \leq V_\pi - V_{\pi^*} = V_\pi - \tilde{V}_\pi + \tilde{V}_\pi - \tilde{V}_{\tilde{\pi}} + \tilde{V}_{\tilde{\pi}} - \tilde{V}_{\pi^*} + \tilde{V}_{\pi^*} - V_{\pi^*} + (1-p)\frac{\bar{V}}{\epsilon_\varphi} - (1-p)\frac{\bar{V}}{\epsilon_\varphi}$$

$$\leq \underbrace{|V_\pi - \tilde{V}_\pi + (1-p)\frac{\bar{V}}{\epsilon_\varphi}|}_{(a)} + \underbrace{|\tilde{V}_\pi - \tilde{V}_{\tilde{\pi}}|}_{(b)} + \underbrace{|\tilde{V}_{\tilde{\pi}} - \tilde{V}_{\pi^*}|}_{(c)} + \underbrace{|\tilde{V}_{\pi^*} - V_{\pi^*} - (1-p)\frac{\bar{V}}{\epsilon_\varphi}|}_{(d)}$$

$$\leq \frac{\epsilon_V}{3} + \frac{\epsilon_V}{3} + 0 + \frac{\epsilon_V}{3} \leq \epsilon_V$$

For (a), first we note that $g_\pi = \sum_{i=1}^k \frac{p_i^\pi}{p} g_{\pi_i}^P$, by definition of conditional expectation. Let $\epsilon_{\text{PR}} = \frac{\epsilon_V}{7\lambda}$. Hence,

$$\underbrace{|V_\pi - \tilde{V}_\pi + (1-p)\frac{\bar{V}}{\epsilon_\varphi}|}_{(a)} = |p(J_\pi(s) + \lambda \sum_{i=1}^k \frac{p_i^\pi}{p} g_{\pi_i}^P) - p(J_\pi(s) + \lambda \sum_{i=1}^k \frac{p_i^\pi}{p} \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}})|$$

$$\leq \lambda \max_{i=1,\ldots,k} |g_{\pi_i}^P - \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}|$$

$$\leq \lambda \frac{4\epsilon_{\text{PR}}}{3}, \quad \text{Corollary } B.6$$

$$\leq \frac{\epsilon_V}{3}$$

20

By similar argument, for (d), together with earlier argument that $g_\pi^* = \sum_{i=1}^k \frac{p_i^*}{p} g_{\pi_{A_i}}^P$ then we also have that:

$$\underbrace{|\tilde{V}_{\pi^*} - V_{\pi^*} - (1-p)\frac{\bar{V}}{\epsilon_\varphi}|}_{(d)} = |p(J_{\pi^*}(s) + \lambda \sum_{i=1}^k \frac{p_i^*}{p} \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}) - p(J_{\pi^*}(s) + \lambda \sum_{i=1}^k \frac{p_i^\pi}{p} g_{\pi_{A_i}}^P)|$$

$$\leq \lambda \max_{i=1,\dots,k} |g_{\pi_{A_i}}^P - \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}|$$

$$\leq \lambda \max_{i=1,\dots,k} |g_{\pi_{A_i}}^P - g_{\pi_i}^P| + |g_{\pi_i}^P - \hat{g}_{\tilde{\pi}_{A_i}}^{\tilde{P}}|$$

$$\leq \lambda \frac{7\epsilon_{\text{PR}}}{3}, \quad \text{Prop } B.5 \text{ and Corollary } B.6$$

$$\leq \frac{\epsilon_V}{3}$$

Further, we have $(c) \leq 0$ holds because $\tilde{\pi}$ is optimal in $\tilde{V}$ (either by assuming $\cup_{i=1}^k A_i$ is the correct choice of AMECS, or using Algo 6 instead of `planTransient`). In either case, $(b) \leq \frac{3\epsilon_{\text{PT}}}{2} \leq \frac{\epsilon_V}{3}$ by Prop B.12 or Prop D.1, where ϵ_{PT} is set to $\epsilon_{\text{PT}} = \frac{2\epsilon_V}{9}$, completing the approximation guarantee.

We now compute the number of samples, per state-action pair, required by Algorithm 1. By Prop B.4, we need $n = \tilde{\mathcal{O}}(\frac{1}{\tilde{\beta}})$ to verify the support of P. After calculating the AMECs $\{(A_i, \mathcal{A}_{A_i})\}_{i=1}^k$, we calculate the gain-optimal policy π_i for each AMEC. By Prop 4.2, we need $n = \tilde{\mathcal{O}}((\frac{|A_i| c_{\max}}{\epsilon_{\text{PR}}(1-\bar{\Delta}_{A_i})})^2) = \tilde{\mathcal{O}}((\frac{\lambda |A_i| c_{\max}}{\epsilon_V(1-\bar{\Delta}_{A_i})})^2)$ for each state-action pair in each end component $(A_i, \mathcal{A}_{A_i})$, since $\epsilon_{\text{PR}} = \frac{\epsilon_V}{7\lambda}$. Finally, for the transient policy π_0, the SSP reduction requires $n = \tilde{\mathcal{O}}((\frac{|\mathcal{S}\setminus\cup_{i=1}^k A_i|\bar{V}^2}{\epsilon_{\text{PT}}\epsilon_\varphi^2 c_{\min}})^2) = \tilde{\mathcal{O}}((\frac{|\mathcal{S}\setminus\cup_{i=1}^k A_i|\bar{V}^2}{\epsilon_V\epsilon_\varphi^2 c_{\min}})^2)$ for each state-action pair outside of the AMECs, by Prop 4.3, since $\epsilon_{\text{PT}} = \frac{2\epsilon_V}{9}$. A similar sample complexity is guaranteed by using Algo 6 in place of `PlanTransient`, where $n = \tilde{\mathcal{O}}((\frac{|\mathcal{S}|\bar{V}^2}{\epsilon_V\epsilon_\varphi^2 c_{\min}})^2)$ is required in place of $n = \tilde{\mathcal{O}}((\frac{|\mathcal{S}\setminus\cup_{i=1}^k A_i|\bar{V}^2}{\epsilon_V\epsilon_\varphi^2 c_{\min}})^2)$. Adding these together, yields the worst-case number of samples necessary in any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$. These sample guarantees hold only when the event $\mathcal{E}$ holds, which itself holds with probability $1 - \delta$ (see Lem B.1).

$\square$

Corollary 5.2 (Gain (Average Cost) Optimality). *There exists $\lambda^* > 0$ s.t. for $\lambda > \lambda^*$, the policy π returned by Alg. 1 satisfies* (4), $g_\pi = \arg\min_{\pi' \in \Pi_{\max}} g_{\pi'}$, *and is probability and gain optimal.*

Proof of Corollary 5.2. Fix some $\lambda > 0$. Let $\pi' = \arg\min_{\pi \in \Pi_{\max}} g_\pi$. Suppose $g_{\pi'} < g_\pi$ but $V_{\pi,\lambda} < V_{\pi',\lambda}$, elementwise. In other words, π is the preferred policy. Then,

$$0 \leq V_{\pi',\lambda} - V_{\pi,\lambda}$$
$$= J_{\pi'} + \lambda g_{\pi'} - J_\pi - \lambda g_\pi$$
$$\leq \max_{\tilde{\pi} \in \Pi} J_{\tilde{\pi}} + \lambda \underbrace{(g_{\pi'} - g_\pi)}_{<0}$$

since $J_{\tilde{\pi}} \geq 0$ for each $\tilde{\pi} \in \Pi$. If $\lambda > \frac{\max_{\tilde{\pi} \in \Pi} J_{\tilde{\pi}}}{g_\pi - g_{\pi'}}$ then we contradict $V_{\pi,\lambda} < V_{\pi',\lambda}$. In particular, if π' is the gain optimal policy then for any $\lambda > \lambda^* = \frac{\max_{\tilde{\pi} \in \Pi} J_{\tilde{\pi}}}{\min_{\{\pi \in \Pi | g_\pi \neq g_{\pi'}\}} g_\pi - g_{\pi'}}$ then π' is preferred to any other policy $\pi \in \Pi$.

$\square$

21

B.2 High Probability Event and Sample Requirement

Definition 4.2 (High Probability Event). A high probability event $\mathcal{E}$:

$$\mathcal{E} = \{\forall s, a, s' \in S \times A \times S, \forall n(s,a) > 1 : |(P(s,a,s') - \widehat{P}(s,a,s'))| \leq \psi_{sas'}(n) \leq \psi(n)\},$$

where $\psi_{sas'}(n) \equiv \sqrt{2\widehat{P}(s,a,s')(1-\widehat{P}(s,a,s')))\xi(n)} + \frac{7}{3}\xi(n)$, $\psi(n) \equiv \sqrt{\frac{1}{2}\xi(n)} + \frac{7}{3}\xi(n)$, and $\xi(n) \equiv \log(\frac{4n^2|\mathcal{S}|^2|\mathcal{A}|}{\delta})/(n-1)$.

