Published in final edited form as:
IJCAI (U S). 2019 ; 28: 4010–4018. doi:10.24963/ijcai.2019/557.

Transfer of Temporal Logic Formulas in Reinforcement Learning

Zhe Xu, **Ufuk Topcu**
University of Texas at Austin

Abstract

Transferring high-level knowledge from a *source task* to a *target task* is an effective way to expedite reinforcement learning (RL). For example, propositional logic and first-order logic have been used as representations of such knowledge. We study the transfer of knowledge between tasks in which the timing of the events matters. We call such tasks *temporal tasks*. We concretize similarity between temporal tasks through a notion of *logical transferability*, and develop a transfer learning approach between different yet *similar* temporal tasks. We first propose an inference technique to extract *metric interval temporal logic* (MITL) formulas in *sequential disjunctive normal form* from labeled trajectories collected in RL of the two tasks. If logical transferability is identified through this inference, we construct a timed automaton for each *sequential conjunctive subformula* of the inferred MITL formulas from both tasks. We perform RL on the *extended state* which includes the locations and clock valuations of the timed automata for the source task. We then establish mappings between the corresponding components (clocks, locations, etc.) of the timed automata from the two tasks, and transfer the *extended Q-functions* based on the established mappings. Finally, we perform RL on the *extended state* for the target task, starting with the transferred extended Q-functions. Our implementation results show, depending on how similar the source task and the target task are, that the sampling efficiency for the target task can be improved by up to one order of magnitude by performing RL in the extended state space, and further improved by up to another order of magnitude using the transferred extended Q-functions.

1 Introduction

Reinforcement learning (RL) has been successful in numerous applications. In practice though, it often requires extensive exploration of the environment to achieve satisfactory performance, especially for complex tasks with sparse rewards [Wang and Taylor, 2017].

The sampling efficiency and performance of RL can be improved if some high-level knowledge can be incorporated in the learning process [Toro Icarte *et al.*, 2018a]. Such knowledge can be also transferred from a *source task* to a *target task* if these tasks are *logically similar* [Taylor and Stone, 2007]. For example, propositional logic and first-order logic have been used as representations of knowledge in the form of *logical structures* for transfer learning [Mihalkova *et al.*, 2007]. They showed that incorporating such logical similarities can expedite RL for the target task [Torrey *et al.*, 2008].

zhexu@utexas.edu.

The transfer of high-level knowledge can be also applied to tasks where the timing of the events matters. We call such tasks as *temporal tasks*. Consider the gridworld example in Figure 1. In the source task, the robot should first reach a green region G^{S} and stay there for at least 4 time units, then reach another yellow region Y^{S} within 40 time units. In the target task, the robot should first reach a green region G^{T} and stay there for at least 5 time units, then reach another yellow region Y^{T} within 40 time units. In both tasks, the green and yellow regions are a priori unknown to the robot. After 40 time units, the robot obtains a reward of 100 if it has completed the task and obtains a reward of −10 otherwise. It is intuitive that the two tasks are similar at a high level despite the differences in the specific regions in the workspace and timing requirements.

Transfer learning between temporal tasks is complicated due to the following factors: (a) No formally defined criterion exists for logical similarities between temporal tasks. (b) Logical similarities are often implicit and need to be identified from data. (c) There is no known automated mechanism to transfer the knowledge based on logical similarities.

In this paper, we propose a transfer learning approach for temporal tasks in two levels: transfer of logical structures and transfer of low-level implementations. For ease of presentation, we focus on Q-learning [Watkins and Dayan, 1992], while the general methodology applies readily to other forms of RL.

In the first level, we represent the high-level knowledge in temporal logic [Pnueli, 1977], which has been used in many applications in robotics and artificial intelligence [Kress-Gazit *et al.*, 2011; To *et al.*, 2016]. Specifically, we use a fragment of metric interval temporal logic (MITL) with bounded time intervals. We transfer such knowledge from a source task to a target task based on the hypothesis of *logical transferability* (this notion will be formalized in Section 4.1) between the two tasks.

To identify logical transferability, we develop an inference technique that extracts *informative* MITL formulas (formalized in Section 3) in *sequential disjunctive normal form*. If the inference process indeed identifies logical transferability, we construct a timed automaton for each *sequential conjunctive subformula* of the inferred MITL formulas. We combine the locations and clock valuations of the timed automaton with the state of the robot to form an *extended state*, and perform RL in the extended state space for the target task.

In the second level, we transfer the *extended Q-functions* (i.e., Q-function on the extended states) from the source task to the target task if logical transferability is identified in the first level. We first perform RL in the extended state space for the source task. Next, we establish mappings between the corresponding components (clocks, locations, etc.) of the timed automata from the two tasks based on the identified logical transferability. Finally, we transfer the obtained optimal extended Q-functions from the source task to the target task based on these mappings, and perform RL for the target task starting with the transferred extended Q-functions.

The implementation of the proposed approach shows, in both levels, that the sampling efficiency is significantly improved for RL of the target task.

1.1 Related Work

Our work is closely related to the work on RL with temporal logic specifications [Aksaray *et al.*, 2016; Li *et al.*, 2017; Toro Icarte *et al.*, 2018b; Fu and Topcu, 2014; Wen *et al.*, 2017; Alshiekh *et al.*, 2018]. The current results mostly rely on the assumption that the high-level knowledge (i.e., temporal logic specifications) are given, while in reality they are often implicit and need to be inferred from data.

The methods for inferring temporal logic formulas from data can be found in [Hoxha *et al.*, 2017; Kong *et al.*, 2017; Bombara *et al.*, 2016; Neider and Gavran, 2018; Xu *et al.*, 2018; Xu and Julius, 2018; Vazquez-Chanlatte *et al.*, 2018; Xu *et al.*, 2019; Shah *et al.*, 2018]. The inference method used in this paper is inspired from [Bombara *et al.*, 2016] and [Xu *et al.*, 2019].

While there has been no existsing work on RL-based transfer learning utilizing similarity between temporal logic formulas, the related work on transferring first-order logical structures or rules for expediting RL can be found in [Taylor and Stone, 2007; Torrey and Shavlik, 2010; Torrey *et al.*, 2008], and the related work on transferring logical relations for action-model acquisition can be found in [Zhuo and Yang, 2014].

2 Preliminaries

2.1 Metric Interval Temporal Logic

Let $\mathbb{B} = \{ \top , \bot \}$ (tautology and contradiction, respectively) be the Boolean domain and $\mathbb{T} = \{0, 1, 2, \dots\}$ be a discrete set of time indices. The underlying system is modeled by a *Markov decision process* (MDP) $\mathscr{M} = (S, A, P)$, where the state space S and action set A are finite, $P: S \times A \times S \rightarrow$ [0,1] is a transition probability distribution. A trajectory $s_{0:L} = s_0 s_1 \cdots s_L$ describing an evolution of the MDP $\mathscr{M}$ is a function from $\mathbb{T}$ to S. Let AP be a set of *atomic predicates*.

The syntax of the MITL$_f$ fragment of time-bounded MITL formulas is defined recursively as follows[1]:

$$\phi := \top \mid \rho \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \Diamond_I \phi \mid \Box_I \phi,$$

where $\rho \in AP$ is an atomic predicate; $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction) are Boolean connectives; $\Diamond$ (eventually) and $\Box$ (always) are temporal operators; and I is a bounded interval of the form $I = [i_1, i_2] (i_1 < i_2, i_1, i_2 \in \mathbb{T})$. For example, the MITL$_f$ formula $\Box_{[2,5]} (x > 3)$ reads as "x is always greater than 3 during the time interval [2, 5]".

A *timed word* generated by a trajectory $s_{0:L}$ is defined as a sequence $\left(\mathscr{L}\left(s_{t_1}\right), t_1\right), \dots, \left(\mathscr{L}\left(s_{t_m}\right), t_m\right)$, where $\mathscr{L}: S \rightarrow 2^{\mathscr{AP}}$ is a labeling function assigning to each state s

[1]Although other temporal operators such as "Until "($\mathscr{U}$) may also appear in the full syntax of MITL, they are omitted from the syntax here as they can be hard to interpret and are not often used for the inference of temporal logic formulas [Kong *et al.*, 2017].

$\in S$ a subset of atomic predicates in $\mathcal{AP}$ that hold true at state s, $t_1 = 0$, $t_m = L$, $t_{k-1} < t_k$ ($k \in [2, m]$) and for $k \in [1, m-1]$, t_{k+1} is the largest time index such that $\mathcal{L}(s_t) = \mathcal{L}\left(s_{t_k}\right)$ for all $t \in [t_k, t_{k+1})$. The satisfaction of an MITL$_f$ formula by timed words as Boolean semantics can be found in [Alur *et al.*, 1996]. We say that a trajectory $s_{0:L}$ satisfies an MITL$_f$ formula ϕ, denoted as $s_{0:L} = \phi$, if and only if the timed word generated by $s_{0:L}$ satisfies ϕ. As the time intervals I in MITL$_f$ formulas are bounded intervals, MITL$_f$ formulas can be satisfied and violated by trajectories of finite lengths.

2.2 Timed Automaton

Let C be a finite set of clock variables. The set $\mathcal{C}_C$ of clock constraints is defined by [Ouaknine and Worrell, 2005]

$$\varphi_C := \top \mid c \bowtie k \mid \varphi_1 \wedge \varphi_2,$$

where $k \in \mathbb{N}$, $c \in C$ and $\bowtie \in \{<, \leq, >, \geq\}$.

Definition 1. [Alur and Dill, 1994] A timed automaton is a tuple $\mathcal{A} = \left(\Sigma, \mathcal{Q}, q^0, C, \mathcal{F}, \Delta\right)$, where Σ is a finite alphabet of input symbols, $\mathcal{Q}$ is a set of locations, $q^0 \in \mathcal{Q}$ is the initial location, C is a finite set of clocks, $\mathcal{F} \subset \mathcal{Q}$ is a set of accepting locations, $\Delta \subset \mathcal{Q} \times \Sigma \times \mathcal{Q} \times \mathcal{C}_C \times 2^C$ is the transition function, $e = (q, \sigma, q', \varphi_C, r_C) \in \Delta$ represents a transition from q to q' labeled by σ, provided the precondition φ_C on the clocks is met, r_C is the set of clocks that are reset to zero..

