2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Synthesis of Temporally-Robust Policies for Signal Temporal Logic Tasks using Reinforcement Learning

Siqi Wang, Shaoyuan Li, Li Yin, Xiang Yin

***Abstract*— This paper investigates the problem of designing control policies that satisfy high-level specifications described by signal temporal logic (STL) in unknown, stochastic environments. While many existing works concentrate on optimizing the spatial robustness of a system, our work takes a step further by also considering *temporal robustness* as a critical metric to quantify the tolerance of time uncertainty in STL. To this end, we formulate two relevant control objectives to enhance the temporal robustness of the synthesized policies. The first objective is to maximize the probability of being temporally robust for a given threshold. The second objective is to maximize the worst-case spatial robustness value within a bounded time shift. We use reinforcement learning to solve both control synthesis problems for unknown systems. Specifically, we approximate both control objectives in a way that enables us to apply the standard Q-learning algorithm. Theoretical bounds in terms of the approximations are also derived. We present case studies to demonstrate the feasibility of our approach.**

I. Introduction

Autonomous systems operating in dynamic environments face the challenge of making complex real-time decisions. These systems, known as *time-critical systems*, must process real-time information to achieve their goals, and the accuracy of their decisions is crucial, especially with temporal constraints involved. For example, an automated guided vehicle may need to retrieve a workpiece within 10 minutes and return it within 20 minutes. However, an ad-hoc approach for real-time decision-making may lead to errors. Consequently, ensuring formal guarantees for real-time systems has gained focus in recent years.

Signal temporal logic (STL) is a formal specification language used to describe high-level temporal behaviors of continuous signals. It extends metric temporal logic for real-time systems by incorporating real-valued predicates on signals [1]–[3]. One major advantage of STL is its ability to provide both Boolean satisfaction and quantitative measures, termed spatial robustness degree. This unique feature has led to the spreading use of STL in cyber-physical systems, including autonomous robots [4], process control systems [5], smart cities [6] and self-driving vehicles [7].

The spatial robustness essentially quantifies the satisfaction of STL tasks based on value changes in the predicate function. However, practical scenarios also involve signal delays or ahead-of-time occurrences during online executions, which necessitates exploring STL satisfaction under signal time uncertainty. To tackle this, *temporal robustness* was introduced in the literature, which quantifies the maximum left or right time shift a signal trajectory can endure to maintain satisfaction or violation of an STL specification [8]. In [9], the authors merged left and right temporal robustness and solved control synthesis. In [10]–[12], the temporal robustness was further investigated for synchronization issues of multi-dimensional signals. Additionally, there are other measures for quantifying the robust satisfaction of STL formulae under time uncertainty, such as conformance [13] or AverageSTL robustness [14], [15].

In open or reactive environments, synthesizing control sequences or policies to ensure the satisfaction of STL tasks is a major challenge. To tackle this, various synthesis methods have emerged, including encoding the STL satisfaction as constraints in a mixed-integer linear program (MILP) [16]–[19] and capturing the satisfaction regions of the STL formula using control barrier functions (CBFs) [20]–[22]. These approaches assume system knowledge; yet in many applications, the system's dynamic is unknown, and trajectories can only be generated through interactions. Thus, reinforcement learning techniques are employed for STL task control synthesis; see, e.g., [23]–[27]. In [23], Q-learning was applied to maximize the expected robust degree for an unknown system modeled by Markov decision processes (MDPs). Moreover, [26] offered more efficient MDP construction methods.

The focus of the aforementioned work is to enhance spatial robustness of control policies. However, in the context of temporal robustness for time-uncertain systems, synthesis methods are only available in recent works such as [9], [10], [28], [29]. These approaches have a common feature, which is to extend the MILP-based approach by encoding the temporal robustness using new variables. Similar to the issue of spatial robustness, these approaches rely on the system model. To the best of our knowledge, optimizing STL task temporal robustness in systems with unknown dynamics remains unexplored.

In this paper, we address the challenge of control policy synthesis for unknown stochastic systems to achieve STL tasks. Unlike previous work focusing on spatial robustness, we extend our focus to include temporal robustness. Specifically, we tackle two control synthesis problems. Firstly, we aim to synthesize a policy that maximizes the probability of the trajectory's temporal robustness exceeding

This work was supported by the National Natural Science Foundation of China (62173226,62061136004).

Siqi Wang, Shaoyuan Li, and Xiang Yin are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. Li Yin is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macao SAR, China. E-mail: {sq_wang, syli, yinxiang}@sjtu.edu.cn

a set threshold. Secondly, we aim to maximize the expected worst-case spatial robustness value of the system's trajectories under time uncertainty. To tackle both problems, we apply reinforcement learning techniques, specifically Q-learning method, for MDPs. Our approach draws inspiration from [23], who uses a τ-MDP structure for reinforcement learning for spatial robustness. Here we reformulate the two problems and select appropriate augmented horizons to handle temporal robustness metrics. We also establish a formal connection between the original problems and the reformulated problems. Our experimental results show that the synthesized policies can effectively enhance the temporal robustness of the system.

II. Preliminaries

This section reviews some basic concepts of signal temporal logic and reinforcement learning of unknown MDPs.

