Model-Free Learning of Safe yet Effective Controllers

Alper Kamil Bozkurt, Yu Wang, and Miroslav Pajic

***Abstract*— We study the problem of learning *safe* control policies that are also *effective*; i.e., maximizing the probability of satisfying a linear temporal logic (LTL) specification of a task, and the discounted reward capturing the (classic) control performance. We consider unknown environments modeled as Markov decision processes. We propose a model-free reinforcement learning algorithm that learns a policy that first maximizes the probability of ensuring safety, then the probability of satisfying the given LTL specification and lastly, the sum of discounted Quality of Control rewards. Finally, we illustrate applicability of our RL-based approach.**

I. Introduction

Many sequential decision making problems involve multiple objectives. For example, a robotic task might require surveillance without hitting the walls while minimizing the energy consumption. Quality of Control (QoC) can be often inherently expressed via scalar reward signals where the overall performance of a control policy (i.e., QoC) is measured by the sum of discounted collected rewards (returns) [1]. Although these return maximization objectives are convenient for learning policies, the rewards may not be readily available and crafting them may not be trivial.

Alternatively, the high-level control objectives can be formulated using linear temporal logic (LTL) [2]. LTL provides a high-level intuitive language to formally specify desired behaviors of a control task. For a given LTL specification, the objective can be described as finding a policy maximizing the probability that the specification is satisfied. Synthesis of such policies directly from LTL specifications has attracted significant interest (e.g., [3], [4]). Although LTL specifications can naturally express many characteristics of interest such as safety, sequencing, conditioning, persistence and liveness; optimization of a quantitative performance measure usually cannot be captured by an LTL specification.

Consequently, in this work, we focus on control design problems where the objectives include satisfaction of desired properties expressed as LTL specifications as well as maximization of performance criteria represented by QoC rewards. In our problem setting, the LTL objectives are prioritized above the QoC objectives. In addition, we separate the safety specifications from the rest of the LTL specification as ensuring control safety is usually of the utmost importance.

The control design problem in stochastic environments has been widely studied for multiple return objectives (e.g., [5] and references therein), as well as for safety and LTL specifications [6]–[9]. Similarly, shielding-like approaches, which prioritize the safety objectives above the QoC objectives, have been introduced (e.g., [10], [11]). These approaches require some partial knowledge such as a topology or an abstraction of the environment to determine unsafe actions to be eliminated or overridden. This allows for the use of reinforcement learning (RL) algorithms to learn an optimal policy subject to the safety constraints. Unfortunately, when the environment is completely unknown, these synthesis or shielding approaches cannot be directly used.

Recently, there has been a considerable interest in the use of RL algorithms to learn policies for LTL objectives (e.g. [12]–[15]). However, only few RL algorithms for multi-objectives with LTL specifications have been proposed for unknown environments. Studies [16], [17] considered time-bounded LTL specifications; [18] proposed a model-based RL method that, under some assumptions about the environment, converges to a near optimal policy for an average QoC reward objective subject to the specifications; [19] introduced a model-free RL approach for multiple LTL objectives, maximizing the weighted sum of the satisfaction probabilities.

In this work, we introduce an RL algorithm for safety, LTL and QoC objectives with a lexicographic order where the safety objectives have the highest priority, while the LTL objectives have higher priority than the QoC-return objectives. As we demonstrate in a case study, it is crucial to consider the satisfaction of safety and LTL specifications as separate objectives with different priorities rather than the satisfaction of a single combined LTL specification because such approach might result in a policy under which the probability of satisfying the safety specification is significantly reduced. We show that our algorithm converges to a policy maximizing the probability that the LTL specification is satisfied under the constraint that the probability of satisfying the safety specification is maximized. Furthermore, the derived policy is near-optimal in terms of QoC returns that can be obtained while having these maximum satisfaction probabilities. Our method is ***completely model-free*** and does not rely on any assumptions about the environment and the specifications.

II. Preliminaries and Problem Statement

A. Markov Decision Processes

We consider the stochastic environments modeled as Markov decision processes (MDPs).

Definition 1. *A (labeled) MDP is a tuple* $\mathcal{M}=(S, s_0, A, P, R, \gamma, \text{AP}, L)$ *where* S *is a finite set of states;* s_0 *is the initial state;* A *is a finite set of actions and* $A(s)$ *denotes the set of actions allowed in a state* s*;* $P : S\times A\times S \mapsto [0,1]$ *is a probabilistic transition function such that* $\sum_{s'\in S} P(s,a,s') = 1$ *if* $a \in A(s)$*, and* 0 *otherwise;* $R : S \mapsto \mathbb{R}$ *is a reward function;* $\gamma \in [0,1)$ *is a discount factor;* AP *is a finite set of atomic propositions; and* $L : S \mapsto 2^{\text{AP}}$ *is a labeling function.*

This work is sponsored in part by the ONR under agreements N00014-17-1-2504, N00014-20-1-2745 and N00014-18-1-2374, AFOSR award number FA9550-19-1-0169, and the NSF CNS-1652544 grant.

Alper Kamil Bozkurt, Yu Wang, and Miroslav Pajic are with Duke University, Durham, NC 27708, USA, {alper.bozkurt, yu.wang094, miroslav.pajic}@duke.edu

In a state s, the controller takes an action $a \in A(s)$ and makes a transition to a state s' with probability (w.p.) $P(s,a,s')$, receiving a reward of $R(s')$ indicating the QoC, and a label $L(s') \subseteq AP$, a set of state properties. The actions to be taken by the controller are determined by a *policy*. In this study, we are interested in policies with limited memory.