Lemma B.1 (High Probability Event holds). *The event $\mathcal{E}$ holds with probability at least $1 - \delta$.*

Proof of Lemma B.1 . We start with the anytime version of Theorem 4 of [42] given by Lemma 27 of [59]:

$$\mathbb{P}\left[\forall\, n \geq 1, \left|\mathbb{E}[Z] - \frac{1}{n}\sum_{i=1}^{n} Z_i\right| > \sqrt{\frac{2\widehat{V}_n \log(4n^2/\delta)}{n-1}} + \frac{7\log(4n^2/\delta)}{3(n-1)}\right] \leq \delta,$$

for any $Z_i \in [0,1]$ iid. By re-setting $\delta \leftarrow \frac{\delta}{|\mathcal{S}|^2|\mathcal{A}|}$, applying union bound over all $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, and observing that $Z_i \sim P(s,a,s')$ is a Bernoulli random variable with empirical variance $\widehat{V}_n = \widehat{P}(s,a,s')(1-\widehat{P}(s,a,s'))$ yields the result:

$\{\forall s, a, s' \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}, \forall n > 1 : \quad |P(s,a,s') - \widehat{P}(s,a,s')| \leq \psi_{sas'}(n)\}$ holds with prob $1 - \delta$ Observing that $\psi_{sas'}(n) \leq \psi(n)$ for all $n > 1$ because $\psi_{sas'}(n)$ takes on a maximum when $\widehat{P}(s,a,s') = \frac{1}{2}$, completes the proof. $\qquad\square$

Lemma B.2 (Inverting $\mathcal{E}$). *Fix $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Under the event $\mathcal{E}$, the number of samples $\psi^{-1}(\rho)$ required to achieve $|P(s,a,s') - \widehat{P}(s,a,s')| \leq \psi_{sas'}(n) \leq \psi(n) < \rho$ is given by:*

$$\psi^{-1}(\rho) = \lceil\frac{2}{\zeta^2}\log(\frac{16|\mathcal{S}|^2|\mathcal{A}|}{\zeta^4\delta})\rceil + 3 = \tilde{\mathcal{O}}(\frac{1}{\rho^2}),$$

where $\zeta \equiv \frac{-\frac{3}{7\sqrt{2}} + \sqrt{(\frac{3}{7\sqrt{2}})^2 + \frac{12}{7}\rho}}{2}$.

Proof of B.2. We have $\psi_{sas'} < \psi(n) = \frac{x}{\sqrt{2}} + \frac{7}{3}x^2 \leq \rho$ where $x^2 = \xi(n) = \frac{\log(4n^2|\mathcal{S}|^2|\mathcal{A}|\delta^{-1})}{n-1}$. Solving the quadratic inequality, we have

$$x \leq \frac{-\frac{3}{7\sqrt{2}} + \sqrt{(\frac{3}{7\sqrt{2}})^2 + \frac{12}{7}\rho}}{2} \equiv \zeta$$

Hence, we have

$$\frac{\log(4n^2|\mathcal{S}|^2|\mathcal{A}|\delta^{-1})}{n-1} \leq \zeta^2$$

$$\implies n \geq \frac{\log(4n^2|\mathcal{S}|^2|\mathcal{A}|\delta^{-1})}{\zeta^2} + 1$$

$$= \frac{1}{\zeta^2}\log(e^{\zeta^2}4n^2|\mathcal{S}|^2|\mathcal{A}|\delta^{-1})$$

$$= \frac{2}{\zeta^2}\log(\underbrace{e^{\frac{\zeta^2}{2}}\sqrt{4|\mathcal{S}|^2|\mathcal{A}|\delta^{-1}}}_{c_2}\, n) \quad (\star)$$

$$\underbrace{\phantom{\frac{2}{\zeta^2}}}_{c_1}$$

By Lemma B.3, if $n > 2c_1\log(c_1 c_2)$ then $n > (\star)$. Simplifying,

$$n \geq \frac{2}{\zeta^2}\log(\frac{16|\mathcal{S}|^2|\mathcal{A}|}{\zeta^4\delta}) + 2$$

Selecting $\psi^{-1}(\rho) = \lceil\frac{2}{\zeta^2}\log(\frac{16|\mathcal{S}|^2|\mathcal{A}|}{\zeta^4\delta})\rceil + 3$ and noting that $\zeta = \tilde{\mathcal{O}}(\rho)$ completes the proof: $n = \tilde{\mathcal{O}}(1/\rho^2)$. $\qquad\square$

Lemma B.3. *(Lemma 10 of [33]) If $\log(c_1 c_2) \geq 1$ and $c_1, c_2 > 0$ then*
$$N > 2c_1\log(c_1 c_2) \implies N > c_1\log(c_2 N)$$

22

B.3 `FindAMEC` **proofs**

Proposition B.4. *(Support Verification* `FindAMEC`*) Under the event $\mathcal{E}$ and Assumption 1, if $n = \phi_{\texttt{FindAMEC}}(\beta) = \frac{5}{\beta}\log(\frac{100|\mathcal{S}|^2|\mathcal{A}|}{\beta^2\delta}) = \tilde{\mathcal{O}}(\frac{1}{\beta})$ samples are collected for each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ then the support of P is verified:*

$$P(s,a,s') = \begin{cases} 0, & \widehat{P}(s,a,s') = 0 \\ 1, & \widehat{P}(s,a,s') = 1 \\ \in [\beta, 1-\beta], & \textit{otherwise} \end{cases}$$

Proof *of Prop B.4* . Fix $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Suppose $\widehat{P}(s,a,s') \in \{0,1\}$ then by $\mathcal{E}$ we have

$$\frac{7\log(4n^2|\mathcal{S}|^2|\mathcal{A}|/\delta)}{3(n-1)} \leq \beta \tag{7}$$

Following the second half of the proof of B.2 with $\zeta^2 = \frac{3\beta}{7}$, we have that if we take $n = \phi_{\texttt{FindAMEC}}(\beta) = \frac{5}{\beta}\log(\frac{100|\mathcal{S}^2\mathcal{A}|}{\beta^2\delta}) > \frac{14}{3\beta}\log(\frac{784|\mathcal{S}^2\mathcal{A}|}{9\beta^2\delta})$ then we have

$$|P(s,a,s') - \widehat{P}(s,a,s')| < \beta \tag{8}$$

Case $\widehat{P}(s,a,s') = 1$. Suppose $\widehat{P}(s,a,s') = 1$. By Eq (8), $P(s,a,s') > 1 - \beta$. By Assumption 1 together with the fact that $\sum_{x \in \mathcal{S}} P(s,a,x) = 1$ then $P(s,a,x) = 0$ for any $x \neq s'$. Therefore, $P(s,a,s') = \widehat{P}(s,a,s') = 1$.

Case $\widehat{P}(s,a,s') = 0$. Suppose $\widehat{P}(s,a,s') = 0$. By Eq (8), $P(s,a,s') < \beta$. Hence $P(s,a,s') = \widehat{P}(s,a,s') = 0$, otherwise violating Assumption 1.

Case, Otherwise. If $P(s,a,s') > 1 - \beta$ or $P(s,a,s') < \beta$ then by following the above arguments we'd yield similar contradictions with Assumption 1. Hence, $P(s,a,s') \in [\beta, 1-\beta]$ $\qquad\square$

23

B.4 `PlanRecurrent` proofs

Proposition 4.2 (PR Convergence & Correctness, Informal). *Let π_A be the gain-optimal policy in AMEC $(A, \mathcal{A})$. Algorithm 2 terminates after at most $\log_2\left(\frac{6|A|c_{\max}}{\epsilon_{PR}(1-\tilde{\Delta}_A)}\right)$ repeats, and collects at most $n = \tilde{\mathcal{O}}(\frac{|A|^2 c_{\max}^2}{\epsilon_{PR}^2(1-\tilde{\Delta}_A)^2})$ samples for each $(s,a) \in (A, \mathcal{A}_A)$. The η-greedy policy π w.r.t. v' (Alg. 2, Line 5) is gain optimal and probability optimal: $|g_\pi - g_{\pi_A}| < \epsilon_{PR}$, $\mathbb{P}[\pi \models \varphi | s_0 \in A] = 1$.*

We formalize Prop 4.2 as follows by adding the necessary PAC statements:

Proposition B.5 (PR Convergence & Correctness, Formal). *Let π_A be the gain-optimal policy in AMEC $(A, \mathcal{A})$. Algorithm 2 terminates after at most $\log_2\left(\frac{6|A|c_{\max}}{\epsilon_{PR}(1-\tilde{\Delta}_A)}\right)$ repeats, and collects at most $n = \tilde{\mathcal{O}}(\frac{|A|^2 c_{\max}^2}{\epsilon_{PR}^2(1-\tilde{\Delta}_A)^2})$ samples for each $(s,a) \in (A, \mathcal{A}_A)$. Under the event $\mathcal{E}$ and Assumption 1 then with probability $1 - \delta$, the η-greedy policy π w.r.t. v' (Alg. 2, Line 5) is gain optimal and probability optimal: $|g_\pi - g_{\pi_A}| < \epsilon_{PR}$, $\mathbb{P}[\pi \models \varphi | s_0 \in A] = 1$.*

Proof of Prop 4.2 & Prop B.5. Let $\pi_{v'}$ be the greedy policy with respect to v'. Let $g_{\tilde{\pi}_A}^{\tilde{P}}$ be the gain of the gain-optimal policy, $\tilde{\pi}_A$, in A with respect to dynamics $\tilde{P}$.

For the approximation error,

$$
\begin{aligned}
0 \le g_\pi^P - g_{\pi_A}^P &= g_\pi^P - g_\pi^{\tilde{P}} + g_\pi^{\tilde{P}} - g_{\pi_{v'}}^{\tilde{P}} + g_{\pi_{v'}}^{\tilde{P}} - g_*^{\tilde{P}} + g_*^{\tilde{P}} - g_{\pi_A}^P \\
&\le \underbrace{|g_\pi^P - g_\pi^{\tilde{P}}|}_{(a)} + \underbrace{|g_\pi^{\tilde{P}} - g_{\pi_{v'}}^{\tilde{P}}|}_{(b)} + \underbrace{|g_{\pi_{v'}}^{\tilde{P}} - g_{\tilde{\pi}_A}^{\tilde{P}}|}_{(c)} + \underbrace{g_{\tilde{\pi}_A}^{\tilde{P}} - g_{\pi_A}^P}_{(d)} \\
&< \frac{\epsilon_{PR}}{3} + \frac{\epsilon_{PR}}{3} + \frac{\epsilon_{PR}}{3} + 0 = \epsilon_{PR}
\end{aligned}
$$

We have the first inequality because π_A is gain optimal in P. By the Simulation Lemma B.8 we have that $(a) < \frac{\epsilon_{PR}}{3}$ by setting $\epsilon_{(2)} = \frac{\epsilon_{PR}}{3}$ in the Lemma. By the η-greedy approximation Lemma B.7 we have $(b) < \frac{\epsilon_{PR}}{3}$ by setting $\epsilon_{(1)} = \frac{\epsilon_{PR}}{3}$ in the Lemma. For (c), since $\pi_{v'}$ represents the approximately optimal policy in $\tilde{P}$ then, by value iteration approximation guarantees, $(c) = |g_{\pi_{v'}}^{\tilde{P}} - g_{\tilde{\pi}_A}^{\tilde{P}}| < \frac{\epsilon_{PR}^{\mathcal{L}}}{2} \le \frac{\epsilon_{PR}}{3}$ by setting $\epsilon_{PR}^{\mathcal{L}} = \frac{2\epsilon_{PR}}{3}$ [22]. It is known that, by optimism and the aperiodicity transformation [22, 29] for the average cost Bellman operator, $g_{\tilde{\pi}_A}^{\tilde{P}} < g_{\pi_A}^P$ implying $(d) < 0$.