A. Signal Temporal Logic Basics

Signal temporal logic (STL) is a formal language used for specifying temporal properties for real-time systems. It is evaluated over dense-time signals in continuous metric space $\mathbb{R}^m$. The syntax of STL is recursively defined as follows [1]:

$$\phi ::= \mathsf{true} \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathbf{U}_{[a,b)} \phi_2, \tag{1}$$

where $\mu : \mathbb{R}^m \to \{\mathsf{true}, \mathsf{false}\}$ is an atomic predicate such that it is satisfied when the value of the associated predicate function $h^\mu(s) > 0$, where $s \in \mathbb{R}^m$; $\neg$ and $\wedge$ are the standard Boolean operators, negation and conjunction respectively, and $\mathbf{U}_{[a,b)}$ is the temporal operator "until" with $a < b$ and $a, b \in \mathbb{N}$. Furthermore, one can induce temporal operators:

- "eventually" by $\mathbf{F}_{[a,b)}\phi := \mathsf{true}\mathbf{U}_{[a,b)}\phi$; and
- "always" by $\mathbf{G}_{[a,b)}\phi := \neg\mathbf{F}_{[a,b)}\neg\phi$.

Definition 1 (Spatial Robustness of STL): Let ϕ be an STL formula, $\mathbf{s} = s_0 s_1 \cdots$ be a signal and $t \in \mathbb{N}$ be a time instant. The *spatial robustness* of ϕ w.r.t. $\mathbf{s}$ at time t, denoted by $\rho(\phi, \mathbf{s}, t)$, is defined recursively by

$$\begin{array}{lcl} \rho(\mathsf{true}, \mathbf{s}, t) & = & +\infty, \\ \rho(\mu, \mathbf{s}, t) & = & h^\mu(s_t), \\ \rho(\neg\phi, \mathbf{s}, t) & = & -\rho(\phi, \mathbf{s}, t), \\ \rho(\phi_1 \wedge \phi_2, \mathbf{s}, t) & = & \min\left(\rho(\phi_1, s, t), \rho(\phi_2, s, t)\right), \\ \rho(\phi_1 \mathbf{U}_{[a,b)} \phi_2, \mathbf{s}, t) & = & \max_{t' \in [a+t, b+t)} \min\{\rho(\phi_2, \mathbf{s}, t'), \\ & & \min_{t'' \in [t, t')} \rho(\phi_1, \mathbf{s}, t'')\}. \end{array} \tag{2}$$

The Boolean semantic of STL is a special instance of the spatial robustness semantic. Let ϕ be an STL formula, $\mathbf{s} = s_0 s_1 \cdots$ be a signal and $t \in \mathbb{N}$ be a time instant. We say ϕ is satisfied by $\mathbf{s}$ at t , denoted by $\mathbf{s}[t] \models \phi$ if its robust value is larger than zero, i.e.,

$$\mathbf{s}[t] \models \phi \Leftrightarrow \rho(\phi, \mathbf{s}, t) > 0. \tag{3}$$

We also define the *characteristic function* of the STL formula ϕ w.r.t. signal $\mathbf{s}$ at time instant t by

$$\mathcal{X}(\phi, \mathbf{s}, t) = \begin{cases} 1 & \text{if} \quad \mathbf{s}[t] \models \phi \\ -1 & \text{otherwise} \end{cases}. \tag{4}$$

For any STL formula, its satisfaction as well as the robust degree can be completely determined within its *horizon* denoted by $\mathrm{hrz}(\phi)$, which can be computed as the maximum sum of the time interval bound of all nested temporal operators; see, e.g., [23].

Hereafter in this work, we will restrict our attention to the following fragment of STL formulae:

$$\Phi ::= \mathbf{F}_{[0,H)}\phi \mid \mathbf{G}_{[0,H)}\phi, \tag{5}$$

where ϕ is a general STL formula in Equation (1) and $H \in \mathbb{N}$ is a time instant.

B. Temporal Robustness of STL

The spatial robustness semantic quantifies the extent of STL satisfaction based on predicate function value changes. However, it cannot capture the robust satisfaction of a formula concerning time shifts. To address this, *temporal robustness* was introduced in [8] to quantify the maximum left or right time shift a signal trajectory can bear to maintain satisfaction or violation of an STL specification.

Definition 2 (Temporal Robustness of STL, [8]): Let ϕ be an STL formula, $\mathbf{s} = s_0 s_1 \cdots$ be a signal and $t \in \mathbb{N}$ be a time instant. The left (respectively, right) temporal robustness, denoted by $\theta^-(\phi, \mathbf{s}, t)$ (respectively, $\theta^+(\phi, \mathbf{s}, t)$), is defined as the maximum left (respectively, right) time shift for a signal $\mathbf{s}$ to maintain the satisfaction or the violation of STL formula ϕ. Formally, we have

$$\begin{array}{ll} \theta^+(\phi, \mathbf{s}, t) & = \mathcal{X}(\phi, \mathbf{s}, t) \cdot \max\left\{ d \;\middle|\; \begin{array}{c} \forall t' \in [t, t+d] \\ \mathcal{X}(\phi, \mathbf{s}, t') = \mathcal{X}(\phi, \mathbf{s}, t) \end{array} \right\} \\ \theta^-(\phi, \mathbf{s}, t) & = \mathcal{X}(\phi, \mathbf{s}, t) \cdot \max\left\{ d \;\middle|\; \begin{array}{c} \forall t' \in [t-d, t] \\ \mathcal{X}(\phi, \mathbf{s}, t') = \mathcal{X}(\phi, \mathbf{s}, t) \end{array} \right\} \end{array} \tag{6}$$

Remark 1: In the provided definition, when signal shifts result in undefined states, we fill these states with the initial or final values. For example, for all $t < 0$, we have $s_t := s_0$.

Remark 2: For simplicity, we only consider left robustness here, given that right shifts or time-delays are more common in real-world scenarios. Hereafter, we will simply use the terminology "temporal robustness" denoted as $\theta(\cdot)$, to stand for left temporal robustness $\theta^-(\cdot)$.