Definition 2. *A **finite-memory policy** for an MDP $\mathcal{M}$ is a tuple $\pi = (M, m_0, T, \mathfrak{A})$ where M is a finite set of modes (memory states); m_0 is the initial mode; $T : M \times S \times M \mapsto [0,1]$ is a probabilistic transition function such that $T(m,s,m')$ is the probability that the policy switches to the mode m' after visiting the state s while operating in the mode m; $\mathfrak{A} : M \times S \times A \mapsto [0,1]$ is a function that maps a given mode $m \in M$ and a state $s \in S$ to the probability that a is taken in s when the current mode is m. A finite-memory strategy is called **pure memoryless** if there is only one mode ($|M| = 1$) and $\mathfrak{A}(m_0, s, a)$ is a point distribution assigning a probability of 1 to a single action in all states $s \in S$.*

Under a finite-memory policy, the action to be taken not only depends on the current state but also the current mode. After each transition, the policy may change its mode based on the visited state. The infinite sequence of states visited during an execution of the policy is called a *path*, denoted by $\rho = s_0 s_1 \ldots$, where s_t is the state visited at time step t. For a given path ρ, we use $\rho[t]$ and $\rho[t{:}]$ to denote the state s_t and the suffix $s_t s_{t+1}$..., respectively. The paths can be considered as sample sequences drawn from the *Markov chain* (MC), denoted by $\mathcal{M}_\pi$, induced by the followed policy π in an MDP $\mathcal{M}$; the state space of the induced MC is composed of the S and M, and the transitions are governed by P, T and $\mathfrak{A}$. We use $\mathcal{M}_\pi$ to denote the induced MC and $\rho \sim \mathcal{M}_\pi$ to denote a path randomly sampled from $\mathcal{M}_\pi$.

The QoC *return* of a path ρ, denoted by $G(\rho)$, is the sum of discounted QoC rewards collected on the path ρ – i.e.,

$$G(\rho) \coloneqq \sum_{t=0}^{\infty} \gamma^t R(\rho[t]); \tag{1}$$

the QoC *return objective* is to find a policy under which the expected QoC return of a path is maximized. When the reward and the probabilistic transition functions are unknown the optimal policies can be obtained using RL methods [1].

B. Linear Temporal Logic

We adopt LTL to specify the desired properties of a control policy. LTL specifications can be formed according to the following grammar:

$$\varphi \coloneqq \text{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathsf{U} \varphi_2, \ a \in \text{AP}. \tag{2}$$

The semantics of an LTL formula φ are defined over paths. A path ρ either satisfies φ, denoted by $\rho \models \varphi$, or not ($\rho \not\models \varphi$). The satisfaction relation is recursively defined as follows: $\rho \models \varphi$ if $\varphi = a$ and $L(\rho[0]) = a$; if $\varphi = \varphi_1 \wedge \varphi_2$, $\rho \models \varphi_1$ and $\rho \models \varphi_2$; if $\varphi = \neg\varphi'$ and $\rho \not\models \varphi'$; if $\varphi = \bigcirc\varphi'$ (*next* φ') and $\rho[1{:}] \models \varphi'$; if $\varphi = \varphi_1 \mathsf{U} \varphi_2$ (φ_1 *until* φ_2) and there exists $t \geq 0$ such that $\rho[t{:}] \models \varphi_2$ and for all $0 \leq i < t$, $\rho[i{:}] \models \varphi_1$. The Boolean operators *or* ($\vee$) and *implies* ($\rightarrow$) can be easily obtained via: $\varphi_1 \vee \varphi_2 \coloneqq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2 \coloneqq \neg\varphi_1 \vee \varphi_2$. In addition, we can derive the commonly used temporal operators *eventually* ($\Diamond$) and *always* ($\Box$) using: $\Diamond\varphi \coloneqq \text{true} \ \mathsf{U} \ \varphi$, and $\Box\varphi \coloneqq \neg(\Diamond\neg\varphi)$.

For an LTL formula, a limit-deterministic Büchi automaton (LDBA) that only accepts the paths satisfying the formula can be automatically constructed [20].

Definition 3. *An LDBA of an LTL formula φ is a tuple $\mathcal{A} = (Q, q_0, \Sigma, \delta, B)$ where Q is a finite set of automata states, $q_0 \in Q$ is an initial state, Σ is a finite alphabet, $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \mapsto 2^Q$ is a transition function, and $B \subseteq Q$ is a set of* accepting states. *The set $\delta(q,\theta)$ is singleton (i.e., $|\delta(q,\theta)| = 1$) for all $q \in Q$ and $\theta \in \Sigma$. The set Q can be partitioned into an initial and an accepting component, namely Q_I and Q_A, such that all the accepting states are in Q_A (i.e., $B \subseteq Q_A$), and Q_A does not have any ε or outgoing transitions; i.e., for any $q \in Q_A$,*

$$\delta(q,\theta) \subseteq \begin{cases} \varnothing & \text{if } \theta = \varepsilon, \\ Q_A & \text{if } \theta \in \Sigma. \end{cases} \tag{3}$$

A path ρ in an MDP $\mathcal{M}$ is accepted if there is an execution $\sigma = q_0 q_1 \ldots$ such that $q_{t+1} \in \delta(q_t, L(\rho[t])) \cup \delta(q_t, \varepsilon)$ for all $t \geq 0$ and $\text{Inf}(\sigma) \cap B \neq \varnothing$, where $\text{Inf}(\sigma)$ denotes the set of automata states visited by σ infinitely many times.

We write $\mathcal{A}_\varphi$ to denote an LDBA constructed by an LTL formula φ. All the accepting states of an LDBA $\mathcal{A}_\varphi$ are in the accepting component and φ is satisfied by a path if and only if the labels of states on the path trigger an execution that visits some of these accepting states infinitely often. The only nondeterministic transitions in an LDBA are the outgoing ϵ-transitions from the states in the initial component Q_I. When an LDBA consumes the label of a state $L(s)$, it deterministically transitions from its current state q to the state in the singleton set $\delta(q, L(s))$ if $\delta(q,\varepsilon)$ is empty. Once the accepting component Q_A is reached, it is not possible to make a nondeterministic transition. LDBAs can be constructed in a way that any ϵ-transition leads to a state in the accepting component, allowing for at most one ϵ-transition in an execution. These LDBAs are called *suitable*, as they facilitate quantitative model-checking of MDPs [20], and we henceforth assume that any given LDBA is suitable.

Safety LTL is an important fragment of LTL, which can be used to specify the properties ensuring "something bad never happens". For instance, any LTL formula in a positive normal form without any temporal modalities other than $\bigcirc$ and $\Box$ is a safety formula [21]. Safety formulas can be translated into simpler automata where the transitions are deterministic and all non-accepting automata states are absorbing, meaning visiting a non-accepting automata state results in rejection. We refer these automata as *safety automata* [21].

C. Problem Formulation

In this work, we focus on RL for three lexicographically ordered objectives: *safety* (primary), *LTL* (secondary) and *QoC* (tertiary). In our problem setting, the safety constraints are specified as an LTL safety formula and meeting these constraints is of highest priority. The other desired temporal properties are expressed as a general LTL formula and we are interested in policies satisfying the LTL formula with the highest probability. Finally, among such policies we want to learn the ones maximizing the expected QoC return. Hence, we formally state our problem as follows.