For the probability of satisfaction, when $s_0 \in A$, following a policy that samples every action in $\mathcal{A}_A$ with positive probability makes the markov chain P_π recurrent. Thus, each $s \in A$ is visited infinitely often. In particular there is some $s^* \in A$ visited infinitely often, implying $\pi \models \varphi$.

Convergence is guaranteed by Lemma B.8: since ρ is halved every iteration then ρ never falls below $\frac{\epsilon_{(2)}(1-\tilde{\Delta}_A)}{2|A|c_{\max}}$, which is reached after $\log_{\frac{1}{2}}(\frac{\epsilon_{PR}(1-\tilde{\Delta}_A)}{6|A|c_{\max}}) = \log_2(\frac{6|A|c_{\max}}{\epsilon_{PR}(1-\tilde{\Delta}_A)})$ iterations (since $\epsilon_{(2)} = \frac{\epsilon_{PR}}{3}$). Further by Lemma B.8, we get the sample complexity $n = \tilde{\mathcal{O}}(\frac{|A|^2 c_{\max}^2}{\epsilon_{PR}^2(1-\tilde{\Delta}_A)^2})$, completing the proof. $\square$

Corollary B.6. *Under the same assumptions as Prop B.5, in addition, $|g_\pi^P - \hat{g}_{\tilde{\pi}_A}^{\tilde{P}}| \le \frac{4\epsilon_{PR}}{3}$.*

Proof. Continuing the same argument as in Prop B.5, we have

$$
0 \le g_\pi^P - g_{\tilde{\pi}_A}^{\tilde{P}} + g_{\tilde{\pi}_A}^{\tilde{P}} - \hat{g}_{\tilde{\pi}_A}^{\tilde{P}} \le \epsilon_{PR} + \frac{\epsilon_{PR}^{\mathcal{L}}}{2} = \frac{4\epsilon_{PR}}{3}
$$

where we use triangle inequality and appeal to Prop B.5 for $|g_\pi^P - g_{\tilde{\pi}_A}^{\tilde{P}}| \le \epsilon_{PR}$ and [22] where $|g_{\tilde{\pi}_A}^{\tilde{P}} - \hat{g}_{\tilde{\pi}_A}^{\tilde{P}}| \le \frac{\epsilon_{PR}^{\mathcal{L}}}{2} \le \frac{\epsilon_{PR}}{3}$ since $\epsilon_{PR}^{\mathcal{L}} = \frac{2\epsilon_{PR}}{3}$. $\square$

Lemma B.7. *(η-greedy approximation) Let P be any dynamics. Let π be a greedy policy in AMEC $(A, \mathcal{A}_A)$ with dynamics P. With $0 \le \eta \le 1$, let π_η be η-greedy with respect to π. Then, for any error $\epsilon_{(1)} > 0$, there exists some threshold $\eta^* \in (0, 1]$ such that when $\eta \in (0, \eta^*]$ we have*

$$
|g_\pi^P - g_{\pi_\eta}^P| \le \epsilon_{(1)} \tag{9}
$$

24

Proof. Let $s_0, \ldots, s_{|A|-1}$ be any ordering of the states in A. The standard (non-optimistic) average cost Bellman equation with known dynamics P is given by $\mathcal{L}v(s) = g(s) + \min_{a \in \mathcal{A}_A(s)} \big(\mathcal{C}(s,a) + P(s,a)v \big)$ for each $s \in A$ for a unique g and v unique up to a constant translation [12]. Furthermore, since the end components are communicating sets then we know that g is a constant vector, i.e. $g = g(s) = g(s')$ for any $s, s' \in A$ [12]. Since v is unique up to translation, we can always set $v(0) = 0$ to make v unique. The evaluation equations, under policy π, is similarly, $\mathcal{L}_\pi v(s) = g_\pi + \mathbb{E}_{a \sim \pi}\big[\mathcal{C}(s,a)\big] + P_\pi(s,a)v$ [12]. For more generality, instead of P_π we consider $\alpha P_\pi + (1-\alpha)I$, an aperiodicity transform with any coefficient $\alpha \in [0,1]$. Then the $\mathcal{L}_\pi$ written as a system takes the form:

$$
0_{2|A|} = \underbrace{\left[\begin{array}{c|c} \alpha P_\pi - (1-\alpha)I & -I \\ \hline C & D \end{array}\right]}_{X_\pi} \underbrace{\begin{bmatrix} v_0 \\ \vdots \\ v_{|A|-1} \\ g_0 \\ \vdots \\ g_{|A|-1} \end{bmatrix}}_{y} - \underbrace{\begin{bmatrix} \mathbb{E}_{a \sim \pi}\mathcal{C}(s_0, a) \\ \vdots \\ \mathbb{E}_{a \sim \pi}\mathcal{C}(s_{|A|-1}, a) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{b_\pi}
$$

with

$$
C = \begin{bmatrix} 1 & 0 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \ddots & \end{bmatrix}, D = \begin{bmatrix} 0 & \cdots & & 0 \\ 1 & -1 & & 0 \\ & \ddots & \ddots & \\ 0 & & 1 & -1 \end{bmatrix}
$$

This system combines $\mathcal{L}v(s) = \mathbb{E}_{a \sim \pi(s)}[\mathcal{C}(s,a)] + (\alpha P_\pi + (1-\alpha)I)v$ together with $g(s) = g(s')$ for any $s, s' \in A$ and $v(0) = 0$.

Succinctly, $X_\pi y - b_\pi = 0$. Similarly, we have $X_{\pi_\eta} y' - b_{\pi_\eta} = 0$. Let $dX = X_{\pi_\eta} - X_\pi$, $db = b_{\pi_\eta} - b_\pi$ and $dy = y' - y$ then $(X_\pi + dX)(y + dy) - (b_\pi + db) = 0$. Hence,

$$
dy = (X_\pi + dX)^{-1}(db - dXy)
$$
$$
= (I + X_\pi^{-1}dX)^{-1}X_\pi^{-1}(db - dXy)
$$

We calculate $\|dX\|_\infty$:

$$
\|dX\|_\infty = \max_{s \in A} \sum_{s' \in A} |\alpha P_{\pi_\eta}(s,s') - \alpha P_\pi(s,s')|
$$
$$
= \max_{s \in A} \sum_{s' \in A} |\alpha((1-\eta)P_\pi(s,s') + \eta P_{Unif}(s,s')) - \alpha P_\pi(s,s')|
$$
$$
= \alpha\eta \max_{s \in A} \sum_{s' \in A} |P_{Unif}(s,s') - P_\pi(s,s')|
$$
$$
\leq \alpha\eta 2|A|
$$

By a similar argument, together with $\mathcal{C} \leq c_{\max}$, then $\|db\|_\infty \leq 2\eta c_{\max}$ Hence,

$$
\|dy\|_\infty \leq \|(I + X_\pi^{-1}dX)^{-1}\|_\infty \|X_\pi^{-1}\|_\infty (\|db\|_\infty + \|dX\|_\infty \|y\|_\infty)
$$
$$
\leq \frac{\|X_\pi^{-1}\|_\infty}{1 - \|X_\pi^{-1}\|_\infty \|dX\|_\infty}(\|db\|_\infty + \|dX\|_\infty \|y\|_\infty)
$$
$$
\leq \frac{\eta\|X_\pi^{-1}\|_\infty}{1 - 2\alpha|A|\eta\|X_\pi^{-1}\|_\infty}(2c_{\max} + 2\alpha|A|\|y\|_\infty)
$$

By selecting

$$
\eta \leq \eta^* = \frac{\epsilon_{(1)}}{\|X_\pi^{-1}\|_\infty(2c_{\max} + 2\alpha|A|\|y\|_\infty) + \epsilon_{(1)}2\alpha|A|\|X_\pi^{-1}\|_\infty}
$$

we get that $\|dy\|_\infty \leq \epsilon_{(1)}$ and therefore $|g_\pi^P - g_{\pi_\eta}^P| \leq \epsilon_{(1)}$, as desired. $\qquad\square$

25

Lemma B.8. *(Simulation Lemma, Avg. Cost) Fix some $\alpha \in (0, 1)$ arbitrary. Let $\tilde{P}$ be the optimistic dynamics achieving the inner minimum of the Bellman equation with respect to $\mathcal{L}_{PR}^{\alpha}$ (see Table 1) in the AMEC given by $(A, \mathcal{A}_A)$. Let π be the η^* stochastic policy as in Lemma B.7. For some error $\epsilon_{(2)} > 0$. Let $m \in \mathbb{N}$ be the smallest value such that $\Delta((\alpha \tilde{P}_\pi + (1 - \alpha)I)^m) < 1$. When n is large enough that $\psi(n) \leq \frac{1}{\alpha^2} \left(\left(\frac{\epsilon_{(2)}(1 - \Delta(\tilde{P}_{\alpha,\pi}^m))}{|A|c_{\max}} + 1 \right)^{1/m} - 1 \right)$ then*

$$|g_\pi^P - g_\pi^{\tilde{P}}| < \epsilon_{(2)}. \tag{10}$$

Let $m = \max_{\pi \in \Pi_A} \min_{m \in \mathbb{N}} \{m | \Delta((\alpha P_{\pi_\eta^*} + (1 - \alpha)I)^m) < 1\}$ and $\bar{\Delta}_A = \max_{\pi \in \Pi_A} \Delta((\alpha P_{\pi_\eta^*} + (1 - \alpha)I)^m)$ for Π_A, the set of deterministic policies in A. Then, in particular, (10) holds after $n = \tilde{\mathcal{O}}(\frac{|A|^2 c_{\max}^2}{\epsilon_{(2)}^2 (1 - \bar{\Delta}_A)^2})$ samples are collected for each state-action pair in $(A, \mathcal{A}_A)$.