Remark 1. We focus on timed automata with discrete time, which are also called tick automata in [Gruber *et al.*, 2005].

A timed automaton $\mathcal{A}$ is *deterministic* if and only if for each location and input symbol there is at most one transition to the next location. We denote by $v = \left(v_1, \ldots, v_{|C|}\right) \in V \subset \mathbb{T}^{|C|}$ the *clock valuation* of $\mathcal{A}$ (we denote by $|C|$ the cardinality of C), where $v_k \in V_k \subset \mathbb{T}$ is the value of clock $c_k \in C$. For a timed word $v = (\sigma_0, t_0), (\sigma_1, t_1), \ldots, (\sigma_m, t_m)$ (where $t_0 < t_1 < \cdots < t_m$, $\sigma_k \in \Sigma$ for $k \in [0, m]$) and writing $d_k := t_{k+1} - t_k$, a *run* of $\mathcal{A}$ on v is defined as

$$(q^0, v^0) \xrightarrow{\sigma_0} (q^1, v^1) \overset{d_0}{\rightsquigarrow} (q^2, v^2) \xrightarrow{\sigma_1} (q^3, v^3) \overset{d_1}{\rightsquigarrow} \ldots \overset{d_{m-1}}{\rightsquigarrow} (q^{2m}, v^{2m}) \xrightarrow{\sigma_m} (q^{2m+1}, v^{2m+1}),$$

where the *flow-step* relation is defined by $(q, v) \xrightarrow{d} (q, v + d)$ where $d \in \mathbb{R}_{>0}$; the *edge-step* relation is defined by $(q, v) \overset{\sigma}{\rightsquigarrow} (q', v')$ if and only if there is an edge $(q, \sigma, q', \varphi_C, r_C) \in \Delta$ such that $\sigma \in \Sigma$, v satisfies φ_C, $v'_k = 0$ for all $c_k \in r_C$ and $v'_k = v_k$ for all $c_k \notin r_C$. A finite run is *accepting* if the last location in the run belongs to $\mathcal{F}$. A timed word v is accepted by $\mathcal{A}$ if there is some accepting run of $\mathcal{A}$ on v.

3 Information-Guided Inference of Temporal Logic Formulas

We now introduce the *information gain* provided by an MITL$_f$ formula, the problem formulation and the algorithm to extract MITL$_f$ formulas from labeled trajectories.

3.1 Information Gain of MITL$_f$ Formulas

We denote by $\mathscr{B}_L$ the set of all possible trajectories with length L generated by the MDP $\mathscr{M}$, and use $\mathscr{G}_L : \mathscr{B}_L \to [0, 1]$ to denote a prior probability distribution (e.g., uniform distribution) over $\mathscr{B}_L$. We use $\mathbb{P}_{\mathscr{B}_L, \phi}$ to denote the probability of a trajectory $s_{0:L}$ satisfying ϕ in $\mathscr{B}_L$ based on $\mathscr{G}_L$.

Definition 2. Given a prior probability distribution $\mathscr{G}_L$ and an *MITL$_f$* formula ϕ such that $\mathbb{P}_{\mathscr{B}_L, \phi} > 0$, we define $\overline{\mathscr{G}}_L^{\phi} : \mathscr{B}_L \to [0, 1]$ as the posterior probability distribution, given that ϕ evaluates to true, which is expressed as

$$\overline{\mathscr{G}}_L^{\phi}(s_{0:L}) := \begin{cases} \dfrac{\mathscr{G}_L(s_0 - L)}{\mathbb{P}_{\mathscr{B}_L, \phi}}, & \text{if } s_{0:L} \models \phi, \\ 0, & \text{otherwise}. \end{cases}$$

The expression of $\overline{\mathscr{G}}_L^{\phi}$ can be derived using Bayes' theorem. We use the fact that the probability of ϕ evaluating to true given $s_{0:L}$ is 1, if $s_{0:L}$ satisfies ϕ; and it is 0 otherwise.

Definition 3. When the prior probability distribution $\mathscr{G}_L$ is updated to the posterior probability distribution $\overline{\mathscr{G}}_L^{\phi}$, we define the information gain as

$$\mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi}\right) := D_{\mathrm{KL}}\left(\overline{\mathscr{G}}_L^{\phi} \middle\| \mathscr{G}_L\right) / L,$$

where $D_{\mathrm{KL}}\left(\overline{\mathscr{G}}_L^{\phi} \middle\| \mathscr{G}_L\right)$ is the Kullback-Leibler divergence from $\mathscr{G}_L$ to $\overline{\mathscr{G}}_L^{\phi}$.

Proposition 1. For an *MITL$_f$* formula ϕ, if $\mathbb{P}_{\mathscr{B}_L, \phi} > 0$, then

$$\mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi}\right) = -\log \mathbb{P}_{\mathscr{B}_L, \phi} / L.$$

Proof. Straightforward from Definitions 2 and 3. □

If $\phi = \top$, then $\mathbb{P}_{\mathscr{B}_L,\phi} = 1$ and $\mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi}\right) = 0$, i.e., tautologies provide no information gain.

For completeness, we also define that the information gain $\mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi}\right) = 0$ if $\mathbb{P}_{\mathscr{B}_L,\phi} > 0$. So if $\phi = \bot$, then $\mathbb{P}_{\mathscr{B}_L,\phi} > 0$ and $\mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi}\right) = 0$, i.e., contradictions provide no information gain.

For two MITL$_f$ formulas ϕ_1 and ϕ_2, we say ϕ_1 is more *informative* than ϕ_2 with respect to the prior probability distribution $\mathscr{G}_L$ if $\mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi_1}\right) > \mathscr{I}\left(\mathscr{G}_L, \overline{\mathscr{G}}_L^{\phi_2}\right)$.

Based on Proposition 1, the computation of the information gain requires the computation of $\mathbb{P}_{\mathscr{B}_L,\phi}$. We point the reader to [Xu *et al.*, 2019] for a recursive method to compute $\mathbb{P}_{\mathscr{B}_L,\phi}$.

3.2 Problem Formulation

We now provide some related definitions for formulating the inference problem. Let a set $\mathscr{P}$ of *primitive structures* [Bombara *et al.*, 2016] used in the rest of the paper be

$$\mathscr{P} := \left\{ \Diamond_I \rho, \Box_I \rho, \Diamond_I \Box_{I'} \rho, \Box_I \Diamond_{I'} \rho \right\}, \tag{1}$$

where $I = [i_1, i_2] (i_1 < i_2, i_1, i_2 \in \mathbb{T})$, $I' = [0, i_2] (i_2 > 0, i_2 \in \mathbb{T})$, and ρ is an atomic predicate. We call an MITL$_f$ formula ϕ a *primitive MITL$_f$ formula* if ϕ follows one of the primitive structures in $\mathscr{P}$ or the negation of such a structure.

Definition 4. For an *MITL$_f$* formula ϕ, we define the start-effect time $t_s(\phi)$ and end-effect time $t_e(\phi)$ recursively as

$$t_s(\rho) = t_e(\rho) = 0, t_s(\neg\phi) = t_s(\phi), t_e(\neg\phi) = t_e(\phi),$$

$$t_s(\phi_1 \wedge \phi_2) = \min\{t_s(\phi_1), t_s(\phi_2)\},$$

$$t_e(\phi_1 \wedge \phi_2) = \max\{t_e(\phi_1), t_e(\phi_2)\},$$

$$t_s\left(\Diamond_{[t_1,t_2]}\phi\right) = t_s(\phi) + t_1, \quad t_e\left(\Diamond_{[t_1,t_2]}\phi\right) = t_e(\phi) + t_2,$$

$$t_s\left(\Box_{[t_1,t_2]}\phi\right) = t_s(\phi) + t_1, \quad t_e\left(\Box_{[t_1,t_2]}\phi\right) = t_e(\phi) + t_2.$$

Definition 5. An *MITL$_f$* formula ϕ is in disjunctive normal form if ϕ is expressed in the form of $\left(\phi_1^1 \wedge \cdots \wedge \phi_1^{n_1}\right) \vee \cdots \vee \left(\phi_m^1 \wedge \cdots \wedge \phi_m^{n_m}\right)$, where each ϕ_i^j is a primitive *MITL$_f$* formula (also called primitive subformula of ϕ). If, for any $i \in [1, m]$ and for all $j, k \in [1, n_i]$ such that $j <$

k, it holds that $t_{\mathrm{e}}(\phi_i^j) < t_{\mathrm{s}}(\phi_i^k)$, then we say $\boldsymbol{\phi}$ is in sequential disjunctive normal form (SDNF) and we call each $\phi_i := \phi_i^1 \wedge \cdots \wedge \phi_i^{n_i}$ a sequential conjunctive subformula.

In the following, we consider MITL$_f$ formulas only in the SDNF for reasons that will become clear in Section 4. We define the *size* of an MITL$_f$ formula $\boldsymbol{\phi}$ in the SDNF, denoted as $\varrho(\phi)$, as the number of primitive MITL$_f$ formulas in $\boldsymbol{\phi}$.