C. Reinforcement Learning for MDPs

We model the underlying dynamic system by a Markov decision process. Formally, an MDP is a tuple $M = (\Sigma, s_0, A, P, R)$, where Σ is the state space, s_0 is the initial state, and A is the action space, $P : \Sigma \times A \times \Sigma \to [0, 1]$ is a transition probability function and $R : \Sigma \to \mathbb{R}$ is the reward function. We assume that the state space and action space are known, but the transition probability is *unknown*. For simplicity, we assume the initial state is unique; however, this can easily be extended to the scenario with an initial distribution.

Given MDP M, a (stationary) control policy is a function $\pi : \Sigma \times A \to [0, 1]$ which assigns probabilities to actions at each state such that for all $s \in \Sigma$, $\sum_{a \in A} \pi(s, a) = 1$. The objective of reinforcement learning is to synthesize a control policy that maximizes the total sum of discounted rewards through simulation data [30], i.e.,

$$\pi^* = \arg\max_{\pi} \mathbb{E} \sum_{t=0}^{T} \gamma^t R_t, \tag{7}$$

where R_t is the random variable for the reward at instant t when the agent applies policy π and $\gamma \in [0,1]$ is a discount factor that balances future rewards.

Q-learning is a prominent algorithm in reinforcement learning to achieve the above objective without knowing the transition probability. It is a model-free, off-policy and temporal-difference method utilizing a Q-table to hold values for state-action pairs. At each instant t, with α being the learning rate, we use the one-step transition data (s_t, a_t, r_t, s_{t+1}) to update the Q-table according to the Bellman Equation as follows:

$$Q(s_t, a_t) := (1-\alpha)Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a \in A} Q(s_{t+1}, a)]. \quad (8)$$

III. Problem Formulation

We define two optimality metrics for the synthesized control policy by taking the temporal robustness value into account and formulate the control synthesis problems.

A. Case of Guaranteed Temporal Robustness

In the first approach, we consider $\delta > 0$ as the minimum required value for the temporal robustness of the signal. Only signals with temporal robustness greater than or equal to δ are considered "robust-enough" signals. The following first problem aims to maximize the probability of robust-enough signals.

Problem 1: (Maximizing Probability with Temporal Robustness Guarantees): Let M be an MDP with unknown P and Φ be an STL formula in form of Equation (5) with time horizon $\text{hrz}(\Phi) = T+1$. Given δ, find a policy π_1^* that maximizes the probability of generating "robust-enough" trajectories. That is

$$\pi_1^* = \arg\max_{\pi} \Pr^{\pi}[\theta(\Phi, \mathbf{s}_{0:T}) \geq \delta], \quad (9)$$

where $\Pr^{\pi}(\cdot)$ denotes the probability under π.

Note that, using indicator function $I(\cdot)$, Equation (9) can be expressed equivalently as

$$\pi_1^* = \arg\max_{\pi} \mathbb{E}^{\pi}[I(\theta(\Phi, \mathbf{s}_{0:T}) \geq \delta)]. \quad (10)$$

B. Case of Spatial-Temporal Robustness

Note that, the above problem formulation does not consider spatial details within signal values. To address this, we extend it to a spatial-temporal robustness joint optimization problem. In this formulation, our goal is twofold: maximizing the satisfaction probability under time uncertainty and enhancing the extent of satisfaction.

Specifically, we consider $\delta > 0$ as an upper bound of possible time shifts. Then we define the worst-case spatial robustness value with time shifts bounded by δ by

$$\rho_{\delta}(\Phi, \mathbf{s}_{0:T}) := \min\{\rho(\Phi, \mathbf{s}_{0:T}, d) \mid d \leq \delta\}. \quad (11)$$

Our objective is to maximize the expectation of such worst-case spatial robustness value.

Problem 2: (Maximizing Expectation of Spatio-Temporal Robustness): Under the same setting as Problem 1, find a control policy π_2^* such that

$$\pi_2^* = \arg\max_{\pi} \mathbb{E}^{\pi}\left[\rho_{\delta}(\Phi, \mathbf{s}_{0:T})\right]. \quad (12)$$

IV. Reinforcement Learning for Temporal Robustness

In this section, we reshape the previous problems into RL-friendly ones. Specifically, we approximate the original objective functions in summation-based forms for which standard Q-learning algorithms can be applied. Theoretical bounds between the original problems and their approximations are established.

A. Construction of τ-MDPs

As noted by [23], for STL spatial robustness, the standard Q-learning algorithm cannot be applied to the original MDP since the reward function is non-Markovian. The same issue also exists for temporal robustness, and time shifts further add complexity. To address this, we use the τ-MDP structure proposed in [23], which essentially augments states in the original MDP by a sequence of states of length τ.

Definition 3 (τ-MDP, [23]): Given an MDP $M = (\Sigma, s_0, A, P, R)$ and a positive integer $\tau \in \mathbb{N}$, its associated τ-MDP is a tuple $M^{\tau} = (\Sigma^{\tau}, s_0^{\tau}, A, P^{\tau}, R^{\tau})$, where

- $\Sigma^{\tau} \subseteq (\Sigma)^{\tau}$ is the set of states.
- s_0^{τ} is the initial state, which is initialized as a string $s_0 s_0 \dots s_0$ of length τ;
- A is the action space, which is the same as M;
- $P^{\tau} : \Sigma^{\tau} \times A \times \Sigma^{\tau} \to [0,1]$ is the transition probability function such that, for any $s_t^{\tau}, s_{t+1}^{\tau} \in \Sigma^{\tau}, a_t \in A$, if $s_{t+1}^{\tau}(i) = s_t^{\tau}(i+1), \forall i \in \{0, 1, \cdots, \tau-2\}$, where $s_t^{\tau}(i)$ denotes the ith element in s_t^{τ}, then we have $P^{\tau}(s_t^{\tau}, a_t, s_{t+1}^{\tau}) = P(s_t^{\tau}(\tau-1), a_t, s_{t+1}^{\tau}(\tau-1))$. Otherwise, $P^{\tau}(s_t^{\tau}, a_t, s_{t+1}^{\tau}) = 0$;
- $R^{\tau} : \Sigma^{\tau} \to \mathbb{R}$ is the reward function defined over the τ-MDP.