Proof. Consider, notationally, $P_\alpha(s, a, s') = \alpha P(s, a, s') + (1 - \alpha)\mathbf{1}_{\{s=s'\}}$ be an aperiodicity transform with $\alpha \in (0, 1)$. When fixed by a policy, then $P_{\alpha,\pi} = \alpha P_\pi + (1 - \alpha)I$. By [46] (Prop. 8.5.8), aperiodicity transforms do not affect gain. Hence $g_\pi^P = g_\pi^{P_\alpha}$ and $g_\pi^{\tilde{P}} = g_\pi^{\tilde{P}_\alpha}$. Let x_{π, P_α} be the stationary distribution of π in P_α and $x_{\pi, \tilde{P}_\alpha}$ be the stationary distribution of π in $\tilde{P}_\alpha$. These quantities exist due to the fact that π has full support over $\mathcal{A}_A$ making both $P_\alpha, \tilde{P}_\alpha$ ergodic (finite, irreducible, recurrent, and aperiodic). Hence,

$$
\begin{aligned}
|g_\pi^P - g_\pi^{\tilde{P}}| &= |g_\pi^{P_\alpha} - g_\pi^{\tilde{P}_\alpha}| \\
&= |\mathbb{E}_{s \sim x_{\pi, P_\alpha}}[\mathbb{E}_{a \sim \pi(s)}[\mathcal{C}(s, a)]] - \mathbb{E}_{s \sim x_{\pi, \tilde{P}_\alpha}}[\mathbb{E}_{a \sim \pi(s)}[\mathcal{C}(s, a)]]| \\
&= |\sum_{s \in A} \mathbb{E}_{a \sim \pi(s)}[\mathcal{C}(s, a)](x_{\pi, P_\alpha}(s) - x_{\pi, \tilde{P}_\alpha}(s))| \\
&\leq c_{\max} \|x_{\pi, P_\alpha} - x_{\pi, \tilde{P}_\alpha}\|_1
\end{aligned}
$$

To bound $\|x_{\pi, P_\alpha} - x_{\pi, \tilde{P}_\alpha}\|_1$, we appeal to classic stationary-distribution perturbation bounds [18]. First, since $\tilde{P}_{\alpha,\pi}$ is ergodic then $\exists m_0 < \infty$ such that for any $m \geq m_0$ then $\Delta(\tilde{P}_{\alpha,\pi}^m) < 1$. Then, in particular, $\|x_{\pi, P_\alpha} - x_{\pi, \tilde{P}_\alpha}\|_1 \leq \frac{\|\tilde{P}_{\alpha,\pi}^m - P_{\alpha,\pi}^m\|_\infty}{1 - \Delta(\tilde{P}_{\alpha,\pi}^m)}$ [51, 18]. Let $E = P_{\pi,\alpha} - \tilde{P}_{\pi,\alpha}$, and thus $\|E\|_\infty = \alpha \|P_\pi - \tilde{P}_\pi\|_\infty \leq \alpha |A|\psi(n)$. Then,

$$
\begin{aligned}
\|\tilde{P}_{\alpha,\pi}^m - P_{\alpha,\pi}^m\|_\infty &= \|\tilde{P}_{\alpha,\pi}^m - (\alpha P_\pi + (1 - \alpha)I)^m\|_\infty \\
&= \|\tilde{P}_{\alpha,\pi}^m - (\alpha E + \alpha \tilde{P}_\pi + (1 - \alpha)I)^m\|_\infty \\
&= \|\tilde{P}_{\alpha,\pi}^m - (\alpha E + \tilde{P}_{\alpha,\pi})^m\|_\infty \\
&\leq (\alpha \|E\|_\infty + 1)^m - 1 \\
&\leq (\alpha^2 |A|\psi(n) + 1)^m - 1
\end{aligned}
$$

where in the second-to-last inequality uses that $\|\tilde{P}_{\alpha,\pi}\|_\infty = 1$ and $\|AB\|_\infty \leq \|A\|_\infty \|B\|_\infty$ for matrices A, B. Putting it all together we have that

$$|g_\pi^P - g_\pi^{\tilde{P}}| \leq c_{\max} \frac{(\alpha^2 |A|\psi(n) + 1)^m - 1}{1 - \Delta(\tilde{P}_{\alpha,\pi}^m)} \tag{11}$$

We therefore require that

$$\psi(n) \leq \frac{1}{\alpha^2 |A|} \left(\left(\frac{\epsilon_{(2)}(1 - \Delta(\tilde{P}_{\alpha,\pi}^m))}{c_{\max}} + 1 \right)^{1/m} - 1 \right) \tag{12}$$

to yield $|g_\pi^P - g_\pi^{\tilde{P}}| < \epsilon_{(2)}$. The equation (12) also holds with $\tilde{P}$ replaced with P, with (some other) m appropriate.

In the AMEC $(A, \mathcal{A}_A)$ then there are at most $|\Pi_A| = |A|^{|\mathcal{A}_A|}$ deterministic policies. For each policy $\pi \in \Pi_A$, there is some η_π^* satisfying Lemma B.7. Let $m = \max_{\pi \in \Pi_A} \min_{m \in \mathbb{N}} \{m | \Delta(P_{\alpha,\pi_\eta^*}^m) < 1\}$

26

and $\bar{\Delta}_A = \max_{\pi \in \Pi_A} \Delta(P^m_{\alpha, \pi_{\eta^*_\pi}}) < 1$ (recall this is guaranteed because $P_{\alpha, \pi_{\eta^*_\pi}}$ is ergodic). Then, when $\psi(n) < \frac{1}{\alpha^2 |A|} \left(\left(\frac{\epsilon_{(2)}(1-\bar{\Delta}_A)}{c_{\max}} + 1 \right)^{1/m} - 1 \right)$ then $|g^P_\pi - g^{\tilde{P}}_\pi| < \epsilon_{((2)}$. By Lemma B.2, we have

$$n = \tilde{\mathcal{O}}\left(\frac{|A|^{\frac{2}{m}} c_{\max}^{\frac{2}{m}}}{\epsilon_{(2)}^{\frac{m}{m}}(1-\bar{\Delta}_A)^{\frac{2}{m}}} \right) = \tilde{\mathcal{O}}\left(\frac{|A|^2 c_{\max}^2}{\epsilon_{(2)}^2 (1-\bar{\Delta}_A)^2} \right), \text{ since } m = 1 \text{ achieves the maximum.} \qquad \square$$

Remark B.9. *We do not require knowledge of $\bar{\Delta}_A < 1$. The existence is sufficient to guarantee convergence.*

Remark B.10. *The function $\Delta(M)$, coefficient of ergodicity of matrix M, is a measure (and bound) of the second largest eigenvalue of M.*

Remark B.11. *In the main paper, we assume that $m = 1$ and $\alpha = 1$, for simplicity in exposition. For full rigor, m may be larger, though typically small. m can be seen as the smallest value making any column of $P^m_{\alpha, \pi}$ dense. From a computational perspective, it is efficient to compute powers of $\tilde{P}_{\alpha, \pi}$ and stop when $\tilde{P}^m_{\alpha, \pi}$ has a dense column, making $\Delta(\tilde{P}_{\alpha, \pi}) < 1$. From there, we can check if ρ (Line 6, Algo 2) satisfies the r.h.s of Eq (12). We present the samples required by maximizing over $m \in \mathbb{N}$.*

B.5 `PlanTransient` **proofs**

Proposition 4.3 (`PlanTransient` Convergence & Correctness, Informal). *Denote the cost- and prob-optimal policy as π'. After collecting at most $n = \tilde{\mathcal{O}}(\frac{|\mathcal{S}\setminus\cup_{i=1}^{k}A_i|^2\bar{V}^4}{c_{\min}^2\epsilon_{PT}^2\epsilon_\varphi^4})$ samples for each $(s,a) \in (\mathcal{S} \setminus \cup_{i=1}^{k}A_i) \times \mathcal{A}$, the greedy policy π w.r.t. v' (Alg. 3, Line 3) is both cost and probability optimal:*
$$\|\tilde{V}_\pi - \tilde{V}_{\pi'}\| < \epsilon_{PT}, \quad |\mathbb{P}[\pi \text{ reaches } \cup_{i=1}^{k}A_i] - \mathbb{P}[\pi' \text{ reaches } \cup_{i=1}^{k}A_i]| \le \epsilon_\varphi.$$

Proposition B.12 (`PlanTransient` Convergence & Correctness, Formal). *Let $\{A_i, g_i\}_{i=1}^{k}$ be the set of inputs to Algorithm 3, together with error $\epsilon_{PT} > 0$. Denote the cost- and prob-optimal policy as π'. After collecting at most $n = \tilde{\mathcal{O}}(\frac{|\mathcal{S}\setminus\cup_{i=1}^{k}A_i|^2\bar{V}^4}{c_{\min}^2\epsilon_{PT}^2\epsilon_\varphi^4})$ samples for each $(s,a) \in (\mathcal{S} \setminus \cup_{i=1}^{k}A_i) \times \mathcal{A}$, under the event $\mathcal{E}$ and Assumption 1 then with probability $1 - \delta$, , the greedy policy π w.r.t. v' (Alg. 3, Line 3) is both cost and probability optimal:*
$$\|\tilde{V}_\pi - \tilde{V}_{\pi'}\| < \epsilon_{PT}, \quad |\mathbb{P}[\pi \text{ reaches } \cup_{i=1}^{k}A_i] - \mathbb{P}[\pi' \text{ reaches } \cup_{i=1}^{k}A_i]| \le \epsilon_\varphi.$$

Proof of 4.3. Convergence follows from boundedness of $\|v\| \le \bar{V}$, and monotone convergence and is well studied [46, 29, 59, 22].