Suppose that we are given a set $\mathcal{S}_L = \left\{\left(s_{0:L}^k, l_k\right)\right\}_{k=1}^{Ns_L}$ of labeled trajectories, where $l_k = 1$ and $l_k = -1$ represent desired and undesired behaviors, respectively. We define the *satisfaction signature* $g_{\boldsymbol{\phi}}(s_{0:L}^k)$ of a trajectory $s_{0:L}^k$ as follows: $g_{\boldsymbol{\phi}}(s_{0:L}^k) = 1$, if $s_{0:L}^k$ satisfies $\boldsymbol{\phi}$; and $g_{\boldsymbol{\phi}}(s_{0:L}^k) = -1$, if $s_{0:L}^k$ does not satisfy $\boldsymbol{\phi}$. Note that here we assume that L is sufficiently large, thus $s_{0:L}^k$ either satisfies or violates $\boldsymbol{\phi}$. A labeled trajectory $\left(s_{0:L}^k, l_k\right)$ is *misclassified* by $\boldsymbol{\phi}$ if $g_{\boldsymbol{\phi}}(s_{0:L}^k) \neq l_k$. We use $CR(S_L, \phi) = \left|\left\{\left(s_{0:L}^k, l_k\right) \in \mathcal{S}_L : g_{\boldsymbol{\phi}}(s_{0:L}^k) = l_k\right\}\right| / \left|\mathcal{S}_L\right|$ to denote the classification rate of $\boldsymbol{\phi}$ in $\mathcal{S}_L$.

Problem 1. Given a set $\mathcal{S}_L = \left\{\left(s_{0:L}^k, l_k\right)\right\}_{k=1}^{NS_L}$ of labeled trajectories, a prior probability distribution $\mathcal{G}_L$, real constant $\zeta \in (0, 1]$ and integer constant $\varrho_{\mathrm{th}} \in (0, \infty)$, construct an *MITL*$_f$ formula $\boldsymbol{\phi}$ in the SDNF that maximizes $\mathcal{I}\left(\mathcal{G}_L, \overline{\mathcal{G}}_L^{\phi}\right)$ while satisfying

- the classification constraint $CR\left(\mathcal{S}_L, \phi\right) \geq \zeta$ and
- the size constraint $\varrho(\phi) \leq \varrho_{\mathrm{th}}$.

Intuitively, as there could be many MITL$_f$ formulas that satisfy the classification constraint and the size constraint, we intend to obtain the most informative one to be utilized and transferred as features of desired behaviors.

We call an MITL$_f$ formula that satisfies both the two constraints of Problem 1 a *satisfying formula* for $\mathcal{S}_L$.

3.3 Solution Based on Decision Tree

We propose an inference technique, which is inspired by [Bombara *et al.*, 2016] and [Xu *et al.*, 2019], in order to solve Problem 1. The technique consists of two steps. In the first step, we construct a decision tree where each non-leaf node is associated with a primitive MITL$_f$ formula (see the formulas inside the circles in Figure 2). In the second step, we convert the constructed decision tree to an MITL$_f$ formula in the SDNF.

In Algorithm 1, we construct the decision tree by recursively calling the *MITLtree* procedure from the root node to each leaf node. There are three inputs to the *MITLtree* procedure: (1) a set $\mathcal{S}$ of labeled trajectories assigned to the current node; (2) a formula $\boldsymbol{\phi}^{\mathrm{path}}$ to reach the current node (also called the *path formula*, see the formulas inside the rectangles in Figure

2); and (3) the depth h of the current node. The set $\mathcal{S}$ assigned to the root node is initialized as $\mathcal{S}_L$ in Problem 1, ϕ^{path} and h are initialized as $\top$ and 0, respectively.

For each node, we set a criterion $stop(\phi^{\text{path}}, h, \mathcal{S})$ to determine whether it is a leaf node (Line 2). Each leaf node is associated with label 1 or −1, depending on whether more than 50% of the labeled trajectories assigned to that node are with label 1 or not.

At each non-leaf node z, we construct a primitive MITL$_f$ formula ϕ_θ parameterized by $\theta \in \Upsilon_z$, where

$$\Upsilon_z := \left\{\theta \middle| \left[t_s(\phi_\theta), t_e(\phi_\theta)\right] \cap \left[t_s(\phi'), t_e(\phi')\right] = \varnothing \text{ for each primitive subformula } \phi' \text{ of } \phi^{\text{path}}\right\}. \quad (2)$$

For example, for an MITL$_f$ formula $\phi_\theta = \Box_{[i_1, i_2]}(x > a)$, we have $\theta = [i_1, i_2, a]$. If $\phi^{\text{path}} = \Diamond_{[1,15]}\Box_{[0,4]}(x > 3)$, then Υ_z ensures that the start-effect time of ϕ_θ is later than the end-effect time of ϕ^{path} (which is 19). Essentially Υ_z guarantees that the primitive MITL$_f$ formula ϕ_θ and primitive subformulas of ϕ^{path} can be reordered to form a sequential conjunctive subformula (see Definition 5).

Algorithm 1 Information-Guided MITL$_f$ Inference.

1: **procedure** $MITLtree(\mathcal{S} = \{(s^k_{0:L}, l_k)\}_{k=1}^{N_\mathcal{S}}, \phi^{\text{path}}, h)$
2: **if** $stop(\phi^{\text{path}}, h, \mathcal{S})$ **then**
3: Create node z_{Leaf} as a leaf node
4: **if** node z_{Leaf} is associated with label 1
5: $z_{\text{Leaf}}.\phi^{\text{path}} \leftarrow \phi^{\text{path}}$
6: **end if**
7: return z
8: **end if**
9: Create node z as a non-leaf node
10: Obtain Υ_z from (2)
11: $z.\phi \leftarrow \arg\max_{\phi \in \mathcal{P}, \theta \in \Upsilon_z} CR(\mathcal{S}, \phi_\theta) + \lambda \mathcal{I}(\mathcal{G}_L, \bar{\mathcal{G}}_L^{\phi_\theta})$
12: $\{\mathcal{S}_\top, \mathcal{S}_\bot\} \leftarrow partition(\mathcal{S}, z.\phi)$
13: $z.left \leftarrow MITLtree(\mathcal{S}_\top, \phi^{\text{path}} \wedge z.\phi, h+1)$
14: $z.right \leftarrow MITLtree(\mathcal{S}_\bot, \phi^{\text{path}} \wedge \neg z.\phi, h+1)$
15: return z
16: **end procedure**

We use particle swarm optimization (PSO) [Eberhart and Shi, 2001] to optimize θ for each primitive structure from $\mathcal{P}$ and compute a primitive MITL$_f$ formula $z.\phi = \phi^*_\theta$ which maximizes the objective function

$$J(\mathcal{S}, \phi_\theta) := CR(\mathcal{S}, \phi_\theta) + \lambda \mathcal{I}\left(\mathcal{G}_L, \overline{\mathcal{G}}_L^{\phi_\theta}\right) \tag{3}$$

in Line 11, where λ is a weighting factor.

With $z.\phi$, we partition the set $\mathcal{S}$ into $\mathcal{S}_\top$ and $\mathcal{S}_\perp$, where the trajectories in $\mathcal{S}_\top$ and $\mathcal{S}_\perp$ satisfy and violate $z.\phi$, respectively (Line 12). Then the procedure is called recursively to construct the left and right sub-trees for $\mathcal{S}_\top$ and $\mathcal{S}_\perp$, respectively (Lines 13, 14).

After the decision tree is constructed, for each leaf node associated with label 1, it is also associated with a path formula $z_{\text{Leaf}}.\phi^{\text{path}}$ (Line 4 to Line 6). The path formula $z_{\text{Leaf}}.\phi^{\text{path}}$ is constructed recursively from the associated primitive MITL$_f$ formulas along the path from the root node to the parent of the leaf node (see Figure 2). We rearrange the primitive subformulas of each $z_{\text{Leaf}}.\phi^{\text{path}}$ in the order of increasing start-effect time to obtain a sequential conjunctive subformula. We then connect all the obtained sequential conjunctive subformulas with disjunctions. In this way, the obtained decision tree can be converted to an MITL$_f$ formula in the SDNF. As in the example shown in Figure 2, if $t_s(\phi_1) < t_s(\phi_2)$ and $t_s(\neg\phi_1) < t_s(\phi_3)$, then the decision tree can be converted to $(\phi_1 \wedge \phi_2) \vee (\neg\phi_1 \wedge \phi_3)$ in the SDNF. If $t_s(\phi_2) < t_s(\phi_1)$ and $t_s(\phi_3) < t_s(\neg\phi_1)$, then the decision tree can be converted to $(\phi_2 \wedge \phi_1) \vee (\phi_3 \wedge \neg\phi_1)$ in the SDNF.

We set the criterion $stop(\phi^{\text{path}}, h, \mathcal{S})$ as follows. If at least ζ (e.g., 95%) of the labeled trajectories assigned to the node are with the same label (Condition I) or the depth h of the node reaches a set maximal depth $h_{\max}$ (Condition II) or Υ_z, as defined in (2), becomes the empty set (Condition III), then the node is a leaf node. If condition I holds for each leaf mode, then the obtained MITL$_f$ formula satisfies the classification constraint of Problem 1. If we set $h_{\max} 2^{h_{\max}-1} \leq \varrho_{\text{th}}$, then the size constraint is guaranteed to be satisfied.

The complexity of Algorithm 1 for the average case can be determined through the Akra-Bazzi method as follows [Bombara *et al.*, 2016]:

$$\Theta\left(N_{\mathcal{S}} \cdot \left(1 + \int_1^{N_{\mathcal{S}}} \frac{f(u)}{u^2} \mathrm{d}u\right)\right),$$

where $f(N_{\mathcal{S}})$ is the complexity of the local PSO algorithm for $N_{\mathcal{S}}$ labeled trajectories, and $\Theta(\cdot)$ denotes the two-sided asymptotic notation for complexity bound.

4 Transfer Learning of Temporal Tasks Based on Logical Transferability

In this section, we first introduce the notion of *logical transferability*. Then, we present the framework and algorithms for utilizing logical transferability for transfer learning.

4.1 Logical Transferability

To define logical transferability, we first define the *structural transferability* between two MITL$_f$ formulas.

For each primitive MITL$_f$ formula ϕ, we use $O_T(\phi)$ to denote the temporal operator in ϕ. For example, $O_T(\Diamond_{[5,8]}(x > 3)) = \Diamond$ (eventually) and $O_T(\Box_{[0,8]}\Diamond_{[0,4]}(x < 5)) = \Box\Diamond$ (always eventually).