Intuitively, the τ-MDP "unfolds" the original MDP by retaining the last τ visited states. That is, each s_t^{τ} denotes the τ-step history, i.e., $s_t^{\tau} = \mathbf{s}_{t-\tau+1:t}$, where τ is determined by the task. For a general STL formula, we need to choose τ as the entire horizon of the formula to gather enough information for robustness computation. Yet, for formulae $\mathbf{F}_{[0,H)}\phi$ or $\mathbf{G}_{[0,H)}\phi$, we only need $\tau = \text{hrz}(\phi) + \delta$. Specifically, we need $\text{hrz}(\phi)$-step information to determine the satisfaction of the internal sub-formula ϕ, and an additional δ-step information to determine the temporal robustness.

B. Approximation of Robust Probability

Before addressing Problem 1, where our objective is to maximize the probability of being temporally robust to a time shift threshold, we first extend the STL semantic to delayed partial signals.

Definition 4: (Satisfaction of Delayed Signal): Given specification Φ, τ-state s_t^{τ}, and a small delay amount $d < \delta$, we denote the delayed (right shifted) signal trace as $\mathbf{s}_{t-h-d+1:t-d}$, where $h = \text{hrz}(\phi)$. The satisfaction of delayed signal is denoted as follows:

$$\text{sat}(\phi, s_t^{\tau}, d) = [\mathbf{s}_{t-h-d+1:t-d} \models \phi'], \quad (13)$$

where ϕ' is obtained from ϕ by right-shifting the effective time window by $t-h-d+1$ steps, e.g., for $\phi = \mathbf{G}_{[0,h)}(s <$

C), we have $\phi' = \mathbf{G}_{[t-h-d+1,t-d)}(s < C)$. In another word, the satisfaction of delayed signal is determined by whether every (or at least) one state on trace $\mathbf{s}_{t-h-d+1:t-d}$ satisfies the inner sub-formula ϕ' by Boolean semantic (3). With a slight abuse of notation, hereafter, sub-formulae are by default shifted according to the elapsed time t and thus we do not differentiate ϕ and ϕ'.

Having defined satisfaction of delayed signals, we can now formally qualify the concept of being temporally robust.

Definition 5: (δ-Temporally-Robust): Given specification Φ, τ-state s_t^τ and temporal robustness lower bound δ, we say τ-state s_t^τ is δ-temporally-robust if s_t^τ satisfies the specification Φ for any delay $d \in [0, \delta]$, and we denote

$$\text{rb}(\Phi, s_t^\tau, \delta) = \begin{cases} 1 & \text{if } \min\limits_{0\le d\le\delta} (\text{sat}(\Phi, s_t^\tau, d)) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Proposition 1: The temporal robustness guarantee of signal $\mathbf{s}_{0:T}$ w.r.t. the overall formula Φ can be determined by the temporal robustness guarantee of partial signal s_t^τ w.r.t the sub-formula ϕ, i.e., the following equation holds:

$$I(\theta(\Phi, \mathbf{s}_{0:T}) \ge \delta) = \begin{cases} \max\limits_{t=0:T} (\text{rb}(\phi, s_t^\tau, \delta)) & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ \min\limits_{t=0:T} (\text{rb}(\phi, s_t^\tau, \delta)) & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{15}$$

Proof: For disjunction and conjunction of sub-formulae, we have the following properties in terms of temporal robustness; see [28]:

$$\begin{aligned} \theta(\phi_1 \vee \phi_2, \mathbf{s}_{0:T}) &= \max(\theta(\phi_1, \mathbf{s}_{0:T}), \theta(\phi_2, \mathbf{s}_{0:T})) \\ \theta(\phi_1 \wedge \phi_2, \mathbf{s}_{0:T}) &= \min(\theta(\phi_1, \mathbf{s}_{0:T}), \theta(\phi_2, \mathbf{s}_{0:T})) \end{aligned} \tag{16}$$

For STL formula $\Phi = \mathbf{F}_{[0,H)}\phi$, we use the operator "$\vee$" to break down the formula, and since $\theta(\phi, \mathbf{s}_{t:t+h-1}) \ge \delta, \forall t \in [0, T]$, we have

$$\theta(\Phi, \mathbf{s}_{0:T}) = \max_{t=0:T}(\theta(\phi, \mathbf{s}_{t:t+h-1})). \tag{17}$$

Since $I(\max\limits_{t=0:T}(\theta(\phi, \mathbf{s}_{t:t+h-1})) \ge \delta)$ is equivalent to $\max\limits_{0:T}(\text{rb}(\phi, s_t^\tau, \delta))$, we obtain that

$$I(\theta(\Phi, \mathbf{s}_{0:T}) \ge \delta) = \max_{t=0:T}(\text{rb}(\phi, s_t^\tau, \delta)).$$

Proof is similar for $\Phi = \mathbf{G}_{[0,H)}\phi$, as $\min\limits_{t=0:T}(\text{rb}(\phi, s_t^\tau, \delta))$ is equivalent to $I(\min\limits_{t=0:T}(\theta(\phi, \mathbf{s}_{t:t+h-1})) \ge \delta)$. ∎

