Adaptive Teaching of Temporal Logic Formulas to Preference-based Learners

Zhe Xu[1], **Yuxin Chen**[2], **Ufuk Topcu**[3]
[1]Arizona State University
[2]University of Chicago
[3]University of Texas at Austin
xzhe1@asu.edu, chenyuxin@uchicago.edu, utopcu@utexas.edu

Abstract

Machine teaching is an algorithmic framework for teaching a target hypothesis via a sequence of examples or demonstrations. We investigate machine teaching for *temporal logic formulas*—a novel and expressive hypothesis class amenable to time-related *task specifications*. In the context of teaching temporal logic formulas, an exhaustive search even for a *myopic* solution takes exponential time (with respect to the time span of the task). We propose an efficient approach for teaching *parametric linear temporal logic formulas*. Concretely, we derive a necessary condition for the minimal time length of a demonstration to eliminate a set of hypotheses. Utilizing this condition, we propose an efficient myopic teaching algorithm by solving a sequence of integer programming problems. We further show that, under two notions of *teaching complexity*, the proposed algorithm has near-optimal performance. We evaluate our algorithm extensively under different classes of learners (i.e., learners with different preferences over hypotheses) and interaction protocols (e.g., non-adaptive and adaptive). Our results demonstrate the effectiveness of the proposed algorithm in teaching temporal logic formulas; in particular, we show that there are significant gains of teaching efficacy when the teacher adapts to feedback of the learner, or adapts to a (non-myopic) oracle.

1 Introduction

Machine teaching, also known as algorithmic teaching, is an algorithmic framework for teaching a target hypothesis via a sequence of examples or demonstrations (Zhu 2015). Due to limited availability of data or high cost of data collection in real-world learning scenarios, machine teaching provides a viable solution for optimizing the training data for a learner to efficiently learn a target hypothesis.

Recently, there has been an increasing interest in designing learning algorithms for inferring *task specifications* from data (e.g., in robotics) (Kong, Jones, and Belta 2017; Vazquez-Chanlatte et al. 2018). Machine teaching can be used to optimize the training data for various learning algorithms (Dasgupta et al. 2019), hence it can be potentially used for task specification inference algorithms. Machine teaching can be also used in adversarial settings (Ma et al. 2019) where an attacker (machine teaching algorithm) manipulates specification inference by modifying the training data. Unfortunately, finding the optimal teaching sequence with minimal *teaching cost* is notoriously hard (Goldman and Kearns 1995). As task specifications are often time-related and the demonstration data are trajectories with a time evolution, an exhaustive search even for the *myopic* solution (i.e., one-step optimal demonstration from the greedy algorithm) has exponential time complexity with the time span of the task, making it prohibitive to run existing myopic teaching algorithms in practice.

In this paper, we investigate machine teaching of target hypothesis represented in *temporal logic* (Pnueli 1977), which has been used to express task specifications in many applications in robotics and artificial intelligence (Kress-Gazit, Wongpiromsarn, and Topcu 2011; To et al. 2015). Specifically, we use a fragment of *parametric linear temporal logic* (pLTL) (Chakraborty and Katoen 2014; Alur et al. 2001). We derive a necessary condition for the minimal time length of a demonstration so that a set of pLTL formulas can be eliminated by this demonstration. Utilizing this necessary condition, we provide a *myopic teaching* approach by solving a sequence of integer programming problems which, under certain conditions, guarantees a logarithmic factor of the optimal teaching cost.

We evaluate the proposed algorithm extensively under a variety of learner types (i.e., learner with different preference models) and interactive protocols (i.e., batched and adaptive). The results show that the proposed algorithm can efficiently teach a given target hypothesis under various settings, and that there are significant gains of teaching efficacy when the teacher adapts to the learner's current hypotheses (up to 31.15% reduction in teaching cost compared to the non-adaptive setting) or with oracles (up to 75% reduction in teaching cost compared to the myopic setting).