Definition 6. Two *MITL$_f$* formulas (in the SDNF)

$$\phi = \left(\phi_1^1 \wedge \cdots \wedge \phi_1^{n_1}\right) \vee \cdots \vee \left(\phi_m^1 \wedge \cdots \wedge \phi_m^{n_m}\right)$$

and

$$\hat{\phi} = \left(\hat{\phi}_1^1 \wedge \cdots \wedge \hat{\phi}_1^{\hat{n}_1}\right) \vee \cdots \vee \left(\hat{\phi}_{\hat{m}}^1 \wedge \cdots \wedge \hat{\phi}_{\hat{m}}^{\hat{n}_{\hat{m}}}\right)$$

are structurally equivalent, if and only if the followings hold:

1. $m = \hat{m}$ and, for every $i \in [1, m]$, $n_i = \hat{n}_i$; and
2. For every $i \in [1, m]$ and every $j \in [1, n_i]$, $O_T\left(\phi_i^j\right) = O_T\left(\hat{\phi}_i^j\right)$.

Definition 7. For two *MITL$_f$* formulas ϕ_1 and ϕ_2 in the SDNF, ϕ_2 is structurally transferable from ϕ_1 if and only if either of the following conditions holds:

1. ϕ_1 and ϕ_2 are structurally equivalent;
2. ϕ_2 is in the form of $\phi_2^1 \vee \cdots \vee \phi_2^p (p > 1)$, where each $\phi_2^k (k = 1, \ldots, p)$ is structurally equivalent with ϕ_1.

Suppose that we are given a source task $\mathcal{T}^{\mathrm{S}}$ in the source environment $\mathcal{E}^{\mathrm{S}}$ and a target task $\mathcal{T}^{\mathrm{T}}$ in the target environment $\mathcal{E}^{\mathrm{T}}$, with two sets $\mathcal{S}_L^{\mathrm{S}}$ and $\mathcal{S}_L^{\mathrm{T}}$ of labeled trajectories collected during the initial episodes of RL (which we call the *data collection phase*) in $\mathcal{E}^{\mathrm{S}}$ and $\mathcal{E}^{\mathrm{T}}$ respectively. The trajectories are labeled based on a given task-related performance criterion. To ensure the quality of inference, the data collection phase is chosen such that both $\mathcal{S}_L^{\mathrm{S}}$ and $\mathcal{S}_L^{\mathrm{T}}$ contain sufficient labeled trajectories with both label 1 and label −l. We give the following definition for logical transferability.

Definition 8. $\mathcal{T}^T$ is logically transferable from $\mathcal{T}^S$ based on $\mathcal{S}_L^S$, $\mathcal{S}_L^T$, ζ and ϱ_{th} (as defined in Problem 1), if and only if there exist satisfying formulas $\boldsymbol{\phi}^S$ for $\mathcal{S}_L^S$ and $\boldsymbol{\phi}^T$ for $\mathcal{S}_L^T$ such that $\boldsymbol{\phi}^T$ is structurally transferable from $\boldsymbol{\phi}^S$.

In the following, we explain the proposed transfer learning approach based on logical transferability in two different levels. We provide a workflow diagram as an overview of the proposed transfer learning approach, as shown in Figure 3.

4.2 Transfer of Logical Structures Based on Hypothesis of Logical Transferability

We first introduce the transfer of logical structures between temporal tasks. To this end, we pose the hypothesis that the target task is logically transferable from the source task. If logical transferability can be indeed identified, we perform RL for the target task utilizing the transferred logical structure. Specifically, we take the following three steps:

Step 1: Extracting MITL$_f$ Formulas in the Source Task—From $\mathcal{S}_L^{\text{S}}$, we infer an MITL$_f$ formula $\boldsymbol{\phi}^{\text{S}}$ using Algorithm 1. If $\boldsymbol{\phi}^{\text{S}}$ is a satisfying formula for $\mathcal{S}_L^{\text{S}}$, we proceed to Step 2.

As in the introductory example, we obtain the satisfying formula $\phi^{\text{S}} = \Diamond_{[1,15]} \Box_{[0,4]} \bar{G}^{\text{S}} \wedge \Diamond_{[21,39]} \bar{Y}^{\text{S}}$ for $\mathcal{S}_L^{\text{S}}$ (see Section 5 for details).

Step 2: Extracting MITL$_f$ Formulas in the Target Task—From $\mathcal{S}_L^{\text{T}}$, we check if it is possible to infer a satisfying MITL$_f$ formula $\boldsymbol{\phi}^{\text{T}}$ for $\mathcal{S}_L^{\text{T}}$ such that $\boldsymbol{\phi}^{\text{T}}$ is structurally transferable from the inferred MITL$_f$ formula $\boldsymbol{\phi}^{\text{S}}$ from the source task. We start from inferring an MITL$_f$ formula that is structurally equivalent with $\boldsymbol{\phi}^{\text{S}}$. This can be done by fixing the temporal operators (the same with those of $\boldsymbol{\phi}^{\text{S}}$), then optimizing the parameters that appear in $\boldsymbol{\phi}^{\text{T}}$ (through PSO) for maximizing the objective function in (3). If a satisfying MITL$_f$ formula is not found, we infer a MITL$_f$ formula $\boldsymbol{\phi}^{\text{T}}$ in the form of $\phi^{\text{T}} = \phi_1^{\text{T}} \vee \phi_2^{\text{T}}$, where ϕ_1^{T} and ϕ_2^{T} are both structurally equivalent with $\boldsymbol{\phi}^{\text{S}}$. In this way, we keep increasing the number of structurally equivalent formulas connected with disjunctions until a satisfying MITL$_f$ formula is found, or the size constraint is violated (i.e., $\varrho\left(\phi^{\text{T}}\right) > \varrho_{\text{th}}$). If a satisfying MITL$_f$ formula is found, we proceed to Step 3; otherwise, logical transferability is not identified.

As in the introductory example, we obtain the satisfying formula $\phi^{\text{T}} = \Diamond_{[5,18]} \Box_{[0,5]} \bar{G}^{\text{T}} \wedge \Diamond_{[24,39]} \bar{Y}^{\text{T}}$ for $\mathcal{S}_L^{\text{T}}$ and $\boldsymbol{\phi}^{\text{T}}$ is structurally equivalent with $\boldsymbol{\phi}^{\text{S}}$, hence logical transferability is identified.

Step 3: Constructing Timed Automata and Performing RL in the Extended State Space for the Target Task—For the satisfying formula $\phi^{\text{T}} = \phi_1^{\text{T}} \vee \cdots \vee \phi_m^{\text{T}}$ in the SDNF, we can construct a deterministic timed automaton (DTA)

$\mathcal{A}^{\phi_i^{\mathrm{T}}} = \left\{2^{AP}, \mathcal{Q}^{\phi_i^{\mathrm{T}}}, q^{0\phi_i^{\mathrm{T}}}, C^{\phi_i^{\mathrm{T}}}, \mathcal{F}^{\phi_i^{\mathrm{T}}}, \Delta^{\phi_i^{\mathrm{T}}}\right\}$ [Alur *et al.*, 1996] that accepts precisely the timed words that satisfy each sequential conjunctive subformula ϕ_i^{T}.

We perform RL in the extended state space $X^{\mathrm{T}} = \cup_i X_i^{\mathrm{T}}$, where each $X_i^{\mathrm{T}} = S^{\mathrm{T}} \times \mathcal{Q}^{\phi_i^{\mathrm{T}}} \times V^{\phi_i^{\mathrm{T}}}$ (S^{T} is the state space for the target task, $V^{\phi_i^{\mathrm{T}}}$ is the set of clock valuations for the clocks in $C^{\phi_i^{\mathrm{T}}}$) is a finite set of extended states. For each episode, the index i is first selected based on some heuristic criterion. For example, if the atomic predicates correspond to the regions to be reached in the state space, we select i such that the centroid of the region corresponding to the atomic predicate in $\phi_i^{1,\mathrm{T}}$ (as in $\phi_i^{\mathrm{T}} = \phi_i^{1,\mathrm{T}} \wedge \cdots \wedge \phi_i^{n_i,\mathrm{T}}$) has the nearest (Euclidean) distance from the initial state s_0^{T}. Then we perform RL in X_i^{T}. For Q-learning, after taking action a^{T} at the current extended state $\chi_i^{\mathrm{T}} = \left(s^{\mathrm{T}}, q^{\mathrm{T}}, v^{\mathrm{T}}\right) \in X_i^{\mathrm{T}}$, a new extended state $\chi_i^{'\mathrm{T}} = \left(s'\mathrm{T}, q'\mathrm{T}, v^{'\mathrm{T}}\right) \in X_i^{\mathrm{T}}$ and a reward R^{T} are obtained. We have the following update rule for the *extended Q-function* values (denoted as $\bar{Q}$):

$$\bar{Q}\left(\chi_i^{\mathrm{T}}, a^{\mathrm{T}}\right) \leftarrow (1-\alpha)\bar{Q}\left(\chi_i^{\mathrm{T}}, a^{\mathrm{T}}\right) + \alpha\left(R^{\mathrm{T}} + \gamma \max_{a'^{\mathrm{T}}} \bar{Q}\left(\chi_i^{'\mathrm{T}}, a^{'\mathrm{T}}\right)\right),$$

where α and γ are the learning rate and discount factor, respectively.