Fix $\lambda > 0$ and drop it from the notation $V_{\pi,\lambda}^P$. Let $\tilde{V}_*^{\tilde{P}}$ be the value function for the optimal policy in $\tilde{P}$. For the approximation error, we have
$$0 \le \tilde{V}_\pi^P - \tilde{V}_*^P = \underbrace{\tilde{V}_\pi^P - \tilde{V}_\pi^{\tilde{P}}}_{(a)} + \underbrace{\tilde{V}_\pi^{\tilde{P}} - \tilde{V}_*^P}_{(b)} < \epsilon_{PT}$$

For (a) we appeal to Lemma $B.15$ and set $\epsilon_{(3)} = \epsilon_{PT}/2$ requiring that $\psi(n) = \frac{\epsilon_{PT}c_{\min}}{14|\mathcal{S}\setminus\cup_{i=1}^{k}A_i|\bar{V}^2(1+\frac{1}{\epsilon_\varphi})^2}$, occuring when $n = \tilde{\mathcal{O}}((\frac{|\mathcal{S}\setminus\cup_{i=1}^{k}A_i|\bar{V}^2}{\epsilon_{PT}\epsilon_\varphi^2 c_{\min}})^2)$ samples per state-action pair have been collected. For (b), by Lemma B.16, by selecting $\epsilon_{PT}^{\mathcal{L}} = \frac{c_{\min}\epsilon_{PT}\epsilon_\varphi}{4\bar{V}}$ we have that
$$V_\pi^{\tilde{P}} - V_*^P \le (1 + \frac{2\epsilon_{PT}^{\mathcal{L}}}{c_{\min}})v - V_*^P$$
$$= \frac{2\epsilon_{PT}^{\mathcal{L}}v}{c_{\min}}$$
$$\le \frac{2\bar{V}\epsilon_{PT}^{\mathcal{L}}}{\epsilon_\varphi c_{\min}} \le \frac{\epsilon_{PT}}{2}.$$

For the probability of satisfaction, by Prop B.13, we have that π and π^* coincide in probability of reaching the states in $\cup_{i=1}^{k}A_i$. $\qquad\square$

Proposition B.13 (Selecting a bound on $\|v\|$). *Let $\{A_i, g_i\}_{i=1}^{k}$ be the set of inputs to Algorithm 3. Let π' have maximal probability of reaching $\cup_{i=1}^{k}A_i$. Then, with error $\epsilon_\varphi > 0$, bounding $\|v\|_\infty = \|\mathcal{L}_{PT}v\|_\infty \le \frac{\bar{V}}{\epsilon_\varphi}$ where $\bar{V} \ge \left(\frac{1}{\beta^{|\mathcal{S}|}}\left(\frac{1-\beta^{|\mathcal{S}|}}{1-\beta}\right) + \lambda\right)c_{\max}$ guarantees that π returned by Algorithm 3 is near probability optimal:*
$$|\mathbb{P}[\pi \models \varphi] - \mathbb{P}[\pi' \models \varphi]| < \epsilon_\varphi$$

Proof of B.13. Suppose $\bar{V} \ge J_\pi + \lambda c_{\max}$ for any $\pi \in \Pi$. Let $\frac{\bar{V}}{\epsilon_\varphi}$ be chosen as upper bound on $\|v\| = \|\mathcal{L}_{PT}v\|$. Denote $\mathbb{P}[\pi \models \varphi]$ as p, and $\mathbb{P}[\pi' \models \varphi]$ as p^*. Suppose, for contradiction, $p^* - p > \epsilon_\varphi$, yet π is returned by the Algorithm. This would imply that $\tilde{V}_\pi \le \tilde{V}_{\pi'}$.

Hence,
$$0 \le \tilde{V}_{\pi'} - \tilde{V}_\pi \le \underbrace{p^*(J_{\pi'} + \lambda\sum_{i=1}^{k}\frac{p_i^*}{p^*}\hat{g}_{\tilde{\pi}A_i}^{\tilde{P}})}_{\le J_\pi + \lambda c_{\max}} - \underbrace{p(J_\pi + \lambda\sum_{i=1}^{k}\frac{p_i}{p}\hat{g}_{\tilde{\pi}A_i}^{\tilde{P}})}_{\ge 0} + \underbrace{(p - p^*)}_{<-\epsilon_\varphi}\frac{\bar{V}}{\epsilon_\varphi}$$
$$< J_\pi + \lambda c_{\max} - \bar{V}$$
$$\le 0$$

28

Hence, we have a contradiction. Thus, $|p^* - p| \leq \epsilon_\varphi$ if $\bar{V} \geq J_\pi + \lambda c_{\max}$ for any $\pi \in \Pi$. In fact, since the solution to $\mathcal{L}_{\mathrm{PT}}$ is deterministic, it suffices to consider only deterministic Π.

We will now bound $J_\pi = \mathbb{E}_{\tau \sim \mathrm{T}_\pi}\left[\sum_{t=0}^{\kappa_\pi} \mathcal{C}(s_t, \pi(s_t))\Big| \tau \models \varphi\right] \leq c_{\max}\mathbb{E}_{\tau \sim \mathrm{T}_\pi}[\kappa_\tau | \tau \models \varphi]$, as this is the only unknown quantity. Here $\mathbb{E}_{\tau \sim \mathrm{T}_\pi}[\kappa_\tau | \tau \models \varphi]$ is the expected number of steps it takes π to leave the transient states. This means that a worst-case bound would be a policy that remains in the transient states as long as possible.

We construct the worst-case scenario and give a justification, a formal proof follows from induction. Suppose the starting state is s_0. If π induces a prob-1 transition back to s_0 then s_0 is recurrent, and so κ_τ would be small. Instead, π induces a prob $1 - \beta$ transition to s_0 and a prob β transition to s_1. Notice that the transition to s_1 must be at least probability β due to Assumption 1. Again, if s_1 gave all of its probability to s_1 or s_0 then a MEC would form and strictly decrease κ_τ. This process repeats until we reach state $s_{|\mathcal{S}|-1}$, which has to have a self-loop. If it does not, then, again a large MEC would form and decrease κ_τ. Of course, this is the well known chain graph, with easily computable expected hitting time: $\mathbb{E}_{\tau \sim \mathrm{T}_\pi}[\kappa_\tau] \leq \frac{1}{\beta^{|\mathcal{S}|}}\frac{1-\beta^{|\mathcal{S}|}}{1-\beta}$. By making $s_{|\mathcal{S}|-1}$ the accepting state, then $\mathbb{E}_{\tau \sim \mathrm{T}_\pi}[\kappa_\tau | \tau \models \varphi] = \mathbb{E}_{\tau \sim \mathrm{T}_\pi}[\kappa_\tau] = \frac{1}{\beta^{|\mathcal{S}|}}\frac{1-\beta^{|\mathcal{S}|}}{1-\beta}$ achieves the bound. Any other choice of accepting states would strictly decrease κ_τ. Hence, we can select

$$\bar{V} \geq \left(\frac{1}{\beta^{|\mathcal{S}|}}\left(\frac{1-\beta^{|\mathcal{S}|}}{1-\beta}\right) + \lambda\right) c_{\max} \geq J_\pi + \lambda c_{\max},$$

completing the proof. $\qquad\square$

Remark B.14. *It may also be possible to empirically estimate J_π rather than take the bound from Prop B.13, considering that we have the structure of P through $\widehat{P}$. We give the high level idea. We know all of the AMECs and rejecting EC, so we have all the transient states (denoted T). Then for some policy π and $P' \in \mathcal{P}$, submatrix $Q_\pi(s, s') = P'_\pi(s, s')$ for $s, s' \in T$ represents the transitions in the transient states. It is well known that $\mathbb{E}_{\tau \sim \mathrm{T}_\pi}[\kappa_\tau] = \|(I - Q)^{-1}\|_\infty$. Taking the max over all $\pi \in \Pi$, $P' \in \mathcal{P}$, and finally multiplying by $c_{\max}$ gives a bound on J_π.*

Lemma B.15. *(Simulation Lemma, Transient Cost [58]) Consider an MDP $(\mathcal{S}, \mathcal{A}, \ldots)$. For any two transition functions $P', P'' \in \mathcal{P}$, policy π, and error $\epsilon_{(3)} > 0$ then*

$$\|\tilde{V}_\pi^{P''}\|_\infty = \|\tilde{V}_\pi^{P'}\|_\infty \leq (1 + \frac{1}{\epsilon_\varphi})\bar{V}, \quad \|\tilde{V}_\pi^{P'} - \tilde{V}_\pi^{P''}\|_\infty \leq \frac{7|\mathcal{S}|\bar{V}^2(1 + \frac{1}{\epsilon_\varphi})^2\psi(n)}{c_{\min}} \leq \epsilon_{(3)}$$

occurring after $n = \tilde{\mathcal{O}}(\frac{|\mathcal{S}|^2\bar{V}^4}{\epsilon_{(3)}^2\epsilon_\varphi^4 c_{\min}^2})$ samples from each state-action pair in $\mathcal{S} \times \mathcal{A}$.