We have established the relation between the temporal robustness of entire trajectory w.r.t. the overall specification Φ and the temporal robustness of partial trajectory w.r.t. the inner sub-formula ϕ. In [23], the author approximated and decomposed the objective function using the LSE (log-sum-exp) method into a sum of step rewards. Here we use the similar philosophy in reformulating our problem. The LSE, also known as a smooth approximation to the maximum function, is defined as follows:

$$\max(x_1, \cdots, x_n) \approx \tfrac{1}{\beta} \log \sum_{i=1}^{n} e^{\beta x_i}. \tag{18}$$

The approximation is bounded by the following inequalities:

$$\begin{aligned} \max(x_1, \ldots, x_n) &\le \tfrac{1}{\beta} \log \sum_{i=1}^{n} e^{\beta x_i} \\ &\le \max(x_1, \ldots, x_n) + \tfrac{1}{\beta} \log n. \end{aligned} \tag{19}$$

The same can be derived for $\min(x_1, \cdots, x_n) = -\max(-x_1, \cdots, -x_n) \approx -\frac{1}{\beta} \log \sum\limits_{i=1}^{n} e^{-\beta x_i}$. Clearly, increasing β will narrow the gap between the approximated value and the actual value.

Problem 1A: (Maximizing Approximated Probability of Being Temporally Robust): Consider an MDP $M = \langle \Sigma, s_0, A, P, R \rangle$ with unknown P, given an STL specification Φ, find a control policy π_{1A}^* that maximizes the following objective function:

$$\pi_{1A}^* = \begin{cases} \arg\max\limits_{\pi} \mathbb{E}[\sum\limits_{t=0}^{T} e^{\beta \cdot \text{rb}(\phi, s_t^\tau, \delta)}] & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ \arg\max\limits_{\pi} \mathbb{E}[\sum\limits_{t=0}^{T} -e^{-\beta \cdot \text{rb}(\phi, s_t^\tau, \delta)}] & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{20}$$

The following result shows that the reformulated problem can be arbitrarily close to the original problem.

Proposition 2: When β goes to infinity, the optimal policy to the approximated problem π_{1A}^* will converge to the optimal policy to the original problem π_1^*.

Proof: The proposition can be manifested from Equation (19), which leads to:

$$\begin{aligned} &\Pr^{\pi_1^*}(\theta(\Phi, \mathbf{s}_{0:T}) \ge \delta) \le \Pr^{\pi_{1A}^*}(\theta(\Phi, \mathbf{s}_{0:T}) \ge \delta) \\ &\le \Pr^{\pi_1^*}(\theta(\Phi, \mathbf{s}_{0:T}) \ge \delta) + \tfrac{1}{\beta} \log(T+1). \end{aligned} \tag{21}$$

∎

Note that, when β is too large, the objective function will become less smooth, which prolonging the learning convergence [30]. Therefore, we select a reasonably large β in the experiments.

Now Problem 1A is in the standard form of reinforcement learning, depending on the type of the STL specification, the step reward is given as:

$$r_t^\tau = R^\tau(s_{t+1}^\tau) = \begin{cases} e^{\beta \cdot \text{rb}(\phi, s_{t+1}^\tau, \delta)} & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ -e^{-\beta \cdot \text{rb}(\phi, s_{t+1}^\tau, \delta)} & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{22}$$

C. Maximizing Spatial-Temporal Robustness

Although our original purpose is to further consider the spatial robustness in addition to the temporal robustness requirement, the purported approach is also expected to expedite the learning process. This is because relying solely on Boolean satisfaction offers little insight into the quality of the current τ-state. For example, for $\Phi = \mathbf{F}_{[0,H)}\phi$, before reaching a satisfying τ-state, the agent receives minimum reward, resulting in infrequent updates to the corresponding entries in the Q-table. Thus the learning process reduces to a Monte-Carlo search, which is far from efficient. However, incorporating spatial semantic can enhance our data efficiency. To this end, we denote the objective function in Problem 2 as J_0 and we obtain the following equivalent form regarding $\Phi = \mathbf{F}_{[0,H)}\phi$ and $\Phi = \mathbf{G}_{[0,H)}\phi$

$$J_0 = \begin{cases} \mathbb{E}[\min\limits_{d\le\delta}\{\max\limits_{t}\{\rho(\phi, s_t^\tau, d)\}\}] & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ \mathbb{E}[\min\limits_{d\le\delta}\{\min\limits_{t}\{\rho(\phi, s_t^\tau, d)\}\}] & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{23}$$

Using the same LSE technique to decompose the `max` and `min` operator in J_0, we obtain the approximated objective function J_1 by

$$J_1 = \begin{cases} \mathbb{E}[\min\limits_{d\le\delta}\{\sum\limits_{t=0}^{T} e^{\beta\rho(\phi,s_t^\tau,d)}\}] & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ \mathbb{E}[\min\limits_{d\le\delta}\{\sum\limits_{t=0}^{T} -e^{-\beta\rho(\phi,s_t^\tau,d)}\}] & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{24}$$

Note that the above objective function J_1 is still not in the additive form, to further obtain the instant reward at each step, we define the objective function J_2 by

$$J_2 = \begin{cases} \mathbb{E}[\sum\limits_{t=0}^{T} e^{\beta\cdot\min\limits_{d\le\delta}\{\rho(\phi,s_t^\tau,d)\}}] & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ \mathbb{E}[\sum\limits_{t=0}^{T} -e^{-\beta\cdot\min\limits_{d\le\delta}\{\rho(\phi,s_t^\tau,d)\}}] & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{25}$$

We use the objective function J_2 to formulate the approximated problem as follows.

Problem 2A: (Maximizing Approximated Expected Spatial-Temporal Robustness): Consider an MDP $M = \langle \Sigma, s_0, A, P, R\rangle$ with unknown P, given an STL specification Φ, find a control policy π_{2A}^* that maximize the objective function J_2.