As in the introductory example, we construct a DTA [see Figure 4 (b)] that accepts precisely the timed words that satisfy $\boldsymbol{\phi}^{\mathrm{T}}$ as there is only one sequential conjunctive subformula in $\boldsymbol{\phi}^{\mathrm{T}}$. We then perform RL in the extended state space $X^{\mathrm{T}} = S^{\mathrm{T}} \times \mathcal{Q}^{\phi^{\mathrm{T}}} \times V^{\phi^{\mathrm{T}}}$, where S^{T} is the state space in the 9×9 gridworld, $\mathcal{Q}^{\phi^{\mathrm{T}}} = \left\{q_0^{\mathrm{T}}, q_1^{\mathrm{T}}, q_2^{\mathrm{T}}, q_3^{\mathrm{T}}\right\}\left(q_0^{\mathrm{T}} = q^{0\phi^{\mathrm{T}}}\right)$ and $V^{\phi^{\mathrm{T}}} = \left\{0, 1, \ldots, 40\right\} \times \left\{0, 1, \ldots, 40\right\}$ (the set of clock valuations for the clocks c_1^{T} and c_2^{T}).

4.3 Transfer of Extended Q-functions Based on Identified Logical Transferability

Next, we introduce the transfer of extended Q-functions if logical transferability can be identified from Section 4.2.

We assume that the sets of actions in the source task and the target task are the same, denoted as A. For the satisfying formula $\phi^{\mathrm{S}} = \phi_1^{\mathrm{S}} \vee \cdots \vee \phi_m^{\mathrm{S}}$ in the SDNF, we construct a DTA corresponding to each ϕ_i^{S} and perform Q-learning in the extended state space for the source task. We denote the obtained optimal extended Q-functions as $\bar{Q}^{\mathrm{S}*}\left(s^{\mathrm{S}}, q^{\mathrm{S}}, v^{\mathrm{S}}, a\right)$. In the following, we explain the details for transferring $\bar{Q}^{\mathrm{S}*}\left(s^{\mathrm{S}}, q^{\mathrm{S}}, v^{\mathrm{S}}, a\right)$ to the target task based on the identified logical transferability.

From Definitions 6 and 7, if $\phi^{\mathrm{T}} = \phi_1^{\mathrm{T}} \vee \cdots \vee \phi_{m^{\mathrm{T}}}^{\mathrm{T}}$ is structurally transferable from $\phi^{\mathrm{S}} = \phi_1^{\mathrm{S}} \vee \cdots \vee \phi_{m^{\mathrm{S}}}^{\mathrm{S}}$, then for all index $i \in [1, m^{\mathrm{T}}]$, the sequential conjunctive subformulas ϕ_i^{T} and $\phi_{\hat{i}}^{\mathrm{S}}$ are structurally equivalent, where $\hat{i} = i \bmod m^{\mathrm{S}}$ (where mod denotes the *modulo* operation). For the DTA $\mathscr{A}^{\phi_i^{\mathrm{T}}}$ and $\mathscr{A}^{\phi_{\hat{i}}^{\mathrm{S}}}$ constructed from ϕ_i^{T} and $\phi_{\hat{i}}^{\mathrm{S}}$ respectively, it can be proven that we can establish bijective mappings: $\xi_{\Sigma}^{i,\hat{i}} : 2^{AP} \to 2^{AP}$, $\xi_{\mathbb{Q}}^{i,\hat{i}} : \mathbb{Q}^{\phi_i^{\mathrm{T}}} \to \mathbb{Q}^{\phi_{\hat{i}}^{\mathrm{S}}}$ and $\xi_{C}^{i,\hat{i}} : C^{\phi_i^{\mathrm{T}}} \to C^{\phi_{\hat{i}}^{\mathrm{S}}}$ such that the *structures* of $\mathscr{A}^{\phi_i^{\mathrm{T}}}$ and $\mathscr{A}^{\phi_{\hat{i}}^{\mathrm{S}}}$ are preserved under these bijective mappings [Glushkov, 1961]. Specifically, we have $\xi_{\mathbb{Q}}^{i,\hat{i}}\left(q^{0\phi_i^{\mathrm{T}}}\right) = q^{0\phi_{\hat{i}}^{\mathrm{S}}}$, $\xi_{\mathbb{Q}}^{i,\hat{i}}\left[\mathscr{F}^{\phi_i^{\mathrm{T}}}\right] = \mathscr{F}^{\phi_{\hat{i}}^{\mathrm{S}}}$ (where $\xi_{\mathbb{Q}}^{i,\hat{i}}\left[\mathscr{F}^{\phi_i^{\mathrm{T}}}\right]$ denotes the point-wise application of $\xi_{\mathbb{Q}}^{i,\hat{i}}$ to elements of $\mathscr{F}^{\phi_i^{\mathrm{T}}}$). Besides, for any $\rho \in 2^{AP}$ and any $q, q' \in \mathbb{Q}^{\phi_i^{\mathrm{T}}}$, we have that

$$e^{\phi_i^{\mathrm{T}}} = \left(q, \rho, q', \varphi_{C_1}^{\phi_i^{\mathrm{T}}}, r_{C_1}^{\phi_i^{\mathrm{T}}}\right) \in \Delta^{\phi_i^{\mathrm{T}}}$$

holds if and only if

$$e^{\phi_{\hat{i}}^{\mathrm{S}}} = \left(\xi_{\mathbb{Q}}^{i,\hat{i}}(q), \xi_{\Sigma}^{i,\hat{i}}(\rho), \xi_{\mathbb{Q}}^{i,\hat{i}}(q'), \varphi_{C_2}^{\phi_{\hat{i}}^{\mathrm{S}}}, r_{C_2}^{\phi_{\hat{i}}^{\mathrm{S}}}\right) \in \Delta^{\phi_{\hat{i}}^{\mathrm{S}}}$$

holds, where $C_1 = C^{\phi_i^{\mathrm{T}}}$ and $C_2 = \xi_C^{i,\hat{i}}\left[C^{\phi_i^{\mathrm{T}}}\right]$. See Figure 4 for an illustrative example.

Algorithm 2 is for the transfer of the extended Q-functions. For all indices i, we first identify a unique primitive MITL$_f$ formula $\phi_i^{j,\mathrm{T}}$ (as in $\phi_i^{\mathrm{T}} = \phi_i^{1,\mathrm{T}} \wedge \cdots \wedge \phi_i^{n_i,\mathrm{T}}$) that is to be satisfied at each extended state $\chi_i^{\mathrm{T}} = \left(s^{\mathrm{T}}, q^{\mathrm{T}}, v^{\mathrm{T}}\right)$ (Line 3). Specifically, according to q^{T} and v^{T}, we identify the index j such that $\phi_i^{1,\mathrm{T}}, \ldots, \phi_i^{(j-1),\mathrm{T}}$ are already satisfied while $\phi_i^{j,\mathrm{T}}$ is still not satisfied.

Next, we identify the state, location and clock valuation in the extended state $\chi_{\hat{i}}^{\mathrm{S}}$ that are the most *similar* to s^{T}, q^{T} and v^{T} respectively in the extended state χ_i^{T}.

Identification of State: We first identify the atomic predicate $\rho_i^{j,\mathrm{T}}$ in the primitive MITL$_f$ formula $\phi_i^{j,\mathrm{T}}$. Then we identify the atomic predicate ρ^{S} corresponding to $\rho_i^{j,\mathrm{T}}$ through the mapping $\xi_{\Sigma}^{i,\hat{i}}$ (Line 5). As in the introductory example, at the locations q_0^{S} and q_0^{T}, we first identify the atomic predicate $\bar{G}^{\mathrm{T}}$, then the mapping $\xi_{\Sigma}^{1,1}$ maps $\bar{G}^{\mathrm{T}}$ to its corresponding atomic predicate $\bar{G}^{\mathrm{S}}$. We use $Cr(\rho)$ to denote the centroid of the region corresponding to the atomic predicate ρ and $\|\cdot\|$ to denote the 2-norm. We identify the state s^{S} in S^{S} such that the relative position of s^{S} with respect to $Cr(\rho^{\mathrm{S}})$ is the most similar (measured in Euclidean distance) to the relative position of s^{T} with respect to $Cr\left(\rho_i^{j,\mathrm{T}}\right)$ (Line 6).

Algorithm 2 Transfer of Extended Q-functions Based on Logical Transferability.

1: **Inputs:** ϕ^{S}, ϕ^{T}, X^{S}, X^{T}, A
2: **for** all index i and all $\chi_i^{\mathrm{T}} = (s^{\mathrm{T}}, q^{\mathrm{T}}, v^{\mathrm{T}}) \in X_i^{\mathrm{T}}$ **do**
3: Identify the unique formula $\phi_i^{j,\mathrm{T}}$ and its atomic predicate $\rho_i^{j,\mathrm{T}}$
4: $\hat{i} \leftarrow \mod(i, m^{\mathrm{S}})$
5: $\rho^{\mathrm{S}} \leftarrow \xi_{\Sigma}^{i,\hat{i}}(\rho_i^{j,\mathrm{T}})$
6: $s^{\mathrm{S}} \leftarrow \arg\min_{s \in S^{\mathrm{S}}} \left\| \left(s - Cr(\rho^{\mathrm{S}})\right) - \left(s^{\mathrm{T}} - Cr(\rho_i^{j,\mathrm{T}})\right) \right\|$
7: $q^{\mathrm{S}} \leftarrow \xi_{\mathcal{Q}}^{i,\hat{i}}(q^{\mathrm{T}})$
8: **for** all $c_l^{\mathrm{T}} \in C^{\phi_i^{\mathrm{T}}}$ and all $v_l^{\mathrm{T}} \in V_l^{\phi_i^{\mathrm{T}}}$ **do**
9: $c_k^{\mathrm{S}} \leftarrow \xi_{C}^{i,\hat{i}}(c_l^{\mathrm{T}})$
10: $v_k^{\mathrm{S}} \leftarrow \arg\min_{v_k \in V_k^{\phi_{\hat{i}}^{\mathrm{S}}}} \left\| v_k - v_l^{\mathrm{T}} \right\|$
11: **end for**
12: $\bar{Q}^{\mathrm{T}}(s^{\mathrm{T}}, q^{\mathrm{T}}, v^{\mathrm{T}}, a) \leftarrow \bar{Q}^{\mathrm{S}*}(s^{\mathrm{S}}, q^{\mathrm{S}}, v^{\mathrm{S}}, a)$ **for** all $a \in A$
13: **end for**
14: Return $\bar{Q}^{\mathrm{T}}$

Identification of Location: We identify the location q^{S} corresponding to q^{T} through the mapping $\xi_{\mathcal{Q}}^{i,\hat{i}}$ (Line 7). As in the introductory example, the mapping $\xi_{\mathcal{Q}}^{1,1}$ maps the locations q_0^{T}, q_1^{T}, q_2^{T} and q_3^{T} to the locations q_0^{S}, q_1^{S}, q_2^{S} and q_3^{S}, respectively (see Figure 4).