Proof. Direct consequence of the definition of $\bar{V}$ from Prop B.13, application of Lemma 2 from [58] and Lemma B.2. $\qquad\square$

Lemma B.16. *(EVI Bound, [58]) Suppose v is returned by VI with accuracy $\epsilon_{PT}^{\mathcal{L}}$ with Bellman equation $\mathcal{L}_{PT}$ (See Table 1). Suppose π is greedy with respect to v. If $\epsilon_{PT}^{\mathcal{L}} \leq \frac{c_{\min}}{2}$ then, element-wise,*

$$v \leq \tilde{V}_{\pi^*}^P, \quad v \leq \tilde{V}_\pi^{\tilde{P}} \leq (1 + \frac{2\epsilon_{PT}^{\mathcal{L}}}{c_{\min}})v$$

29

C   Conjecture on Sample Complexity

As we have proven in Theorem 5.1, the optimal policy creates a set of AMECs which coincide with $(A_i, \mathcal{A}_i)_{i=1}^{k}$. For any potential AMEC, we need to guarantee probabilistic closure. For each state-action pair $(s, a) \in A_i \times \mathcal{A}_{A_i}$ we have to sample enough times to guarantee that we have "collected" all of the possible unique transitions (s, a, s'). Indeed, this is similar to the famous coupon collection problem, where we want to know how much time it will take to collect all unique transitions (s, a, s'). Suppose there are m unique tuples each with probability $\beta = \frac{1}{m}$.

We can use a Chebyshev-based lower bound:

$$\mathbb{P}[N > m \log m - \log(\frac{1}{\delta})m] \geq \delta$$

Simplifying, we get that $\mathbb{P}[N > m \log(\frac{m}{\delta})] \geq \delta$. Thus, the number of transitions needed is

$$N = \Omega(m \log(\frac{m}{\delta})) = \Omega(\frac{1}{\beta} \log(\frac{1}{\beta\delta})) = \tilde{\Omega}(\frac{1}{\beta})$$

Further, [65] show that indeed $N \geq \frac{\beta}{\log \beta}$.

30

D Additional Algorithms

In this section we discuss the additional subroutines used in this paper. We discuss the case where selecting $\cup_{i=1}^k A_i$ as the terminal states for SSP in Algo 1 can fail and an alternative solution.

D.1 Value Iteration

Our version of Value Iteration `VI` (Algo 4) is a two-in-one version, due to the similarity of Relative VI (used in `PlanRecurrent`) and SSP (used in `PlanTransient`). The general idea is that you apply the Bellman Operator $\mathcal{L}$ onto your current iterate v_n repeatedly until $d(v_{n+1}, v_n)$ exceeds ϵ. When we wish to find the gain, then V_T (terminal states) is empty, and we use shifting by the first value of $v_n(0)$ for stability [12]. In other words, we subtract $v_n(0)$ from every value of v_n. On the other hand, if a set of terminal costs is provided then these represent the set of states that we want to reach through SSP and the value $v_n(s) = V_T(s)$ is known and must be kept fixed throughout applications of $\mathcal{L}$. The only difference in our application of $\mathcal{L}$ over standard Bellman operators is that $\mathcal{L}$ is optimistic and has an interior minimization over $\min_{p \in \mathcal{P}(s,a)} p^T v_n$ (See Table 1). To solve this, minimization we use a modified version from [29] given in Algo 5. The idea of Algo 5 is simple: put all the mass of $\tilde{P}$ onto the lowest possible values of v_n while still being consistent with $\widehat{P}$. This is efficient as it requires an ordering over v and then a single pass over the states $\tilde{S}$. The calculated probability $p(\tilde{s}_l)$ (see Algo 5) are what we call the optimistic dynamics $\tilde{P}(s, a, \tilde{s}_l)$.

Algorithm 4 Value Iteration (`VI`)

Param: Optimistic Bellman Operator $\mathcal{L}$, Error Measure d, accuracy $\epsilon > 0$, V_T terminal values (optional)
1: Set $n = 0, v_0 = 0_S, v_1 = \mathcal{L}v_0$
2: **repeat**
3:    $n \xleftarrow{+} 1$
4:    **if** V_T is empty **then**
5:       Shift $v_n \leftarrow v_n - v_n(0)\mathbf{1}$ // Relative Value Iteration
6:    **else**
7:       $v_n(s) \leftarrow V_T(s)$ for $s \in V_T$ // SSP
8:    Apply operator $v_{n+1}, \tilde{P} \leftarrow \mathcal{L}v_n$ // Bellman Backup
9: **until** $d(v_{n+1}, v_n) > \epsilon$
10: **return** $v_{n+1}, v_n, \tilde{P}$

Algorithm 5 `InnerMin` (for PT/PR)

Param: A set of states $\tilde{S}$, current estimate from VI v_n, estimates $\widehat{P}(s, a, \cdot)$ for a specific (s, a) pair with $s \in \tilde{S}$, errors $\psi(n)$, lower bound β (See Assumption 1)
1: Sort $\tilde{S} = \{\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_m\}$ according to $v_n(\tilde{s}_1) \le v_n(\tilde{s}_2) \le \dots \le v_n(\tilde{s}_m)$, where v_n is the current
2: Set
$$p(\tilde{s}_1) = \begin{cases} \min(1 - \beta, \widehat{P}(s, a, \tilde{s}_1) + \psi(n)), & \widehat{P}(s, a, \tilde{s}_1) \notin \{0, 1\} \\ 1, & \widehat{P}(s, a, \tilde{s}_1) = 1 \\ 0, & \widehat{P}(s, a, \tilde{s}_1) = 0 \end{cases}$$
3: For remaining $j > 1$, set $p(\tilde{s}_j) = \widehat{P}(s, a, \tilde{s}_j)$
4: Set $l \leftarrow m$
5: **while** $\sum_{\tilde{s}_j \in \tilde{S}} p(\tilde{s}_j) > 1$ **do**
6:    Reset
$$p(\tilde{s}_l) = \begin{cases} \max(\beta, 1 - \sum_{\tilde{s}_j \ne \tilde{s}_l} p(\tilde{s}_j)), & \widehat{P}(s, a, \tilde{s}_l) \notin \{0, 1\} \\ 1, & \widehat{P}(s, a, \tilde{s}_l) = 1 \\ 0, & \widehat{P}(s, a, \tilde{s}_l) = 0 \end{cases}$$
7:    Decrement $l \leftarrow l - 1$
8: Set $\tilde{P}(s, a, \tilde{s}) = p(\tilde{s})$ for each $\tilde{s} \in \tilde{S}$
9: **return** $\tilde{P}(s, a, \tilde{s})$

31

D.2 Modified Algorithm handling Blocking Failure in Algorithm 1
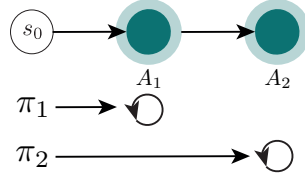
Figure 4: *Blocking Issue. If A_1 is included in the terminal AMECs (the states we want to reach) then once it is reached π_{A_1} is instantiated and A_1 becomes recurrent, implying only π_1 is considered. However, even though it may be the case that $J_{\pi_1} < J_{\pi_2}$, we may still have $V_{\pi_1} > V_{\pi_2}$. This example demonstrates the necessity to pick the terminal AMECs properly, rather than just the union of all AMECs found, to avoid blocking.*

One of the failure modes of Algorithm 1 is in its selection of which AMECs are the necessary AMECs to reach. In fact, by selecting unnecessary AMECs, the SSP procedure fails to treat some AMECs as transient states when in fact, maybe, lower cost could have been achieved if they were. One way to see this is to consider a single directional chain of AMECS (See Figure 4). In the figure, two policies can be considered: (1) π_1 that reaches for A_1 and then starts π_{A_1} when A_1 is reached, and (2) π_2 that reaches for A_2 and then starts π_{A_2} when A_2 is reached. It may be the case that $V_{\pi_2} < V_{\pi_1}$ despite $J_{\pi_2} > J_{\pi_1}$, since it requires a longer cost path to reach the desired AMEC. Despite this observation, when A_1 is selected as terminal states in the subroutine `PlanTransient` (Algo 3), we disallow consideration of π_2 at all. As explained in the proof of Theorem 5.1, whatever AMECs are induced by π^* coincide with $AMEC = \{A_i, \mathcal{A}_{A_i}\}_{i=1}^k$. Let $\Omega = 2^{AMEC} \setminus \varnothing$, all non-empty subsets of AMECs (possible targets). Since all accepting trajectories of π^* land in an AMEC, then another way of looking at π^* is:

$$\pi^* = \min_{\omega \in \Omega} \min_{\pi \in \tilde{\Pi}(\omega)} V_\pi$$

where $\tilde{\Pi}(\omega) = \{\pi \in \Pi_{\max} | \pi(s, a) = \pi_{A_i}(s, a) \text{ for } s \in A_i \in \omega, a \in \mathcal{A}_{A_i}(s)\}$, which is a policy class where the only degrees of freedom are outside of ω. In other words, $\pi \in \tilde{\Pi}(\omega)$ is followed until the trajectory hits $A_i \in \omega$ and then π_{A_i} is followed thereafter.

We will reconcile this failure mode of `PlanTransient` through a modified, nonblocking, subroutine `NoBlockPlanTransient` (Algo 6).

Algorithm 6 `NoBlockPlanTransient` (NB-PT)

Param: States & gains: $\{(A_i, g_i)\}_{i=1}^k$, err. $\epsilon_{\text{PT}} > 0$
1: Set $v(s) = \infty$ for each $s \in \mathcal{S}$.
2: Sample ϕ_{PT} times $\forall (s, a) \in \mathcal{S} \times \mathcal{A}$
3: **for** $\omega \in 2^{\{A_i\}_{i=1}^k} \setminus \varnothing$ **do**
4:     Set $V_T(s) = \lambda g_i$ for $s \in A_i \subseteq \omega$
5:     $v'_\omega, v_\omega, \tilde{P} \leftarrow \text{VI}(\mathcal{L}_{\text{PT}}, d_{\text{PT}}, \epsilon_{\text{PT}}^\mathcal{L}, V_T)$
6:     **if** $\mathbb{E}_{s \sim d_0}[v'_\omega(s)] < \mathbb{E}_{s \sim d_0}[v(s)]$ **then**
7:         Set $v = v'_\omega$
8: Set $\pi \leftarrow$ greedy policy w.r.t v
9: **return** π

The proof of correctness follows from the fact that v'_ω closely tracks V_π where π is greedy wrt v'_ω. Then, selecting the smallest V_π coincides with V_{π^*}.