Specifically, the step reward is given by

$$R^\tau = \begin{cases} e^{\beta\cdot\min\limits_{d\le\delta}\{\rho(\phi,s_t^\tau,d)\}} & \text{if } \Phi = \mathbf{F}_{[0,H)}\phi \\ -e^{-\beta\cdot\min\limits_{d\le\delta}\{\rho(\phi,s_t^\tau,d\}} & \text{if } \Phi = \mathbf{G}_{[0,H)}\phi \end{cases} \tag{26}$$

The following result shows that the optimal value of the approximated Problem 2A provides a lower bound for the optimal value of the original Problem 2.

Proposition 3: Maximizing the objective function of Problem 2A maximizes the objective function of Problem 2.

Proof: For $\Phi = \mathbf{F}_{[0,H)}\phi$, we denote $e^{\beta\rho(\phi,s_t^\tau,d)}$ as M_t^d, where $d \in \{0,1,\dots,\delta\}$. The approximated objective function J_1 can be written as

$$J_1 = \min\left\{\sum_{t=0}^{T} M_t^0, \sum_{t=0}^{T} M_t^1, \dots, \sum_{t=0}^{T} M_t^\delta\right\}. \tag{27}$$

Also, the objective in Problem 2A can be written as

$$J_2 = \sum_{t=0}^{T} \min\left\{M_t^0, M_t^1, \dots, M_t^\delta\right\}. \tag{28}$$

For each t, we set $\min\{M_t^0,\dots,M_t^\delta\} = \underline{M}_t$, and we have $M_t^d \ge \underline{M}_t, \forall d \in \{0,\dots,\delta\}$. Plugging the inequalities into Equation (27), we obtain $J_1 \ge \min\{\sum_{t=0}^{T}\underline{M}_t,\dots,\sum_{t=0}^{T}\underline{M}_t\} = \sum_{t=0}^{T}\underline{M}_t = J_2$. The equality only holds when $M_t^0 = \cdots = M_t^\delta$. Furthermore, Equation (19) establishes the relation between J_0 and J_1, since J_1 is a direct LSE decomposition of J_2. Namely, we have $\frac{1}{\beta}\log J_1 \le J_0 + \frac{1}{\beta}\log(T+1)$. Thus $\frac{1}{\beta}\log J_2 \le J_0 + \frac{1}{\beta}\log(T+1)$ also holds as $J_2 \le J_1$, which means that $\frac{1}{\beta}\log J_2 - \frac{1}{\beta}\log(T+1)$ is a lower bound for the objective function value of the original Problem 2. Thus maximizing J_2 maximizes the lower bound of J_0. The proof for $\Phi = \mathbf{G}_{[0,H)}\phi$ is analogous. ■

V. Case Studies

In this section, we illustrate the effectiveness of our algorithm by conducting a set of four simulation experiments[1]. All algorithms were implemented using `MATLAB` and simulated in `Coppeliasim` on a Windows 10 computer with an Intel Core i7-8550U 1.80GHz processor.

System Descriptions: We consider a scenario, where a warehouse robot navigates in a manufacturing factory floor as depicted in Figure 2. There are three functional areas in the workspace: a storage area marked by the parcel icon and two loading stations marked by A and B. These areas will be used later for specifying tasks. Depending on the specific task, the workspace is abstracted as $n\times n$ grid worlds shown in Figure 1, where star marks the initial state and yellow regions mark the goal states. We consider the high-level decision-making problem on the abstracted grid world, where agent can choose to move to its adjacent grids or to stay for each decision time step. If the agent chooses an action leading to collisions with the boundaries, then it will be forced to stay at the same grid. The high-level decisions from grid to grid are then executed by a low-level hybrid controller that ensures collision avoidance along the way. The objective is to synthesize a control policy that generates trajectories over grids satisfying a given STL task.

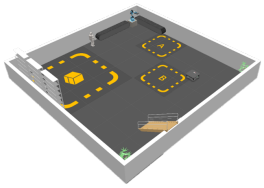

Fig. 2: Aerial view of the factory floor

Policy Learning Setups: We modify the standard tabular Q-learning algorithm to synthesize policies. To approximate reward functions, we choose $\beta = 50$. The learning rate for the update is chosen as $\alpha = \max\{0.95\times 0.999^i, 0.0001\}$ at the ith episode and the discount factor is chosen as $\gamma = 0.9999$. We settle for the policy after 10^4 episodes of training.

Result Evaluations: Once a control policy is obtained, we evaluate it by generating 1000 simulation trajectories. Note that, we do not explicitly introduce time delays in the simulation, but evaluate each trajectory by its robustness to potential delays. For each trajectory, we compute the following four metrics by existing computation methods for STL: (i) Boolean satisfaction, (ii) satisfaction with temporal robustness guarantees, (iii) spatial robustness value and (iv) temporal robustness value. Then we compute the statistic values, including the average satisfaction rate $\Pr(\mathbf{s}_{0:T} \models \Phi)$, the average time-robust satisfaction rate $\Pr[\theta(\Phi,\mathbf{s}_{0:T}) \le \delta]$, the average spatial robustness $\bar{\rho}(\Phi,\mathbf{s}_{0:T})$ and the average temporal robustness $\bar{\theta}(\Phi,\mathbf{s}_{0:T})$ and summarize it in Table I. For each scenario, we pick one of the sample trajectories with the highest episodic reward for the purpose of demonstration as shown in Figure 1.

[1]Sample videos and codes are available at `https://github.com/WSQsGithub/TimeRobustLearning`.