Identification of Clock Valuation: For each clock $c_l^{\mathrm{T}} \in C^{\phi_i^{\mathrm{T}}}$, we identify the clock c_k^{S} corresponding to c_l^{T} through the mapping $\xi_{C}^{i,\hat{i}}$ (Line 9). As in the introductory example, the mapping $\xi_{C}^{1,1}$ maps the clocks c_1^{T} and c_2^{T} to the corresponding clocks c_1^{S} and c_2^{S}, respectively (see Figure 4). Then for each clock valuation $v_l^{\mathrm{T}} \in V_l^{\phi_i^{\mathrm{T}}}$, we identify the (scalar) clock valuation $v_k^{\mathrm{S}} \in V_k^{\phi_{\hat{i}}^{\mathrm{S}}}$ which is the most similar (in scalar value) to v_l^{T} (Line 10).

In this way, $\bar{Q}^{S*}\left(s^S, q^S, v^S, a\right)$ from the source task are transferred to $\bar{Q}^T\left(s^T, q^T, v^T, a\right)$ in the target task. Finally, we perform Q-learning of $\mathcal{T}^T$ in the extended state space, starting with the transferred extended Q-functions $\bar{Q}^T\left(s^T, q^T, v^T, a\right)$.

5 Implementation

In this section, we illustrate the proposed approach on a case study[2]. We consider the introductory example in the 9×9 gridworld as shown in Figure 1. The robot has three possible actions at each time step: go straight, turn left or turn right. After going straight, the robot may slip to adjacent cells with probability of 0.04. After turning left or turning right, the robot may stay in the original direction with probability of 0.03. We first perform Q-learning on the τ-states (i.e., the τ-horizon trajectory involving the current state and the most recent $\tau - 1$ past states, see [Aksaray *et al.*, 2016], we set τ=5) for the source task and the target task. We set $\alpha = 0.8$ and $\gamma = 0.99$. For each episode, the initial state is randomly selected.

We use the first 10000 episodes of Q-learning as the *data collection phase*. From the source task, all the 46 trajectories with cumulative rewards above 0 are labeled as 1, and 200 trajectories randomly selected out of the remaining 9954 trajectories are labeled as −1. From the target task, all the 19 trajectories with cumulative rewards above 0 are labeled as 1 and 200 trajectories randomly selected out of the remaining 9981 trajectories are labeled as −1.

For the inference problem (Problem 1), we set $\varrho_{\text{th}} = 4$ and $\zeta = 0.95$. For Algorithm 1, we set $\lambda = 0.01$ and $h_{\max} = 2$. We use the position of the robot as the state, and the atomic predicates ρ correspond to the rectangular regions in the 9×9 gridworld. For computing the information gain of MITL$_f$ formulas, we use the uniform distribution for the prior probability distribution $\mathcal{G}_L$. Following the first two steps illustrated in Section 4.2, we obtain the following satisfying formulas:

$$\phi^S = \Diamond_{[1,15]} \Box_{[0,4]} \bar{G}^S \wedge \Diamond_{[21,39]} \bar{Y}^S \text{ and}$$
$$\phi^T = \Diamond_{[5,18]} \Box_{[0,5]} \bar{G}^T \wedge \Diamond_{[24,39]} \bar{Y}^T,$$

where

$$\bar{G}^S = (x >= 3) \wedge (x <= 4) \wedge (y >= 3) \wedge (y <= 5),$$

$$\bar{Y}^S = (x >= 7) \wedge (x <= 8) \wedge (y >= 5) \wedge (y <= 8),$$

[2]Dur to space limitations, an additional case study can be found in the longer version https://bit.ly/2Zjx7mt.

$$\bar{G}^{\mathrm{T}} = (x >= 5) \wedge (x <= 6) \wedge (y >= 6) \wedge (y <= 7),$$

$$\bar{Y}^{\mathrm{T}} = (x >= 4) \wedge (x <= 7) \wedge (y >= 1) \wedge (y <= 2).$$

$\boldsymbol{\phi}^{\mathrm{S}}$ reads as "first reach $\bar{G}^{\mathrm{S}}$ during the time interval [1, 15] and stay there for 4 time units, then reach $\bar{Y}^{\mathrm{S}}$ during the time interval [21, 39]". $\boldsymbol{\phi}^{\mathrm{T}}$ reads as "first reach $\bar{G}^{\mathrm{T}}$ during the time interval [5, 18] and stay there for 5 time units, then reach $\bar{Y}^{\mathrm{T}}$ during the time interval [24, 39]". The regions $\bar{G}^{\mathrm{S}}$, $\bar{Y}^{\mathrm{S}}$, $\bar{G}^{\mathrm{T}}$ and $\bar{Y}^{\mathrm{T}}$ are shown in Figure 5 (a) (b). It can be seen that $\boldsymbol{\phi}^{\mathrm{T}}$ is structurally equivalent with $\boldsymbol{\phi}^{\mathrm{S}}$, hence logical transferability is identified.

For comparison, we also obtain $\boldsymbol{\phi}^{'\mathrm{T}}$ without considering the information gain, i.e., by setting $\lambda = 0$ in (3):

$$\phi^{'\mathrm{T}} = \Diamond_{[1,18]} \Box_{[0,4]} \bar{G}^{\mathrm{T}} \wedge \Diamond_{[24,40]} \bar{Y}^{'\mathrm{T}},$$

where

$$\bar{G}^{\mathrm{T}} = (x >= 5) \wedge (x <= 6) \wedge (y >= 6) \wedge (y <= 7),$$

$$\bar{Y}'\mathrm{T} = (x >= 0) \wedge (x <= 8) \wedge (y >= 0) \wedge (y <= 2).$$

$\boldsymbol{\phi}^{'\mathrm{T}}$ reads as "first reach $\bar{G}^{\mathrm{T}}$ during the time interval [1, 18] and stay there for 4 time units, then reach $\bar{Y}'^{\mathrm{T}}$ during the time interval [24, 40]".$\bar{G}^{\mathrm{T}}$ and $\bar{Y}'^{\mathrm{T}}$ are shown in Figure 5 (c). It can be seen that $\boldsymbol{\phi}^{\mathrm{T}}$ implies $\boldsymbol{\phi}^{'\mathrm{T}}$, hence $\boldsymbol{\phi}^{'\mathrm{T}}$ is less informative than $\boldsymbol{\phi}^{\mathrm{T}}$ with respect to the prior probability distribution $\mathscr{G}_L$.

We use Method I to refer to the Q-learning on the τ-states. In comparison with Method I, we perform Q-learning in the extended state space with the following three methods:

Method II: Q-learning with $\boldsymbol{\phi}^{'\mathrm{T}}$ (i.e., on the extended state that includes the locations and clock valuations of the timed automata constructed from $\boldsymbol{\phi}^{'\mathrm{T}}$).

Method III: Q-learning with $\boldsymbol{\phi}^{\mathrm{T}}$.

Method IV: Q-learning with $\boldsymbol{\phi}^{\mathrm{T}}$ and starting from the transferred extended Q-functions.

Figure 6 shows the learning results with the four different methods. Method I takes an average of 834590 episodes to converge to the optimal policy (with the first 50000 episodes shown in Figure 6), while Method III and Method IV take an average of 13850 episodes and 2220 episodes for convergence to the optimal policy, respectively. It should be noted that although Method II performs better than Method I in the first 50000 episodes, it does not

achieve optimal performance in 2 million episodes (as ϕ'^{T} is not sufficiently informative). In sum, the sampling efficiency for the target task is improved by up to one order of magnitude by performing RL in the extended state space with the inferred formula ϕ^{T}, and further improved by up to another order of magnitude using the transferred extended Q-functions.

6 Discussions

We proposed a transfer learning approach for temporal tasks based on logical transferability. We have shown the improvement of sampling efficiency in the target task using the proposed method.

There are several limitations of the current approach, which leads to possible directions for future work. Firstly, the proposed logical transferability is a qualitative measure of the logical similarities between the source task and the target task. Quantitative measures of logical similarities can be further established using similarity metrics between the inferred temporal logic formulas from the two tasks. Secondly, as some information about the task may not be discovered during the initial episodes of reinforcement learning (especially for more complicated tasks), the inferred temporal logic formulas can be incomplete or biased. We will develop methods for more complicated tasks by either breaking the tasks into simpler subtasks, or iteratively performing inference of temporal logic formulas and reinforcement learning as a closed loop process. Finally, we use Q-learning as the underlying learning algorithm for the transfer learning approach. The same methodology can be also applied to other forms of reinforcement learning, such as actor-critic methods or model-based reinforcement learning.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

This research was partially supported by AFOSR FA9550-19-1-0005, DARPA D19AP00004, NSF 1652113 and ONR N00014-18-1-2829.

References

Aksaray Derya, Jones Austin, Kong Zhaodan, Schwager Mac, and Belta Calin. Q-learning for robust satisfaction of signal temporal logic specifications. In IEEE CDC’16, pages 6565–6570, 12 2016.

Alshiekh Mohammed, Bloem Roderick, Ehlers Rüdiger, Könighofer Bettina, Niekum Scott, and Topcu Ufuk. Safe reinforcement learning via shielding. In AAAI’18, 2018.

Alur Rajeev and Dill David L.. A theory of timed automata. Theoretical Computer Science, 126:183–235, 1994.