Proposition D.1 (Proof of Correctness and Convergence of `NoBlockPlanTransient`). *After collecting at most $n = \tilde{\mathcal{O}}\left(\frac{|\mathcal{S}|^2 \bar{V}^4}{c_{\min}^2 \epsilon_{Pr}^2 \epsilon_\varphi^4}\right)$ samples for each $(s, a) \in \mathcal{S} \times \mathcal{A}$, under the event $\mathcal{E}$ and Assumption 1 then with probability $1 - \delta$, , the greedy policy π w.r.t. v' (Alg. 3, Line 3) is both cost and probability optimal:*

$$\|\tilde{V}_\pi - \tilde{V}_{\pi^*}\| < \frac{3\epsilon_{PT}}{2}, \quad |\mathbb{P}[\pi \models \varphi] - \mathbb{P}[\pi^* \models \varphi]| \le \epsilon_\varphi.$$

32

Proof. Suppose $v_\omega < v'_\omega$ for any $\omega' \in \Omega$, with $\omega \in \Omega$. Fix some ω'. Denote the greedy policies $\pi_{v_\omega}, \pi_{v_{\omega'}}$ wrt $v_\omega, v_{\omega'}$. Suppose $\tilde{V}_{\pi_{v_{\omega'}}} < \tilde{V}_{\pi_{v_\omega}}$. Then an error was made and

$$0 \leq \tilde{V}^P_{\pi_{v_\omega}} - \tilde{V}^P_{\pi_{v_{\omega'}}} \leq \tilde{V}^P_{\pi_{v_\omega}} - \tilde{V}^{\tilde{P}}_{\pi_{v_\omega}} + \tilde{V}^{\tilde{P}}_{\pi_{v_\omega}} - v_\omega + v_\omega - v_{\omega'} + v_{\omega'} - \tilde{V}^{\tilde{P}}_{\pi_{v_{\omega'}}} + \tilde{V}^{\tilde{P}}_{\pi_{v_{\omega'}}} - \tilde{V}^P_{\pi_{v_{\omega'}}}$$

$$\leq \frac{\epsilon_{\mathrm{PT}}}{2} + \frac{\epsilon_{\mathrm{PT}}}{2} + 0 + 0 + \frac{\epsilon_{\mathrm{PT}}}{2}$$

$$\leq \frac{3\epsilon_{\mathrm{PT}}}{2}$$

where the second line comes from grouping each pair of elements from the first line and applying the bounds found in proof of Proposition B.12.

On the other hand, suppose $p + \epsilon_\varphi = \mathbb{P}[\pi_{v_\omega} \models \varphi] + \epsilon_\varphi < \mathbb{P}[\pi_{v_{\omega'}} \models \varphi] = p'$. The same proof as in Prop B.13 applies to show that the probability of satisfaction remains close:

$$0 \leq \tilde{V}_{\pi_{v_{\omega'}}} - \tilde{V}_{\pi_{v_\omega}} \leq \underbrace{p'(J_{\pi_{v_{\omega'}}} + \lambda \sum_{i=1}^{k} \frac{p'_i}{p'} \widehat{g}_{\tilde{\pi}_{A_i}})}_{\leq J_\pi + \lambda c_{\max}} - \underbrace{p(J_{\pi_{v_\omega}} + \lambda \sum_{i=1}^{k} \frac{p_i}{p} \widehat{g}_{\tilde{\pi}_{A_i}})}_{\geq 0} + \underbrace{(p - p')}_{<-\epsilon_\varphi} \frac{\bar{V}}{\epsilon_\varphi}$$

$$< J_\pi + \lambda c_{\max} - \bar{V}$$

$$\leq 0$$

showing that $|p - p'| < \epsilon_\varphi$.

In particular, since the choice of ω' was arbitrary, it holds for ω' achieving $\omega' = \min_{\omega \in \Omega} \min_{\pi \in \tilde{\Pi}(\omega)} V_\pi$. Therefore the previous bounds all hold for with p' replaced with p^* and $\tilde{V}^P_{\pi_{v_{\omega'}}}$ replaced with $\tilde{V}^P_{\pi^*}$.

It is clear we can think of this non-blocking subroutine as checking the different inputs to Algo 3, which requires $\phi_{\mathrm{PT}}(\omega) = \frac{\epsilon_{\mathrm{PT}} c_{\min}}{14|\mathcal{S}\backslash\omega|\bar{V}^2(1+\frac{1}{\epsilon_\varphi})^2}$, occuring when $n = \tilde{\mathcal{O}}((\frac{|\mathcal{S}\backslash\omega|\bar{V}^2}{\epsilon_{\mathrm{PT}}\epsilon_\varphi^2 c_{\min}})^2)$ samples per state-action pair have been collected. Taking the maximum over $\omega \in \Omega$, we have $n = \tilde{\mathcal{O}}((\frac{|\mathcal{S}|\bar{V}^2}{\epsilon_{\mathrm{PT}}\epsilon_\varphi^2 c_{\min}})^2)$ samples required for each state-action pair in $\mathcal{S} \times \mathcal{A}$. $\qquad\square$

Remark D.2. *Recall $\mathcal{S}^*$ is set of accepting states in Product-MDP $\mathcal{X}$. This subroutine appears to have an exponential runtime in $|\mathcal{S}^*|$; Ω is at most $2^{|\mathcal{S}^*|}$, which is not related to the typical PAC parameters. In general, Ω is modestly small.*

Remark D.3. *While the runtime scales poorly with $|\mathcal{S}^*|$, the sample complexity remains PAC.*

Remark D.4. *We believe it is possible to bring the runtime of the subroutine to be polynomial in $|\mathcal{S}^*|$ by leveraging the MEC quotient structure (see [11]), but leave that for future work.*

33

E Experiments

E.1 Environments and Details
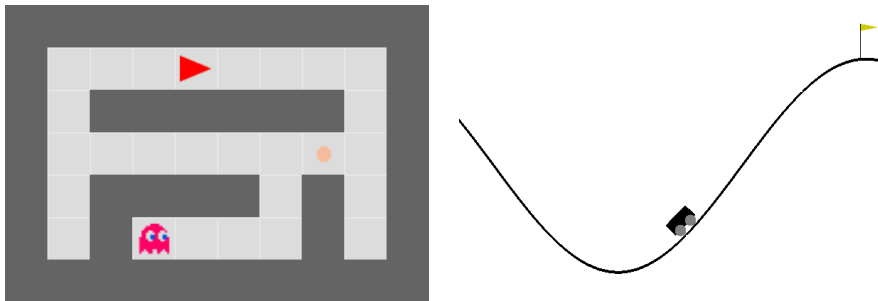
Figure 5: *Environment Illustrations. (Left) Pacman. φ is for the agent, the red triangle, to eventually collect the food, given by the yellow dot, and always avoid the ghost, the red semicircle with eyes. (Right) Mountain Car (MC). φ is to eventually reach the flag.*

Pacman. This environment (pictured in Fig 5 Left) is a 5x8 gridworld. The starting positions of the agent (red triangle), food (yellow circle), and ghost (red semicircle with eyes), are as illustrated in Fig 5. The agent has 4 cardinal directions at each state in addition to a "do nothing" action. The LTL specification is to eventually reach the food and to forever avoid the ghost "F(food) & G(!ghost)", where the food state is labelled "food" and the ghost state is labelled "ghost". Once the food is picked up, it is gone. The ghost chases the agent (following the shortest path) with probability .4 and chooses a random action with probability .6. Though this is an infinite horizon problem, as there is no terminal state, we allow a maximum horizon of $H = 100$ in our experiments. We track how long the agent has avoided the ghost and whether the agent has picked up the food. To simplify verification, we say the agent has satisfied the spec if the food has been picked up and the ghost has been avoided for all H timesteps. The cost function is defined as 1 everywhere.

For the shaped LCRL baseline, we use progression through the LDBA as a "reward": if the agent progresses to a new state in the automaton then the cost of that transition is .1 instead of 1. The authors of LCRL used similar ideas in their code as well. However, we must note that progression-based cost shaping eliminates any guarantee of LTL satisfaction. An agent is incentivized to find cycles in the LDBA rather than find an accepting state. In the case when no such cycles exist, then this form of cost shaping can work.

Mountain Car This domain (pictured in Fig 5 Right) is a discretization of the Mountain Car domain from OpenAI [14], with state-space given by tuple (position, velocity) and cost of 1. We discretize the position space into 32 equal size bins and the velocity into 32 geometrically-spaced bins, allowing more granularity around low velocity than high velocity, making 32^2 bins (states) in the MDP. The starting state is the standard MC starting state, but then placed in the appropriate bin. A bin can be converted back to (pos,vel), for purposes of sampling from P, by uniformly selecting from the valid positions/velocities implied by the bin. The agent has 3 actions: accelerate left, do nothing, accelerate right. The specification is to eventually reach the goal state "F(goal)", the standard task, where any bin with position beyond the flag position is labelled "goal".

For the shaped LCRL baseline, we use a cost function of $c = .1$ if the change in position is positive and the agent accelerated right, likewise if the change is negative and the agent accelerated left, otherwise $c = 1$. This cost function should incentivize the agent to seek actions which make the car go faster. Unlike the previous experiment, here cost-shaping has no effect on the guarantee of LTL satisfaction.