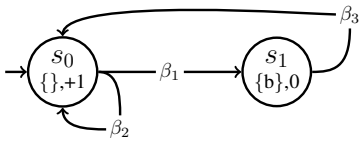

Fig. 1: An example deterministic MDP. The circles represent the states and the arrows represent the actions. The labels and QoC rewards of the states s_0 and s_1 are $\{\}, +1$ and $\{b\}, 0$ respectively.

Problem 1. *For a given MDP $\mathcal{M}$ where the transition probabilities P and the rewards R are unknown, a safety specification ψ and an LTL specification φ, design a model-free RL algorithm that learns a policy $\pi^R_{\psi,\varphi} \in \Pi^R_{\psi,\varphi}$ where*

$$\Pi^R_{\psi,\varphi} \coloneqq \underset{\pi\in\Pi_{\psi,\varphi}}{\operatorname{argmax}} \mathbb{E}_{\rho\sim\mathcal{M}_\pi}\left[G(\rho)\right], \tag{4}$$

$$\Pi_{\psi,\varphi} \coloneqq \underset{\pi\in\Pi_{\psi}}{\operatorname{argmax}} Pr_{\rho\sim\mathcal{M}_\pi}\left\{\rho \mid \rho \models \varphi\right\}, \tag{5}$$

$$\Pi_{\psi} \coloneqq \underset{\pi\in\Pi}{\operatorname{argmax}} Pr_{\rho\sim\mathcal{M}_\pi}\left\{\rho \mid \rho \models \psi\right\}, \tag{6}$$

and Π denotes the set of all finite-memory policies.

Intuitively, our objective is to learn a *lexicographically optimal* policy $\pi^R_{\psi,\varphi}$ that first maximizes the probability of ensuring the safety, then the probability of satisfying the LTL specification and lastly, the sum of discounted QoC rewards. However, such optimal policies might not always exist as illustrated in the following example.

Example 1. *Consider the MDP from Fig. 1 where the LTL specification φ is $\square\lozenge b$ and there is no safety specification (i.e., $\psi = \text{true}$). The policy π^* choosing β_2 in s_0 maximizes the QoC return. Now, let $\pi^R_{\psi,\varphi}$ be a policy in the set $\Pi^R_{\psi,\varphi}$ defined in* (4)*. The expected value of the QoC return obtained under $\pi^R_{\psi,\varphi}$ is strictly less than the one obtained under π^* because β_1 needs to be eventually taken by π^*_φ to satisfy φ, which leads to a reward of zero. Thus, a policy that randomly follows either π^* or $\pi^R_{\psi,\varphi}$ in s_0 increases the expected QoC return; the more often β_2 is chosen, the higher QoC return is obtained. In addition, such a mixed policy satisfies φ as well, because the mixed policy takes β_1 in s_1 with some positive probability due to $\pi^R_{\psi,\varphi}$. As a result, we obtain a policy better than $\pi^R_{\psi,\varphi}$, which leads to a contradiction.*

Consequently, we consider *near-optimality* for the QoC objective. Specifically, for a given $\nu{>}0$, a policy $\pi^{R\nu}_{\psi,\varphi}$ is called ν-optimal if it belongs to the set $\Pi^{R\nu}_{\psi,\varphi}$ defined as

$$\Pi^{R\nu}_{\psi,\varphi} \coloneqq \left\{\pi \in \Pi_{\psi,\varphi} \;\middle|\; v^R_{\psi,\varphi} - \mathbb{E}_{\rho\sim\mathcal{M}_\pi}\left[G(\rho)\right] \leq \nu\right\}, \tag{7}$$

where $v^R_{\psi,\varphi} \coloneqq \sup_{\pi\in\Pi_{\psi,\varphi}} \mathbb{E}_{\rho\sim\mathcal{M}_\pi}\left[G(\rho)\right]$. We note that the supremum exists as we assume the rewards are bounded.

III. Model-free RL for Lexicographic Safety, LTL and QoC objectives

In this section, we introduce our model-free RL algorithm that learns lexicographically near-optimal policies defined in (7). Our algorithm first constructs a product MDP by composing the initial MDP with the automata of the given safety and LTL specifications, where the satisfaction of these specifications is reduced into meeting the acceptance conditions of the automata; it then uses the safety and the LTL rewards crafted for the acceptance conditions as described in [15] and the QoC rewards of the original MDP to learn the policies via a model-free approach similar to [22].

A. Product MDP Construction

The first step in our approach is to construct a product MDP of the given MDP and the automata derived from the given safety and LTL specifications. The product MDP is merely a representation of the synchronous execution of the MDP and the automata. Thus, even if the transition graph and probabilities of the MDPs are unknown, the product MDP can be conceptually constructed; i.e., the resulting probabilities, and the transition graph, will be unknown.

Definition 4. *A product MDP of an MDP $\mathcal{M}$, a safety automaton $\mathcal{A}_\psi$, and an LDBA $\mathcal{A}_\varphi$ is a tuple $\mathcal{M}^\times = (S^\times, s^\times_0, A^\times, P^\times, R^\times, \gamma, B^\times_\psi, B^\times_\varphi)$ such that $S^\times = S \times Q_\psi \times Q_\varphi$ is the set of product states; $s^\times_0 = \langle s_0, q_{0\psi}, q_{0\varphi}\rangle$ is the initial product state; $A^\times = A \cup \{\epsilon_{q_\varphi} \mid q_\varphi \in Q_\varphi\}$ where $A^\times(\langle s, q_\psi, q_\varphi\rangle)$ denotes the set $A(s) \cup \{\epsilon_{q'_\varphi} \mid q'_\varphi \in \delta_\varphi(q_\varphi, \varepsilon)\}$; $P^\times : S^\times \times A^\times \times S^\times \mapsto [0,1]$ is the transition function such that $P^\times(\langle s, q_\psi, q_\varphi\rangle, a, \langle s', q'_\psi, q'_\varphi\rangle) =$*

$$\begin{cases} P(s,a,s') & \text{if } a{\in}A(s), q'_\psi{\in}\delta_\psi(q_\psi, L(s)), q'_\varphi{\in}\delta_\varphi(q_\varphi, L(s)),\\ 1 & \text{if } a{=}\epsilon_{q'_\varphi}, q'_\varphi{\in}\delta_\varphi(q_\varphi,\varepsilon), q'_\psi{=}q_\psi, s{=}s',\\ 0 & \text{otherwise;}\end{cases}$$

$R^\times : S^\times \mapsto \mathbb{R}$ is the reward function such that $R^\times(\langle s, q_\psi, q_\varphi\rangle) = R(s)$; γ is the discount factor; $B^\times_\psi = S \times B_\psi \times Q_\varphi$ is the set of accepting states for ψ; $B^\times_\varphi = S \times Q_\psi \times B_\varphi$ is the set of accepting states for φ.