Alur Rajeev, Feder Tomás, and Henzinger Thomas A.. The benefits of relaxing punctuality. J. ACM, 43(1):116–146, 1 1996.

Bombara Giuseppe, Vasile Cristian-Ioan, Penedo Francisco, Yasuoka Hirotoshi, and Belta Calin. A decision tree approach to data classification using signal temporal logic. In Proc. HSCC’16, pages 1–10, 2016.

Eberhart and Shi Yuhui. Particle swarm optimization: developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), volume 1, pages 81–86 vol. 1, 5 2001.

Fu Jie and Topcu Ufuk. Probably approximately correct MDP learning and control with temporal logic constraints. Robotics: Science and Systems, abs/1404.7073, 2014.

Glushkov Victor M.. The abstract theory of automata. Russian Mathematical Surveys, 16(5):1, 1961.

Gruber Hermann, Holzer Markus, Kiehn Astrid, and König Barbara. On timed automata with discrete time - structural and language theoretical characterization In Developments in Language Theory, pages 272–283. Springer Berlin Heidelberg, 2005.

Hoxha Bardh, Dokhanchi Adel, and Fainekos Georgios. Mining parametric temporal logic properties in model-based design for cyber-physical systems. International Journal on Software Tools for Technology Transfer, pages 79–93, 2 2017.

Kong Zhaodan, Jones Austin, and Belta Calin. Temporal logics for learning and detection of anomalous behavior. IEEE TAC, 62(3):1210–1222, 3 2017.

Kress-Gazit Hadas, Wongpiromsarn Tichakorn, and Topcu Ufuk. Correct, reactive, high-level robot control. IEEE Robotics Automation Magazine, 18(3):65–74, 9 2011.

Li Xiao, Vasile Cristian-Ioan, and Belta Calin. Reinforcement learning with temporal logic rewards. In Proc. IROS'17, pages 3834–3839, 9 2017.

Mihalkova Lilyana, Huynh Tuyen N., and Mooney Raymond J.. Mapping and revising markov logic networks for transfer learning In Proc. AAAI'07, pages 608–614, Vancouver, BC, 7 2007.

Neider Daniel and Gavran Ivan. Learning linear temporal properties. In Formal Methods in Computer Aided Design (FMCAD), pages 1–10, 2018.

Ouaknine Joël and Worrell James. On the decidability of metric temporal logic In Proc. Annual IEEE Symposium on Logic in Computer Science, LICS'05, pages 188–197, Washington, DC, USA, 2005 IEEE Computer Society.

Pnueli Amir. The temporal logic of programs In Proc. 18th Annu. Symp. Found. Computer Sci, pages 46–57, Washington, D.C., USA, 1977.

Shah Ankit, Kamath Pritish, Shah Julie A, and Li Shen. Bayesian inference of temporal task specifications from demonstrations In Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, and Garnett R, editors, NeurIPS, pages 3808–3817. Curran Associates, Inc., 2018.

Taylor Matthew E. and Stone Peter. Cross-domain transfer for reinforcement learning In Proc. ICML'07, pages 879–886, New York, NY, USA, 2007 ACM.

To Son Thanh, Roberts Mark, Apker homas, Johnson Benjamin, and Aha David W.. Mixed propositional metric temporal logic: A new formalism for temporal planning In Magazzeni D, Sanner S, & Thiebaux S (Eds.) Planning for Hybrid Systems: Papers from the AAAI Workshop (Technical Report WS-16–13), Phoenix, AZ, 2016 AAAI Press.

Icarte Rodrigo Toro, Klassen Toryn Q., Valenzano Richard, and McIlraith Sheila A.. Advice-based exploration in model-based reinforcement learning In Proc. CCAI'18, pages 72–83, 2018.

Icarte Rodrigo Toro, Klassen Toryn Q., Valenzano Richard, and McIlraith Sheila A.. Teaching multiple tasks to an RL agent using LTL In AAMAS'18, pages 452–461, Richland, SC, 2018.

Torrey Lisa and Shavlik Jude. Policy transfer via markov logic networks In De Raedt Luc, editor, Inductive Logic Programming, pages 234–248, Berlin, Heidelberg, 2010 Springer Berlin Heidelberg.

Torrey Lisa, Shavlik Jude W., Walker Trevor, and Maclin Richard. Rule extraction for transfer learning. In Rule Extraction from Support Vector Machines, pages 67–82, 2008.

Vazquez-Chanlatte Marcell, Jha Susmit, Tiwari Ashish, Ho Mark K., and Seshia Sanjit A.. Learning task specifications from demonstrations. In NeurIPS, pages 5372–5382, 2018.

Wang Zhaodong and Taylor Matthew E.. Improving reinforcement learning with confidence-based demonstrations In Proc. IJCAI'17, pages 3027–3033. AAAI Press, 2017.

Watkins Christopher J. C. H. and Dayan Peter. Q-learning. Machine Learning, 8(3):279–292, 5 1992.

Wen Min, Papusha Ivan, and Topcu Ufuk. Learning from demonstrations with high-level side information. In Proc. IJCAI'17, pages 3055–3061, 2017.

Xu Zhe and Julius Agung. Census signal temporal logic inference for multiagent group behavior analysis. IEEE Trans. Autom. Sci. Eng, 15(1):264–277, 1 2018.

Xu Zhe, Saha Sayan, Hu Botao, Mishra Sandipan, and Julius Agung. Advisory temporal logic inference and controller design for semiautonomous robots. IEEE Trans. Autom. Sci. Eng, pages 1–19, 2018.

Xu Zhe, Ornik Melkior, Julius Agung, and Topcu Ufuk. Information-guided temporal logic inference with prior knowledge. In Proc. IEEE Amer. Control Conf, 2019.

Zhuo Hankz Hankui and Yang Qiang. Action-model acquisition for planning via transfer learning. Artificial Intelligence, 212:80 – 103, 2014.

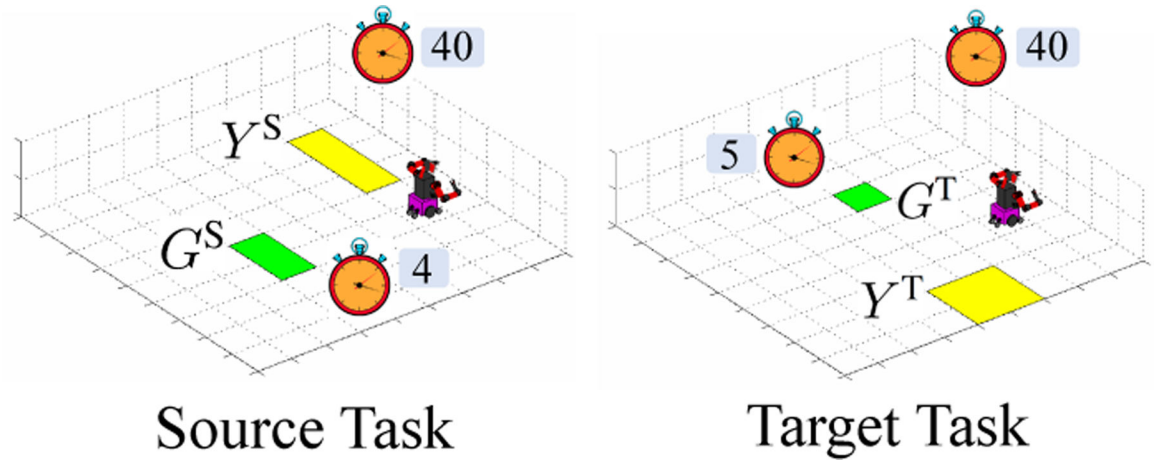

Figure 1:
An illustrative example where the source task and the target task are *logically similar*.

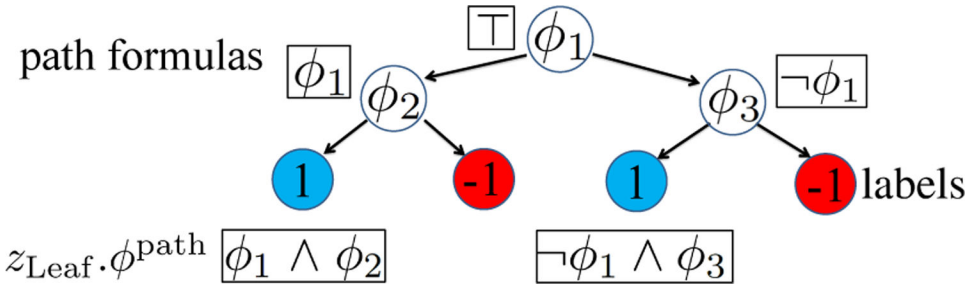

Figure 2:
Illustration of a decision tree which can be converted to an MITL$_f$ formula in the SDNF.