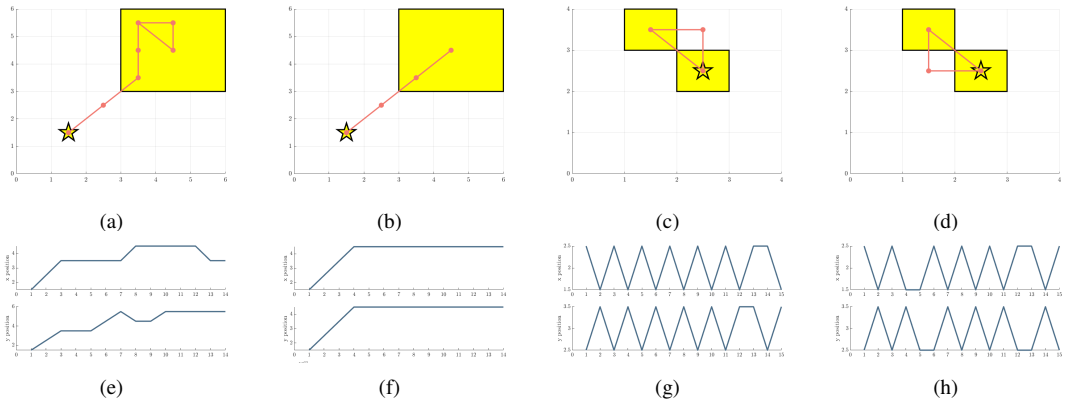

Fig. 1: (a)-(d) From left to right: Sample Trajectories generated by π^*_{1A} for reachability task, π^*_{2A} for reachability task, π^*_{1A} for patrolling task and π^*_{2A} for patrolling task

TABLE I: Experiment parameters and results

Prob.	Task	Grid size	#Q-entry	δ	$\Pr(\mathbf{s}_{0:T} \models \Phi)$	$\Pr[\theta(\Phi, \mathbf{s}_{0:T}) \geq \delta]$	$\bar{\rho}(\Phi, \mathbf{s}_{0:T})$	$\tilde{\theta}(\Phi, \mathbf{s}_{0:T})$
1A	Φ_1	6	12358	2	0.967	0.963	0.994	8.782
2A	Φ_1	6	13135	2	0.957	0.95	1.274	8.154
1A	Φ_2	4	4574	1	0.648	0.55	0.043	-3.76
2A	Φ_2	4	4534	1	0.657	0.568	0.079	-3.626

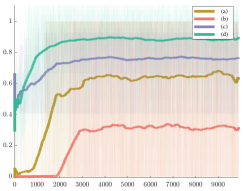

Fig. 3: Learning curves of training the policies in Figure 1(a-d), respectively

Reachability Task: In the first case study, we consider a 6×6 world and a reachability task described by

$$\Phi_1 = \mathbf{F}_{[0,12)}\mathbf{G}_{[0,2)}(s \in Goal).$$

Specifically, within 14 steps, the agent needs to move to the storage area and stay for at least 2 steps. We synthesize policies for both Problems 1A and 2A and show the sample trajectories in abstract space in Figures 1(a) and (e) for Problem 1A and Figures 1(b) and (f) for Problem 2A, respectively. The robot manages to find the shortest path to the goal region and stays there until the task ends even though the STL specification only requires it to stay there for 2 steps. Figure 1(b) further shows that the robot is driven to the center of the goal region as a result of considering spatial robustness. In this case, the policy of Problem 2A generates more spatially robust trajectories.

Patrolling Task: In the second case study, we consider a 4×4 world and a patrolling task described by

$$\Phi_2 = \mathbf{G}_{[0,12)}\left[\mathbf{F}_{[0,3)}(s \in A) \wedge \mathbf{F}_{[0,3)}(s \in B)\right].$$

Specifically, within 15 time steps, the robot needs to visit regions A and B (two yellow grids) every 3 time steps to load and unload workpieces. We still synthesize policies for both Problems 1A and 2A. Sample trajectories are provided in Figures 1(c) and (g) for Problem 1A, and Figures 1(d) and(h) for Problem 1B, respectively. For this task, since we take temporal robustness into account, the robot leaves the goal region immediately once the work piece is (un)loaded. Compared with the reachability tasks, this patrolling task is more difficult to achieve since we need to satisfy the sub-task for the entire horizon, which explains the relatively low average temporal robustness compared to the reachability task, as indicated in Table I.

Discussions: Table I shows that the performance distinction between policies from the two problem formulations is statistically insignificant. This is largely because that even though two problems are theoretically formulated differently, the shaped reward is numerically similar. Also, because β has to be sufficiently large to approximate $\min$ or $\max$ operators, $e^{\beta x}$ becomes very large when $x > 0$ and close to 0 otherwise. This scaling further narrows the gap between the two problems. Nevertheless, Figure 1 shows that the spatial robustness maximization drives the agent towards the goal region's center.

Figure 3 reveals intriguing insights from the learning curves. For the patrolling task, the potential of expediting learning through spatial robustness consideration is evident. While for the reachability task, from Figure 3, it seems that the optimum is later reached. However, this is because that the agent gets significantly higher reward at the central grid of the goal region than the others. The agent trained on Problem 2A can already generate as good trajectories as the agent trained on Problem 1A before the latter policy reaches convergence. The fact that the central grid which generates the best reward when visited, is distant from the initial state, contributes to the late convergence as it takes longer to find the central grid.

VI. Conclusion and Future Work

In this work, we propose a novel reinforcement learning approach to enhance the temporal robustness of signal temporal logic tasks for unknown stochastic systems. We present two optimization problems to maximize temporal robustness probability and expected spatial-temporal robustness. Additionally, we provided approximation techniques that enable the application of standard Q-learning techniques. Experimental results demonstrate the effectiveness of our proposed approach. In the future, we plan to extend our results to more general fragments of STL tasks. We also aim to investigate how to incorporate the concept of asynchronous temporal robustness in reinforcement learning and to consider the case of continuous state and action space.