Due to the ε-transitions in $\mathcal{A}_\varphi$, $\mathcal{M}^\times$ has ε-actions in addition to the actions in $\mathcal{M}$. Taking an ε-action ε_{q_φ} does not cause an actual transition in $\mathcal{M}$; it only makes $\mathcal{A}_\varphi$ to move to q_φ. A policy $\pi^\times$ for $\mathcal{M}^\times$ induces a finite-memory policy π for $\mathcal{M}$ where the automata states and transitions act as the memory mechanism. The mode of the induced policy π at any time step is represented by the joint state $\langle q_\psi, q_\varphi\rangle$ of the states that $\mathcal{A}_\psi$ and $\mathcal{A}_\varphi$ are in. When a state s is visited, π switches its mode to $\langle q'_\psi, q'_\varphi\rangle$, where q'_ψ and q'_φ are the automata states that $\mathcal{A}_\psi$ and $\mathcal{A}_\varphi$ transition to after consuming $L(s)$. The induced policy π then takes the action that $\pi^\times$ takes in $\langle s, \langle q'_\psi, q'_\varphi\rangle\rangle$.

A path $\rho^\times$ in $\mathcal{M}^\times$ satisfies the *safety condition*, denoted by $\rho^\times \models \square B^\times_\psi$, if $\rho^\times$ only visits the product states in $B^\times_\psi$. This means that its induced path ρ in $\mathcal{M}$ is accepted by $\mathcal{A}_\psi$ and thereby satisfying the safety specification ψ. Similarly, if $\rho^\times$ visits some product states in $B^\times_\varphi$ infinitely many times, $\rho^\times$ satisfies the *Büchi condition*, denoted by $\rho^\times \models \square\lozenge B^\times_\varphi$, and its induced path ρ satisfies the LTL specification φ. Finally, the QoC return of $\rho^\times$, denoted by $G^\times(\rho^\times)$, is equivalent to the QoC return of its induced path.

B. Lexicographically Optimal Policies for Product MDPs

Pure memoryless policies suffice for safety, Büchi [2], and QoC objectives [1]. However, mixing might be necessary to obtain a lexicographically near-optimal policy defined in (7), as illustrated in Example 1. Hereafter, we focus on memoryless, but possibly mixed, policies for the product MDP as their induced policies in the original MDP suffice for lexicographic near-optimality as stated in the lemma.

Lemma 1. *For a given MDP M, a safety automaton $\mathcal{A}_\psi$, an LDBA $\mathcal{A}_\varphi$ and their product MDP $\mathcal{M}^\times$, let $\Pi^{R\nu\times}_{\psi,\varphi}$, $\Pi^\times_{\psi,\varphi}$ and $\Pi^\times_\psi$ be the sets of policies defined as follows:*

$$\Pi^{R\nu\times}_{\psi,\varphi} := \left\{\pi^\times \in \Pi^\times_{\psi,\varphi} \middle| v^{R\times}_{\psi,\varphi} - \mathbb{E}_{\rho^\times \sim \mathcal{M}^\times_{\pi^\times}}\left[G^\times(\rho^\times)\right] \leq \nu\right\}, \quad (8)$$

$$\Pi^\times_{\psi,\varphi} := \underset{\pi^\times \in \Pi^\times_\psi}{\operatorname{argmax}} Pr_{\rho^\times \sim \mathcal{M}^\times_{\pi^\times}}\left\{\rho^\times \mid \rho^\times \models \Box\Diamond B^\times_\varphi\right\}, \quad (9)$$

$$\Pi^\times_\psi := \underset{\pi^\times \in \Pi^\times}{\operatorname{argmax}} Pr_{\rho^\times \sim \mathcal{M}^\times_{\pi^\times}}\left\{\rho^\times \mid \rho^\times \models \Box B^\times_\psi\right\}; \quad (10)$$

here, $\Pi^\times$ denotes the set of all memoryless policies for $\mathcal{M}^\times$ and $v^{R\times}_{\psi,\varphi} := \sup_{\pi^\times \in \Pi^\times_{\psi,\varphi}} \mathbb{E}_{\rho^\times \sim \mathcal{M}^\times_{\pi^\times}}[G^\times(\rho^\times)]$. Then $\Pi^{R\nu\times}_{\psi,\varphi}$ is non-empty and any policy in $\Pi^{R\nu\times}_{\psi,\varphi}$ induces a policy for $\mathcal{M}$ belonging to the set $\Pi^{R\nu}_{\psi,\varphi}$ defined in (7)*.*

Before proving this lemma, we first introduce notation for satisfaction probabilities. We write $Pr_{\pi^\times}(s^\times \models \phi)$ for the probability of satisfying ϕ after visiting $s^\times$; i.e.,

$$Pr_{\pi^\times}(s^\times \models \phi) := Pr_{\rho^\times_{s^\times} \sim \mathcal{M}^\times_{\pi^\times, s^\times}}\left\{\rho^\times_{s^\times} \mid \rho^\times_{s^\times} \models \phi\right\}, \quad (11)$$

where $\rho^\times_{s^\times} \sim \mathcal{M}^\times_{\pi^\times, s}$ denotes the random path drawn from the $\mathcal{M}^\times_{\pi^\times, s^\times}$, which is obtained from $\mathcal{M}^\times_{\pi^\times}$ by changing its initial state to $s^\times$. Similarly, we denote the satisfaction probability after taking $a^\times$ in $s^\times$ by $Pr_{\pi^\times}((s^\times, a^\times) \models \phi)$, which can be described as

$$Pr_{\pi^\times}\big((s^\times, a^\times) \models \phi\big) := \sum_{s^{\times\prime} \in S^\times} P^\times(s^\times, a^\times, s^{\times\prime}) Pr_{\pi^\times}(s^{\times\prime} \models \phi), \quad (12)$$

and we denote the maximum satisfaction probabilities by

$$Pr_{\max}\big((s^\times, a^\times) \models \phi\big) := \max_{\pi^\times} Pr_{\pi^\times}\big((s^\times, a^\times) \models \phi\big). \quad (13)$$