Related Work

There has been a surge of interest in machine teaching in several different application domains, including personalized educational systems (Zhu 2015), citizen sciences (Chen et al. 2018a; Mac Aodha et al. 2018), adversarial attacks (Ma et al. 2019) and imitation learning (Brown and Niekum 2019). Most theoretical work in algorithmic machine teaching assumes the version space model (Goldman and Kearns 1995; Gao et al. 2017; Chen et al. 2018b; Mansouri et al. 2019). Recently, some teaching complexity results have been extended beyond version space learners, such as Bayesian learners (Zhu 2013) and gradient learners (Liu et al. 2017) (e.g., learners implementing a gradient-based optimization algorithm). However, most algorithms are restricted to simple concept classes. In this work, we aim to understand the complexity of teaching the class of pLTL formulas. There has been extensive study in modeling a learner for temporal logic formulas, for example, see (Hoxha, Dokhanchi, and Fainekos 2017; Kong, Jones, and Belta 2017; Xu et al. 2019a; Bombara et al. 2016; Xu et al. 2019b; Neider and Gavran 2018; Yan, Xu, and Julius 2019; Xu and Julius 2018; Vazquez-Chanlatte et al. 2018; Xu et al. 2019; Shah et al. 2018), while it is much less understood in the context of machine teaching. In this paper, we abstract these learner models as preference-based version space learners, and focus on developing efficient algorithms for *teaching* such learners.

2 Parametric Linear Temporal Logic

In this section, we present an overview of parametric linear temporal logic (pLTL) (Chakraborty and Katoen 2014; Alur et al. 2001). We start with the syntax and semantics of pLTL. The domain $\mathbb{B} = \{\top, \bot\}$ ($\top$ and $\bot$ represents True and False respectively) is the Boolean domain and the time index set $\mathbb{T} = \{0, 1, \dots\}$ is a discrete set of natural numbers. We assume that there is an underlying system $\mathcal{H}$. The state s of the system $\mathcal{H}$ belongs to a finite set S of states. A trajectory $\rho_L = s_0 s_1 \cdots s_{L-1}$ of length $L \in \mathbb{Z}_{>0}$ describing an evolution of the system $\mathcal{H}$ is a function from $\mathbb{T}$ to S, and we denote $\rho_L(t) := s_t$. A set $\mathcal{AP} = \{\pi_1, \pi_2, \dots, \pi_n\}$ is a set of atomic predicates. $\mathcal{L} : S \to 2^{\mathcal{AP}}$ is a function assigning a subset of atomic predicates in $\mathcal{AP}$ to each state $s \in S$. The syntax of the (F,G)-fragment bounded pLTL is defined recursively as[1]

$$\phi := \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid G_{\leq\tau}\phi \mid F_{\leq\tau}\phi,$$

where π is an *atomic predicate*; $\neg$ and $\wedge$ stand for negation and conjunction, respectively; $G_{\leq\tau}$ and $F_{\leq\tau}$ are temporal operators representing "parameterized always" and "parameterized eventually", respectively ($\tau \in \mathbb{T}$ is a temporal parameter). From the above-mentioned operators, we can also derive other operators such as $\vee$ (disjunction) and $\Rightarrow$ (implication). In the following content of the paper, we refer to (F,G)-fragment bounded pLTL as pLTL$_f$ for brevity.

Next, we introduce the Boolean semantics of a pLTL$_f$ formula in the strong and the weak view (Eisner et al. 2003; Ho, Ouaknine, and Worrell 2014). In the following, $(\rho_L, t) \models_{\mathrm{S}} \phi$ (resp. $(\rho_L, t) \models_{\mathrm{W}} \phi$) means the trajectory ρ_L strongly (resp. weakly) satisfies ϕ at time t, and $(\rho_L, t) \not\models_{\mathrm{S}} \phi$ (resp. $(\rho_L, t) \not\models_{\mathrm{W}} \phi$) means the trajectory ρ_L fails to strongly (resp. weakly) satisfy ϕ at time t.