REFERENCES

[1] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *FORMATS*, pp. 152–166, Springer, 2004.

[2] A. Donzé, T. Ferrere, and O. Maler, "Efficient robust monitoring for STL," in *International Conference on Computer Aided Verification*, pp. 264–279, Springer, 2013.

[3] X. Yu, W. Dong, S. Li, and X. Yin, "Model predictive monitoring of dynamical systems for signal temporal logic specifications," *Automatica*, vol. 160, p. 111445, 2024.

[4] G. Silano, T. Baca, R. Penicka, D. Liuzza, and M. Saska, "Power line inspection tasks with multi-aerial robot systems via signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4169–4176, 2021.

[5] S. S. Farahani, S. Soudjani, R. Majumdar, and C. Ocampo-Martinez, "Formal controller synthesis for wastewater systems with signal temporal logic constraints: The Barcelona case study," *Journal of Process Control*, vol. 69, pp. 179–191, 2018.

[6] M. Ma, E. Bartocci, E. Lifland, J. A. Stankovic, and L. Feng, "A novel spatial–temporal specification-based monitoring system for smart cities," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11793–11806, 2021.

[7] M. Hekmatnejad, S. Yaghoubi, A. Dokhanchi, H. B. Amor, A. Shrivastava, L. Karam, and G. Fainekos, "Encoding and monitoring responsibility sensitive safety rules for automated vehicles in signal temporal logic," in *ACM-IEEE International Conference on Formal Methods and Models for System Design*, pp. 1–11, 2019.

[8] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *FORMATS*, pp. 92–106, Springer, 2010.

[9] A. Rodionova, L. Lindemann, M. Morari, and G. J. Pappas, "Combined left and right temporal robustness for control under STL specifications," *IEEE Control Systems Letters*, vol. 7, pp. 619–624, 2022.

[10] A. Rodionova, L. Lindemann, M. Morari, and G. Pappas, "Temporal robustness of temporal logic specifications: Analysis and control design," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 1, pp. 1–44, 2022.

[11] L. Lindemann, A. Rodionova, and G. Pappas, "Temporal robustness of stochastic signals," in *ACM International Conference on Hybrid Systems: Computation and Control*, pp. 1–11, 2022.

[12] X. Yu, X. Yin, and L. Lindemann, "Efficient stl control synthesis under asynchronous temporal robustness constraints," in *IEEE Conference on Decision and Control*, pp. 6847–6854, IEEE, 2023.

[13] J. V. Deshmukh, R. Majumdar, and V. S. Prabhu, "Quantifying conformance using the Skorokhod metric," *Formal Methods in System Design*, vol. 50, pp. 168–206, 2017.

[14] T. Akazaki and I. Hasuo, "Time robustness in mtl and expressivity in hybrid system falsification," in *International Conference on Computer Aided Verification*, pp. 356–374, Springer, 2015.

[15] Z. Lin and J. S. Baras, "Optimization-based motion planning and runtime monitoring for robotic agent with space and time tolerances," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1874–1879, 2020.

[16] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *IEEE Conference on Decision and Control*, pp. 81–87, 2014.

[17] V. Kurtz and H. Lin, "Mixed-integer programming for signal temporal logic with fewer binary variables," *IEEE Control Systems Letters*, vol. 6, pp. 2635–2640, 2022.

[18] D. Sun, J. Chen, S. Mitra, and C. Fan, "Multi-agent motion planning from signal temporal logic specifications," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3451–3458, 2022.

[19] X. Yu, C. Wang, D. Yuan, S. Li, and X. Yin, "Model predictive control for signal temporal logic specifications with time interval decomposition," in *IEEE Conference on Decision and Control*, pp. 7849–7855, IEEE, 2023.

[20] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2018.

[21] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[22] W. Xiao, C. A. Belta, and C. G. Cassandras, "High order control lyapunov-barrier functions for temporal logic specifications," in *American Control Conference*, pp. 4886–4891, 2021.

[23] D. Aksaray, A. Jones, Z. Kong, M. Schwager, and C. Belta, "Q-learning for robust satisfaction of signal temporal logic specifications," in *IEEE Conference on Decision and Control*, pp. 6565–6570, 2016.

[24] A. Balakrishnan and J. V. Deshmukh, "Structured reward functions using STL," in *ACM International Conference on Hybrid Systems: Computation and Control*, pp. 270–271, 2019.

[25] K. C. Kalagarla, R. Jain, and P. Nuzzo, "Model-free reinforcement learning for optimal control of markov decision processes under signal temporal logic specifications," in *IEEE Conference on Decision and Control*, pp. 2252–2257, 2021.

[26] H. Venkataraman, D. Aksaray, and P. Seiler, "Tractable reinforcement learning of signal temporal logic objectives," in *Learning for Dynamics and Control*, pp. 308–317, 2020.

[27] J. Ikemoto and T. Ushio, "Deep reinforcement learning under signal temporal logic constraints using Lagrangian relaxation," *IEEE Access*, vol. 10, pp. 114814–114828, 2022.

[28] A. Rodionova, L. Lindemann, M. Morari, and G. J. Pappas, "Time-robust control for STL specifications," in *IEEE Conference on Decision and Control*, pp. 572–579, 2021.

[29] A. T. Buyukkocak and D. Aksaray, "Temporal relaxation of signal temporal logic specifications for resilient control synthesis," in *IEEE Conference on Decision and Control*, pp. 2890–2896, 2022.

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