Proof. Let $A^\times_\psi(s^\times)$ denote the action set maximizing satisfaction probabilities for the safety condition in $s^\times$ defined as

$$A^\times_\psi(s^\times) := \underset{a^\times \in A^\times(s^\times)}{\operatorname{argmax}} Pr_{\max}\big((s^\times, a^\times) \models \Box B^\times_\psi\big). \quad (14)$$

A policy $\pi^\times$ maximizes the probability of satisfying the safety condition (i.e., $\pi^\times \in \Pi^\times_\psi$) if and only if it does not choose an action $a^\times \notin A^\times_\psi(s^\times)$ (this can be shown using the techniques provided for the reachability probabilities in [2]); thus, all the actions not in $A^\times_\psi(s^\times)$ can be pruned. We now can define similar action sets for the Büchi condition as

$$A^\times_{\psi,\varphi}(s^\times) := \underset{a^\times \in A^\times_\psi(s^\times)}{\operatorname{argmax}} Pr_{\max}\big((s^\times, a^\times) \models \Box\Diamond B^\times_\varphi\big), \quad (15)$$

where the maximum probabilities are obtained for the pruned MDP. Choosing from $A^\times_{\psi,\varphi}(s^\times)$ in state $s^\times$, similar to the safety condition, is necessary but not sufficient for the Büchi condition; the actions should eventually lead to a state in $B^\times_\varphi$ whenever it is possible. This can be overcome by following a policy $\pi^\times$ choosing any action in $A^\times_{\psi,\varphi}(s^\times)$ with some positive probability. Under $\pi^\times$, a state in $B^\times_\varphi$ is eventually reached and visited infinitely many times (i.e., $\pi^\times \in \Pi^\times_{\psi,\varphi}$).

We can now focus on the MDP obtained by pruning all the actions that do not belong $A^\times_{\psi,\varphi}(s^\times)$ in any state $s^\times$. Let $\pi^\times_u$ denote a policy that uniformly chooses a random action and $\pi^\times_o$ be an optimal policy maximizing the expected QoC return in this pruned MDP. As previously discussed, a policy $\pi^\times_\upsilon$ that follows $\pi^\times_u$ w.p. υ and $\pi^\times_o$ w.p. $1-\upsilon$ belongs to $\Pi^\times_{\psi,\varphi}$. As υ goes to zero, the expected QoC return under $\pi^\times_\upsilon$ approaches the maximum expected QoC return that can be obtained in the pruned MDP. Thus, there exists a sufficiently small υ such that $\pi^\times_\upsilon$ belongs to $\Pi^{R\nu\times}_{\psi,\varphi}$. □

C. Reduction from Safety and Büchi Conditions to Rewards

We provide a reduction from the safety and Büchi acceptance conditions to the rewards introduced in [15] to enable model-free RL to learn lexicographically optimal policies. We note that these crafted reward are independent from the QoC rewards that are native to the original MDP. The idea behind the reduction is to provide small rewards whenever an acceptance state is visited in order to encourage the repeated visits, and discount in a way that ensures the values approach the satisfaction probabilities as the rewards provided goes to zero. Since the reduction requires state-based discounting, we extend the definition of the return in this context as

$$G^\times_{R^\times,\Gamma^\times}(\rho^\times) := \sum_{t=0}^{\infty}\left(\prod_{i=0}^{t}\Gamma^\times(\rho^\times[i])\right) R^\times(\rho^\times[t]), \quad (16)$$

where $R^\times$ and $\Gamma^\times$ are given reward and discount functions.

We further define the near-optimal action set in a state $s^\times$ for given $R^\times$, $\Gamma^\times$ and $\tau > 0$ as

$$A^\times_{R^\times,\Gamma^\times,\tau}(s^\times) := \Big\{a^\times \in A^\times(s^\times) \mid \underset{\mathfrak{a}^\times \in A^\times(s^\times)}{\operatorname{argmax}} q^\times_{\max,R^\times,\Gamma^\times}(s^\times, \mathfrak{a}^\times) - q^\times_{\max,R^\times,\Gamma^\times}(s^\times, a^\times) \leq \tau\Big\} \quad (17)$$

where $q^\times_{\max,R^\times,\Gamma^\times}(s^\times, a^\times)$ denotes the maximum expected return for $R^\times$ and $\Gamma^\times$ obtained after taking $a^\times$ in $s^\times$.

The following lemma provides a way to obtain the optimal action sets for the safety condition via learning the near-optimal actions for the reward and discount functions crafted for the condition.

Lemma 2. *There exist sufficiently small $r_\psi > 0$ and $\tau_\psi > 0$ such that for all $s^\times \in S^\times$,*

$$A^\times_\psi(s^\times) = \left\{a^\times \in A^\times(s^\times) \;\middle|\; v^\times_\psi(s^\times) - q^\times_\psi(s^\times, a^\times) \leq \tau_\psi\right\};$$

here $A^\times_\psi(s^\times)$ is defined in (14)*, $q^\times_\psi(s^\times, a^\times)$ denotes the maximum expected return that can be obtained by taking $a^\times$ in $s^\times$ for $\Gamma^\times_\psi(s^\times) := 1 - r_\psi$ and*

$$R^\times_\psi(s^\times) := \begin{cases} r_\psi & \text{if } s^\times \in B^\times_\psi, \\ 0 & \text{if } s^\times \notin B^\times_\psi; \end{cases}$$

finally, $v^\times_\psi(s^\times) := \max_{\mathfrak{a}^\times \in A^\times(s^\times)} q^\times_\psi(s^\times, \mathfrak{a}^\times)$.

Proof. Any safety condition can be considered as a Büchi condition and therefore the proof immediately follows from Theorem 1 in [15]. The idea is that as r_ψ goes to zero, the safety return of a path approaches one if it stays in safe states and zero if it reaches an unsafe state; hence, the maximum expected safety return approaches to the maximum satisfaction probability. Since the MDP is finite, for sufficiently small $\tau_\psi > 0$ there must exist $r_\psi > 0$ such that $A^\times_\psi(s^\times)$ and the near-optimal action sets are equivalent. □

We now establish a similar result for the Büchi condition.