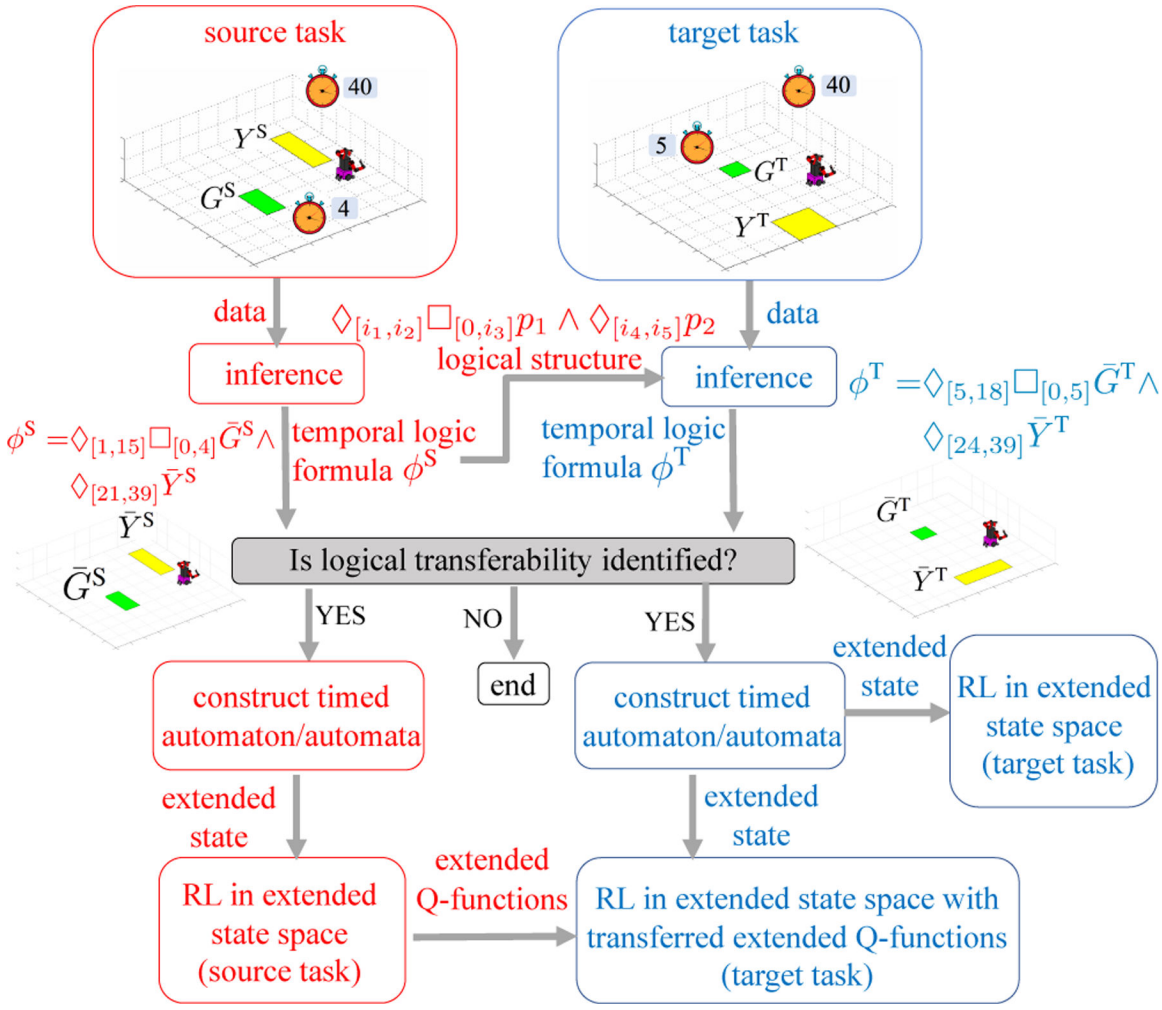

Figure 3:
Workflow diagram of the proposed transfer learning approach based on logical transferability.

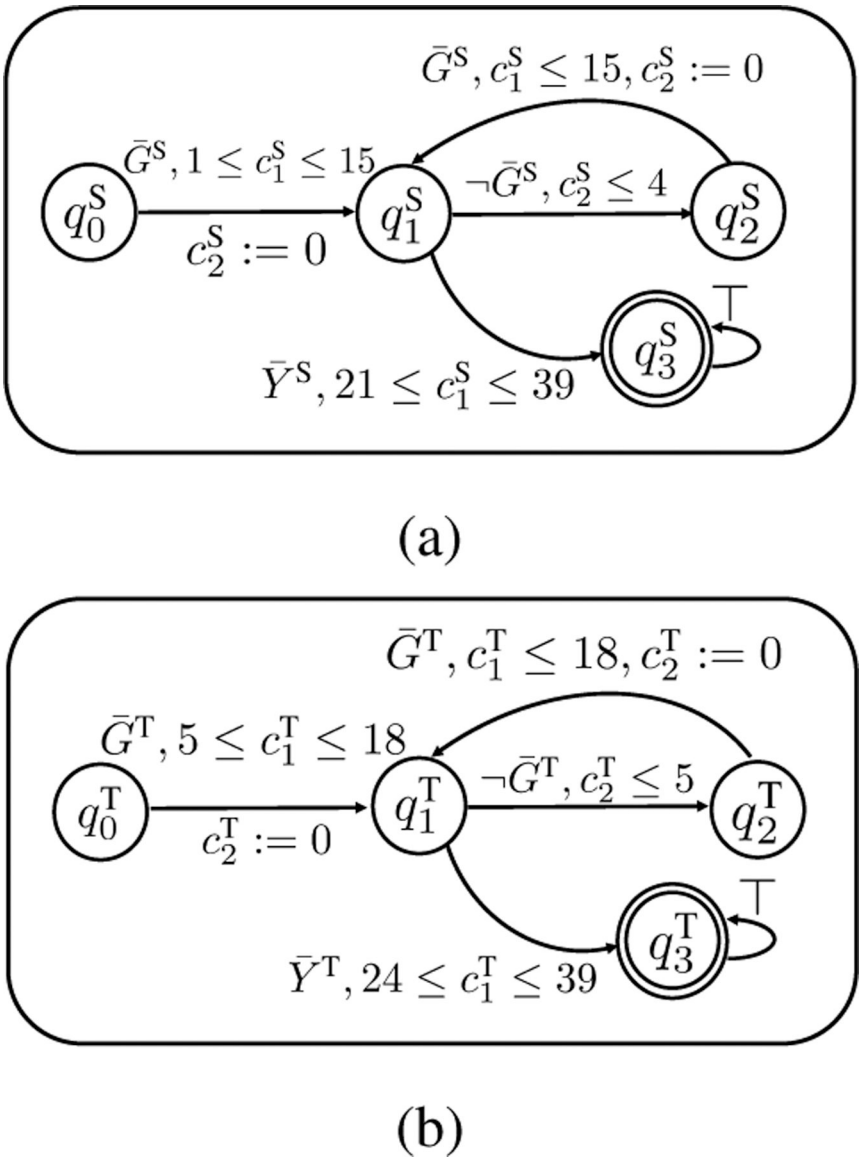

Figure 4:

The deterministic timed automata (DTA) of two structurally equivalent formulas (a) $\Diamond_{[1,15]} \Box_{[0,4]} \bar{G}^{\mathrm{S}} \wedge \Diamond_{[21,39]} \bar{Y}^{\mathrm{S}}$ and (b) $\Diamond_{[5,18]} \Box_{[0,5]} \bar{G}^{\mathrm{T}} \wedge \Diamond_{[24,39]} \bar{Y}^{\mathrm{T}}$. The locations q_0^{S}, q_1^{S}, q_2^{S} and q_3^{S} correspond to q_0^{T}, q_1^{T}, q_2^{T} and q_3^{T}, respectively. The atomic predicate $\bar{G}^{\mathrm{S}}$ and $\bar{Y}^{\mathrm{S}}$ correspond to $\bar{G}^{\mathrm{T}}$ and $\bar{Y}^{\mathrm{T}}$, respectively. The clocks c_1^{S} and c_2^{S} correspond to c_1^{T} and c_2^{T}, respectively.

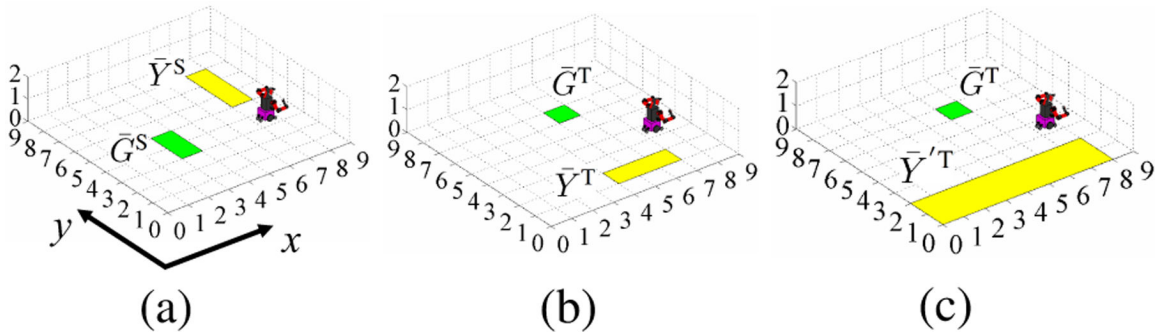

Figure 5:
Inferred regions in the case study.

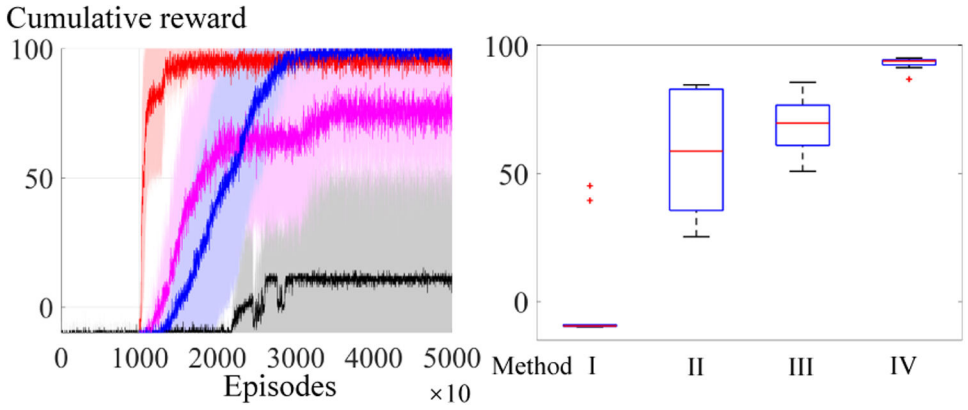

Figure 6:

Learning results in the case study: cumulative rewards of 10 independent simulation runs averaged for every 10 episodes (left) and boxplot of the 10 runs for the average cumulative rewards of 40000 episodes after the data collection phase (right). Black: Method I; magenta: Method II; blue: Method III; red: Method IV.