Safe Delivery This domain (pictured in Fig 1 Right) is a 4-state MDP: (0) start state, (1) sniffed packet, (2) stolen packet (3) delivered packet. In each state, the agent has two actions, A and B. The transition function P in the MDP is given by $P(0, A, 1) = 1$, $P(0, B, 2) = .5$, $P(0, B, 3) = .5$, $P(1, A, 3) = 1$, $P(1, B, 3) = 1$, $P(2, A, 2) = 1$, $P(2, B, 2) = 1$, $P(3, A, 3) = 1$, $P(3, B, 3) = 1$. In other words, choosing action A in the initial state immediately leads to a sniffed packet, which subsequently leads to the packet being delivered by any action. Alternatively, choosing action B in the initial state has a $50 - 50$ chance of having the packet stolen or immediately delivered, regardless

34

of action. Once, stolen, it remains stolen. Once delivered, the packet remains delivered, regardless of action. The states are labelled as $L(0) = L(3) =$"safe". The specification is to always stay in safe states: "G(safe)". Let all the costs be 1. The Product-MDP can be seen in Figure 2 Right.

The probability-optimal and cost-optimal policy is then choosing B is state 0 and then arbitrarily afterward. The maximum probability of satisfying the policy is 50% because 50% of the time the packet gets stolen. Though this is an infinite horizon problem, as there is no terminal state, we allow a maximum horizon of $H = 100$ in our experiments. Thus, the average number of timesteps should be $.5 * H = 50$.

Similarly to Pacman, for the shaped LCRL baseline, we use progression through the LDBA as a "reward": if the agent progresses to a new state in the automaton then the cost of that transition is .5 instead of 1.

Infinite Loop This environment (pictured in Fig 1 Left) is a 2x5 gridworld. The agent starts in the bottom right corner. The agent has 4 cardinal directions at each state in addition to a "do nothing" action. We consider two specifications:

φ_1: The LTL specification is to perpetually visit the office (in the top right corner) followed by the coffee room (top left corner): "GF(o & XFc)", where the office is labelled o and the coffee room is labelled c. The Product MDP is illustrated in Figure 2 Center.

φ_2: We require the agent to

$$\text{"G((c -> XXXXXo) \& (o ->XXXXXc)) \& Xo",} \tag{13}$$

meaning to get to first get to office in 1 step, then repeatedly reach the coffee room in 5 steps followed by the office in 5 steps.

Similarly to Pacman, for the shaped LCRL baseline, we use progression through the LDBA as a "reward": if the agent progresses to a new state in the automaton then the cost of that transition is .5 instead of 1.

E.2 Hyperparameters

We use the following hyperparameters for our experiments. Each set of hyperparameters was run with 20 seeds, with the exception of Safe Delivery which was run with 40 seeds.

Table 6: Hyperparameters

Param(s)	Infinite Loop φ_1	Infinite Loop φ_2	Safe Delivery	Pacman	MC
$\bar{V}$	50	50	10	100	150
$c_{\min}$	1	1	1	1	1
$c_{\max}$	1	1	1	1	1
φ	GF(o & XFc)	See φ_2 in (13)	G(!unsafe)	F(food0) & G!ghost	Fgoal
ϵ	3	3	3	3	10
δ	.1	.1	.1	.1	.1
LCRL Params	Infinite Loop	Infinite Loop 2	Safe Delivery	Pacman	MC
Max Traj len.	100	100	100	100	200
γ	.99	.99	.99	.99	.95
Learning rate	.95	.95	.95	.95	.9

E.3 Additonal Results

In this section we examine additional results for the experiments we ran.

For the Infinite Loop environment under φ_2, we see (Figure 6 Left Column) that our method is able to follow the trajectory specified by φ_2 even in low sample regimes. The learning signal for LCRL is very poor as the episode terminates extremely quickly if the agent does not get to the next location that it needs to be in within the allotted time. The shaped LCRL only does marginally better, but still struggles to satisfy the LTL with any probability.

For the Safe Delivery environment, we see (Figure 6 Left Right) that our method picks out the probability-optimal policy. LCRL is nearly optimal. The sparsity of this problem is significantly less

35
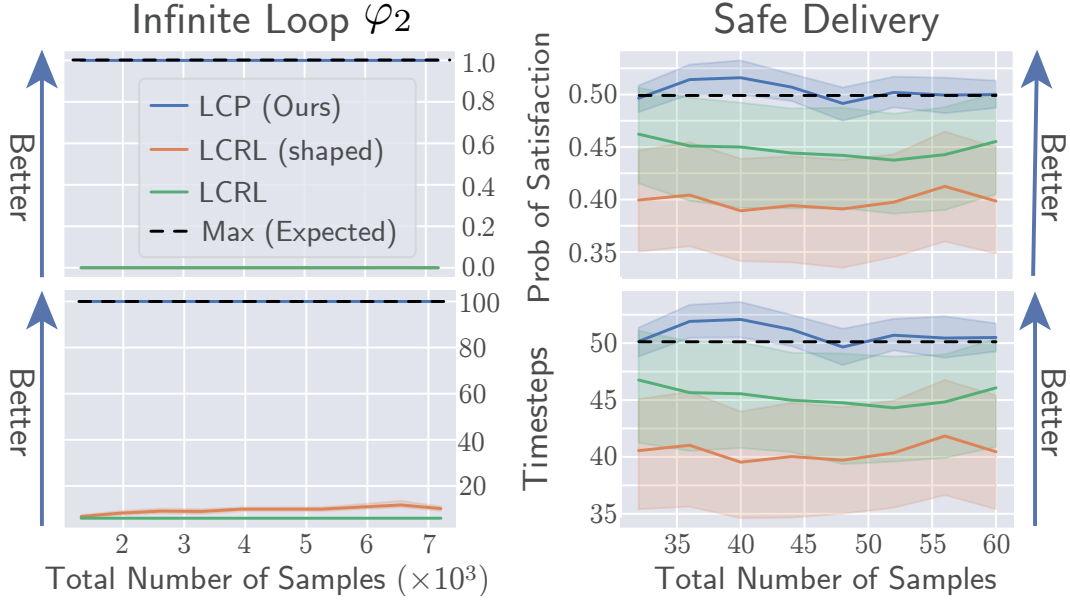
Figure 6: *Additional Results. (Left Column) Infinite Loop 2. φ is a specific trajectory that needs to be followed: first get to the office in 1 timestep and then the coffee room in 5 and then back to the office in 5, over and over. (Right) Safe Delivery (Right Column). φ is to always be safe.*

as the feedback for spec satisfaction verification comes after a single timestep. Interestingly, cost shaping in Safe Delivery performs worse than straight LCRL. This isn't surprising since, as noted, the verification feedback comes after a single timestep and is more important than any cost-shaping. However, cost-shaping muddles the feedback making shaped LCRL perform worse. We speculate that with only a few hundred or thousand more samples, both LCRL and shaped LCRL would reach the optimal policy. Recall that LCRL and shaped LCRL are not the same as Q-learning, as they operate in the product-MDP rather than the underlying MDP. Thus, these observations are still consistent with our Motivation section (Section 2), insisting that Q-learning would have trouble in this environment.

E.4 Policies
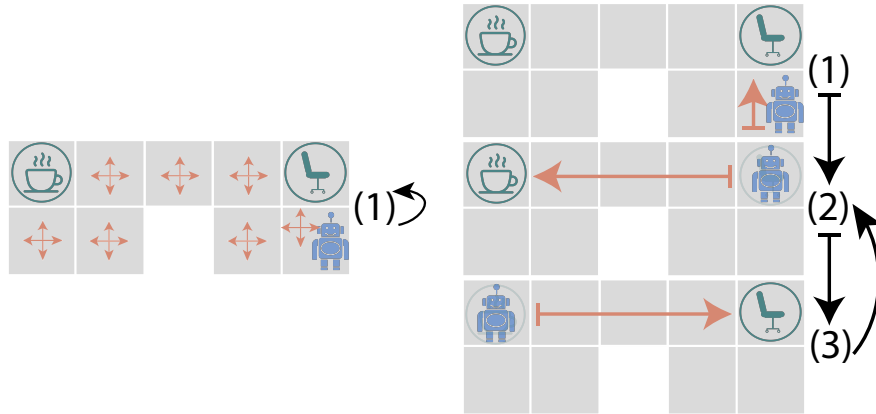
Figure 7: *Types of policies for different φ. (Left) Infinite Loop φ_1. φ_1 is to go perpetually walk between the office and the coffee room (Right) Infinite Loop φ_2. φ_2 is to get to the office in 1 time step then perpetually, take 5 timesteps to get to the coffee room and 5 steps back to the office.*

In this section we examine the policies induced by different specificity in specifications. In particular, we consider the Infinite Loop environment with two different specifications φ_1, φ_2, see Section E.1, E.2 for a description. For φ_1, we only require that the agent "eventually" navigate between the office

36

and coffee room. The agent is incentivized to stay in place (create a cost-1 cycle) for as long as possible and very infrequently take a random action. Of course, eventually taking random actions will loop the agent between the office and coffee room. This behavior is illustrated in Figure 7 Left, where the agent is always in LDBA state 1 and takes random actions with low probability and does nothing with high probability. It takes exponential time for the agent to make a single loop between the office and coffee room.

On the other hand, we may want the agent to move quickly. In this case, we can be more specific and use specification φ_2. The behavior for an agent satisfying φ_2 is illustrated in Figure 7 Right. The agent gets to LDBA state 2 by first reaching the office in a single time step. Then the agent loops between LDBA states 2 and 3 by reaching the coffee room and office, repeatedly, within the allotted time. If the agent does not reach the office or chair within the allotted time, there is a fourth LDBA state (unpictured) which is a sink denoting failure of the spec. In essence, the LDBA has created the options, or hierarchy, of solving the problem, as noted in Section 2. It takes 10 timesteps for the agent to make a single loop between the office and coffee room.

Notice that the high level description of the task is unchanged, but the details of how the task is accomplished is much more specific in φ_2 rather than φ_1. This demonstrates that writing LTL task specifications is flexible, but requires thought about "how" the task should be accomplished.

37