Lemma 3. *There exist sufficiently small $r_\varphi > 0$ and $\tau_\varphi > 0$ such that for all $s^\times \in S^\times$, $A^\times_{\psi,\varphi}(s^\times) = \{a^\times \in A^\times_\psi(s^\times) \mid v^\times_{\psi,\varphi}(s^\times) - q^\times_{\psi,\varphi}(s^\times, a^\times) \leq \tau_\varphi\}$; here $A^\times_{\psi,\varphi}(s^\times)$ is from* (15)*; $q^\times_{\psi,\varphi}(s^\times, a^\times)$ denotes the maximum expected return that can be obtained by taking $a^\times$ in $s^\times$ for*

$$R^{\times}_{\varphi}(s^{\times}):=\begin{cases} r_{\varphi} & \text{if } s^{\times}\in B^{\times}_{\varphi},\\ 0 & \text{if } s^{\times}\notin B^{\times}_{\varphi},\end{cases}\quad \Gamma^{\times}_{\varphi}(s^{\times}):=\begin{cases} 1-r_{\varphi} & \text{if } s^{\times}\in B^{\times}_{\varphi},\\ 1-r^2_{\varphi} & \text{if } s^{\times}\notin B^{\times}_{\varphi};\end{cases}$$

in the MDP where all the actions not in $A^{\times}_{\psi}(s^{\times\prime})$ are pruned in each $s^{\times\prime}$; and $v^{\times}_{\psi,\varphi}(s^{\times}) \coloneqq \max_{\mathfrak{a}^{\times}\in A^{\times}(s^{\times})} q^{\times}_{\psi,\varphi}(s^{\times},\mathfrak{a}^{\times})$.

Proof. The proof follows from Theorem 1 in [15]. Here, we provide the key idea. Unlike the safety condition, the Büchi condition requires repeatedly visiting some states in $B^{\times}_{\varphi}$ and the frequency of the visits is not important. To capture that, the rewards are discounted less in the states that do not belong to $B^{\times}_{\varphi}$; as r_{φ} goes to zero, the discounting due to visiting these states vanishes. In the limit, the Büchi return of a path is 1 only if the path visits some states in $B^{\times}_{\varphi}$ infinitely many times, and 0 otherwise. □

D. Model-Free Learning Algorithm

We now unify these ideas into a single RL algorithm. For any state $s^{\times}$, our algorithm simultaneously learns the optimal action set for the safety condition $A^{\times}_{\psi}(s^{\times})$, and among those actions learns the ones optimal for the Büchi condition $A^{\times}_{\psi,\varphi}(s^{\times})$. Additionally, it learns an optimal υ-greedy policy that chooses the best action among $A^{\times}_{\psi,\varphi}(s^{\times})$ in $s^{\times}$ for the QoC objective w.p. $1-\upsilon$ and uniformly chooses a random action in $A^{\times}_{\psi,\varphi}(s^{\times})$ w.p. υ.

Our algorithm uses Q-learning to obtain the sets $A^{\times}_{\psi}(s^{\times})$ and $A^{\times}_{\psi,\varphi}(s^{\times})$ for each state $s^{\times}$, and SARSA to obtain a near-optimal policy belonging to the set $\Pi^{R\nu\times}_{\psi,\varphi}$. Both Q-learning and SARSA learn an estimate of the value, which is the expected QoC return obtained after taking $a^{\times}$ in $s^{\times}$, denoted by $Q(s^{\times},a^{\times})$ [1]. The update of these estimates in our approach is shown in Algorithm 2. After observing a transition $(s^{\times},a^{\times},r,s^{\times\prime},a^{\times\prime})$, SARSA updates $Q^{R}_{\psi,\varphi}(s^{\times},a^{\times})$ using $r+\gamma Q^{R}_{\psi,\varphi}(s^{\times\prime},a^{\times\prime})$ as the target value. Q-learning, on the other hand, updates $Q_{\psi}(s^{\times},a^{\times})$ and $Q_{\psi,\varphi}(s^{\times},a^{\times})$ using r and the maximum value estimate in the next state $s^{\times\prime}$, independently from the action taken.

Under regular conditions on the step size α and a proper exploration policy visiting every state action pair large number of times, Q-learning converges to the optimal values and SARSA converges to the values of the policy being followed [1]. In our approach, an ϵ-greedy policy where ϵ gradually decreased to zero is used as the exploration policy. The greedy action to be taken is chosen from the estimate $\hat{A}^{\times}_{\psi,\varphi}(s^{\times})$, which is obtained using $Q_{\psi}(s^{\times},a^{\times})$ and $Q_{\psi,\varphi}(s^{\times},a^{\times})$ as described in Algorithm 3. Lastly, an υ-greedy policy decides if a random action or a greedy one for the QoC objective will taken from $\hat{A}^{\times}_{\psi,\varphi}(s^{\times})$.

We now show that our overall model-free RL approach summarized in Algorithm 1 learns a lexicographically near-optimal policy assuming that the step size α and the exploration parameter ϵ are properly decreasing to 0.

Theorem 1. *Consider an MDP $\mathcal{M}$, a safety specification ψ, an LTL specification φ, and a near-optimality parameter ν. Algorithm 1 converges to a policy $\pi^{\times}$ for the product MDP $\mathcal{M}^{\times}$, inducing a policy π for $\mathcal{M}$ belonging to the set $\Pi^{R\nu}_{\psi,\varphi}$ from* (7)*, for sufficiently small positive $r_{\psi},r_{\varphi},\tau_{\psi},\tau_{\varphi}$ and υ.*

Proof. Due to Lemma 2, Lemma 3 and the convergence of Q-learning, the estimate $\hat{A}^{\times}_{\psi}(s^{\times})$ and thereby $\hat{A}^{\times}_{\psi,\varphi}(s^{\times})$ converge to $A^{\times}_{\psi}(s^{\times})$ and $A^{\times}_{\psi,\varphi}(s^{\times})$, respectively for sufficiently small positive $r_{\psi},r_{\varphi},\tau_{\psi},\tau_{\varphi}$. Similarly, due to Lemma 1, there exists a $\upsilon>0$ ensuring the near-optimality. □

IV. Case Study

We implemented Algorithm 1 on top of *CSRL* [23], [24]. We set the discount factor γ=0.99, safety reward r_{ψ}=0.0001, and the LTL reward r_{φ}=0.01. In addition, we initialized the learning rate α, safety threshold τ_{ψ}, LTL threshold τ_{φ}, and the parameter υ for the LTL objective to 0.5, and gradually decreased them to 0.05. Similarly, we slowly decreased the exploration parameter ϵ from 0.5 to 0.005.

We conducted our numerical experiments on the 5×6 grid world shown in Fig. 2, where each cell represents a state. The agent starts in the cell $(0,0)$ moves from one cell to one of its neighbors via four actions: *up*, *down*, *right* and *left*. When an action is taken, the agent moves in the intended direction w.p. 0.8; and moves sideways w.p. 0.2 (0.1 for each direction). If the direction is blocked by an obstacle or the boundary of the grid, the agent stays in the same cell. Lastly, if the agent moves into an absorbing state it cannot leave.

The safety specification is $\psi=\square\neg(d\wedge\bigcirc d)$, which means, in the context of this grid, the cell $(0,3)$ should not be visited consecutively. The LTL specification is $\varphi=\square\lozenge b\wedge\lozenge\square c$, which can be interpreted as the right part of the grid must be eventually reached and must not be left after some time steps, and the cell $(2,5)$ must be repeatedly visited. Both formulas are translated to a 3-state automaton. All the rewards are zero except for the reward of the cell $(0,5)$, which is 1.

There are two ways to reach the right part of the grid to satisfy φ: (i) through the cell labeled with d ($(0,3)$) and (ii) through the cells between the absorbing states ($(3,2)$, $(3,3)$). Through (i), the right part can be reached w.p. 1 ($Pr_{\max}(s_0\models\varphi)$=1); however, the safety property ψ can be violated w.p. 0.2 ($Pr_{\max}(s_0\models\psi)$=0.8), due to the probability that the agent goes sideways and hit the boundary or the obstacle and stays in $(0,3)$ in two consecutive time steps. Through (ii), ψ is guaranteed to be satisfied ($Pr_{\max}(s_0\models\psi)=1$) although the probability that the agent gets stuck in one of the absorbing states is 0.36 ($Pr_{\max}(s_0\models\varphi)=0.64$). Thus, a policy prioritizing the safety chooses (ii) over (i) while a policy with a single objective $\varphi'=\psi\wedge\varphi$ prefers (i) because $Pr_{\max}(s_0\models\varphi')=0.8$ through (i) and $Pr_{\max}(s_0\models\varphi')=0.64$ through (ii).

The policy shown in Fig. 2 is learned after $128K$ ($K=2^{10}$) episodes, each with a length of $1K$. The agent learned to eliminate the actions other than *left* in $(0,2)$ and the ones other than *right* in $(0,4)$ for satisfaction of ψ. Similarly, except for $(3,2)$ and $(3,3)$, the agent prunes all the actions that might lead to an absorbing state due to φ. Since the only positive reward can be obtained in $(0,5)$, the policy aims to reach $(0,5)$ as soon as possible using the remaining actions. Once reached, the policy takes the action ε_2 to change the LDBA state to satisfy the Büchi condition. In addition, it takes a random action w.p. $\upsilon=0.05$ in $(0,5)$, $(1,5)$, $(2,5)$, $(3,5)$, $(4,5)$ and $(1,4)$, which ensures visiting $(2,5)$, the cell labeled with b and c, infinitely many times. The maximum

Algorithm 1: Model-free RL for lexicographical safety, LTL and QoC objectives

Input: Safety ψ, LTL φ, MDP $\mathcal{M}$
Translate ψ and φ to $\mathcal{A}_\psi$ and $\mathcal{A}_\varphi$
Construct $\mathcal{M}^\times$ of $\mathcal{M}$, $\mathcal{A}_\psi$ and $\mathcal{A}_\varphi$
Initialize Q_ψ, $Q_{\psi,\varphi}$ and $Q^R_{\psi,\varphi}$ on $\mathcal{M}^\times$
for $i = 0, 1, \ldots$ **do**
  for $t = 0$ **to** $\mathcal{T} - 1$ **do**
    $a_t^\times \leftarrow$get_action$(Q_\psi, Q_{\psi,\varphi}, Q^R_{\psi,\varphi}, s_t^\times)$
    Take $a_t^\times$ and observe r_t and $s_{t+1}^\times$
    if $t > 0$ **then**
      update_Qs$(Q_\psi, Q_{\psi,\varphi}, Q^R_{\psi,\varphi}, (s_{t-1}^\times, a_{t-1}^\times, r_{t-1}, s_t^\times, a_t^\times))$
    end if
  end for
end for

Algorithm 2: update_Qs

Input: $Q_\psi, Q_{\psi,\varphi}, Q^R_{\psi,\varphi}, (s^\times, a^\times, r, s^{\times\prime}, a^{\times\prime})$
if $s^{\times\prime} \in B_\psi^\times$ **then**
  $Q_\psi(s^\times, a^\times) \leftarrow (1-\alpha)Q_\psi(s^\times, a^\times)$
  $+\alpha(r_\psi + (1-r_\psi)\max_{\mathfrak{a}^\times} Q_\psi(s^{\times\prime}, \mathfrak{a}^\times))$
else
  $Q_\psi(s^\times, a^\times) \leftarrow (1-\alpha)Q_\psi(s^\times, a^\times)$
end if
if $s^{\times\prime} \in B_\varphi^\times$ **then**
  $Q_{\psi,\varphi}(s^\times, a^\times) \leftarrow (1-\alpha)Q_{\psi,\varphi}(s^\times, a^\times)$
  $+\alpha(r_\varphi + (1-r_\varphi)\max_{\mathfrak{a}^\times} Q_{\psi,\varphi}(s^{\times\prime}, \mathfrak{a}^\times))$
else
  $Q_{\psi,\varphi}(s^\times, a^\times) \leftarrow (1-\alpha)Q_{\psi,\varphi}(s^\times, a^\times)$
  $+\alpha((1-r_\varphi^2)\max_{\mathfrak{a}^\times} Q_{\psi,\varphi}(s^{\times\prime}, \mathfrak{a}^\times))$
end if
$Q^R_{\psi,\varphi}(s^\times, a^\times) \leftarrow (1-\alpha)Q^R_{\psi,\varphi}(s^\times, a^\times)$
$+\alpha(r+\gamma Q^R_{\psi,\varphi}(s^{\times\prime}, a^{\times\prime}))$

Algorithm 3: choose_action

Input: $Q_\psi, Q_{\psi,\varphi}, Q^R_{\psi,\varphi}$ and $s^\times$
$u \sim \mathcal{U}(0,1)$
if $u \leq \epsilon$ **then**
  return a random action $a^\times \in A^\times(s^\times)$
end if
$V_\psi(s^\times) \leftarrow \max_{a^\times \in A^\times(s^\times)} Q_\psi(s^\times, \mathfrak{a}^\times)$
$\hat{A}_\psi^\times(s^\times) \leftarrow \{a^\times \in A^\times(s^\times) \mid V_\psi(s^\times) - Q_\psi(s^\times, a^\times) \leq \tau_\psi\}$
$V_{\psi,\varphi}(s^\times) \leftarrow \max_{a^\times \in \hat{A}_\psi^\times(s^\times)} Q_{\psi,\varphi}(s^\times, \mathfrak{a}^\times)$
$\hat{A}_{\psi,\varphi}^\times(s^\times) \leftarrow \{a^\times \in \hat{A}_\psi^\times(s^\times) | V_{\psi,\varphi}(s^\times) - Q_{\psi,\varphi}(s^\times, a^\times) \leq \tau_\varphi\}$
if $u \leq \epsilon + \upsilon$ **then**
  return a random action $a^\times \in \hat{A}_{\psi,\varphi}^\times(s^\times)$
end if
return $a^\times \in \operatorname{argmax}_{\mathfrak{a}^\times \in \hat{A}_{\psi,\varphi}^\times(s^\times)} Q^R_{\varphi,\psi}(s^\times, a^\times)$

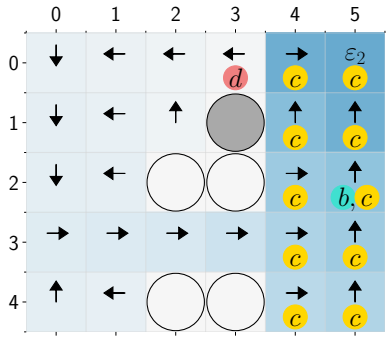

Fig. 2: The grid world with the learned policy. Empty circles: absorbing states. Filled circle: obstacle. Encircled letters: labels. Arrows: actions. Estimated values are represented by the shades of blue; the darker, the higher value.

expected QoC return that can be obtained after visiting $(0, 5)$ is about 88, and approximately a QoC return of 85 is obtained by following the learned policy.

V. Conclusion

In this paper, we have introduced a model-free RL method that learns near-optimal policies for given safety, LTL and QoC objectives, where the safety objective is prioritized above the LTL objective, which is prioritized above the QoC objective. Our algorithm learns the safest actions and among those learns the actions sufficient to maximize the probability of satisfying the LTL objective. Finally, using these actions, our algorithm converges to a policy maximizing the expected return for a given parameter υ controlling the randomness of the obtained policy. We have shown that as long as υ is positive, the satisfaction probability for the LTL objective is maximized and as υ goes to zero, the expected return approaches its supremum value. Lastly, we have demonstrated the applicability of our algorithm on a case study.

References

[1] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.

[3] H. Kress-Gazit, F. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.

[4] E. Plaku and S. Karaman. Motion planning with temporal-logic specifications: progress and challenges. *AI communications*, 29(1):151–162, 2016.

[5] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[6] M. Svoreňová and M. Kwiatkowska. Quantitative verification and strategy synthesis for stochastic games. *European Journal of Control*, 30:15–30, 2016.

[7] E. M. Hahn, V. Hashemi, H. Hermanns, M. Lahijanian, and A. Turrini. Interval Markov decision processes with multiple objectives: from robust strategies to Pareto curves. *ACM Transactions on Modeling and Computer Simulation*, 29(4):1–31, 2019.

[8] K. Chatterjee, J. Katoen, M. Weininger, and T. Winkler. Stochastic games with lexicographic reachability-safety objectives. In *International Conference on Computer Aided Verification (CAV)*, pages 398–420, 2020.

[9] K. C. Kalagarla, R. Jain, and P. Nuzzo. Synthesis of discounted-reward optimal policies for Markov decision processes under linear temporal logic specifications. *arXiv:2011.00632*, 2020.

[10] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. In *AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[11] G. Avni, R. Bloem, K. Chatterjee, T. A. Henzinger, B. Könighofer, and S. Pranger. Run-time optimization for learned controllers through quantitative games. In *International Conference on Computer Aided Verification (CAV)*, pages 630–649, 2019.

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

[13] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 2107–2116, 2018.

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

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

[16] X. Li and C. Belta. Temporal logic guided safe reinforcement learning using control barrier functions. *arXiv:1903.09885*, 2019.

[17] D. Aksaray, Y. Yazicioglu, and A. S. Asarkaya. Probabilistically guaranteed satisfaction of temporal logic constraints during reinforcement learning. *arXiv:2102.10063*, 2021.

[18] J. Křetínskỳ, G. A. Pérez, and J. Raskin. Learning-based mean-payoff optimization in an unknown MDP under omega-regular constraints. In *International Conference on Concurrency Theory*, 2018.

[19] L. Hammond, A. Abate, J. Gutierrez, and M. Wooldridge. Multi-agent reinforcement learning with temporal logic specifications. In *International Conference on Autonomous Agents and MultiAgent Systems*, pages 583–592, 2021.

[20] S. Sickert, J. Esparza, S. Jaax, and J. Křetínskỳ. Limit-deterministic Büchi automata for linear temporal logic. In *International Conference on Computer Aided Verification (CAV)*, pages 312–332, 2016.

[21] O. Kupferman and M. Y. Vardi. Model checking of safety properties. *Formal Methods in System Design*, 19(3):291–314, 2001.

[22] Z. Gábor, Z. Kalmár, and C. Szepesvári. Multi-criteria reinforcement learning. In *Int. Conf. on Machine Learning*, pages 197–205, 1998.

[23] A. K. Bozkurt, Y. Wang, M. M. Zavlanos, and M. Pajic. Model-free reinforcement learning for stochastic games with linear temporal logic objectives. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[24] A. K. Bozkurt, Y. Wang, and M. Pajic. Learning optimal strategies for temporal tasks in stochastic games. *arXiv:2102.04307*, 2021.