Definition 1. *The Boolean semantics of the pLTL$_f$ in the strong view is defined recursively as*

$$\begin{aligned}
(\rho_L, t) &\models_{\mathrm{S}} \pi \textit{ iff } t \leq L-1 \textit{ and } \pi \in \mathcal{L}(\rho_L(t)),\\
(\rho_L, t) &\models_{\mathrm{S}} \neg\phi \textit{ iff } (\rho_L, t) \not\models_{\mathrm{W}} \phi,\\
(\rho_L, t) &\models_{\mathrm{S}} \phi_1 \wedge \phi_2 \textit{ iff } (\rho_L, t) \models_{\mathrm{S}} \phi_1 \textit{ and } (\rho_L, t) \models_{\mathrm{S}} \phi_2,\\
(\rho_L, t) &\models_{\mathrm{S}} F_{\leq\tau}\phi \textit{ iff } \exists t' \in [t, t+\tau], s.t.\ (\rho_L, t') \models_{\mathrm{S}} \phi,\\
(\rho_L, t) &\models_{\mathrm{S}} G_{\leq\tau}\phi \textit{ iff } (\rho_L, t') \models_{\mathrm{S}} \phi, \forall t' \in [t, t+\tau].
\end{aligned}$$

Definition 2. *The Boolean semantics of the pLTL$_f$ in the weak view is defined recursively as*

$$\begin{aligned}
(\rho_L, t) &\models_{\mathrm{W}} \pi \textit{ iff either } t > L-1,\\
&\qquad \textit{or } \big(t \leq L-1 \textit{ and } \pi \in \mathcal{L}(\rho_L(t))\big),\\
(\rho_L, t) &\models_{\mathrm{W}} \neg\phi \textit{ iff } (\rho_L, t) \not\models_{\mathrm{S}} \phi,\\
(\rho_L, t) &\models_{\mathrm{W}} \phi_1 \wedge \phi_2 \textit{ iff } (\rho_L, t) \models_{\mathrm{W}} \phi_1 \textit{ and } (\rho_L, t) \models_{\mathrm{W}} \phi_2,\\
(\rho_L, t) &\models_{\mathrm{W}} F_{\leq\tau}\phi \textit{ iff } \exists t' \in [t, t+\tau], s.t.\ (\rho_L, t') \models_{\mathrm{W}} \phi,\\
(\rho_L, t) &\models_{\mathrm{W}} G_{\leq\tau}\phi \textit{ iff } (\rho_L, t') \models_{\mathrm{W}} \phi, \forall t' \in [t, t+\tau].
\end{aligned}$$

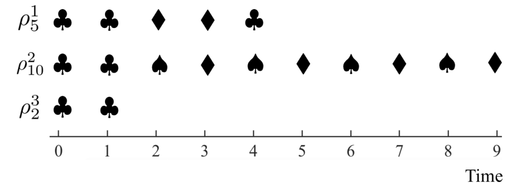

Figure 1: Three trajectories of different lengths.

Intuitively, if a trajectory of finite length can be extended to infinite length, then the strong view indicates that the truth value of the formula on the infinite-length trajectory is already "determined" on the trajectory of finite length, while the weak view indicates that it may not be "determined" yet. As shown in the simple example in Fig. 1, with the set $S = \{\clubsuit, \spadesuit, \blacklozenge\}$ of states and three trajectories ρ^1_5, ρ^2_{10} and ρ^3_2, $F_{\leq 4}\clubsuit$ is strongly satisfied by all three trajectories, while $F_{\leq 4}\spadesuit$ is strongly violated by ρ^1_5, strongly satisfied by ρ^2_{10}, and weakly satisfied (and also weakly violated) by ρ^3_2.

3 Teaching pLTL$_f$ Formulas with Demonstrations of Varying Lengths

We now provide the framework for teaching pLTL$_f$ formulas with *demonstrations* of varying lengths.

Teaching Model

Let $\mathcal{Z} := S^+ \times \{-1, 1\}$ be the ground set of *demonstrations* (labeled trajectories), where S^+ denotes the *Kleene plus* of S, and $\{-1, 1\}$ is the set of labels. There is a *hypothesis set* $\Phi = \{\phi_1, \dots, \phi_n\}$ consisting of $n \in \mathbb{Z}_{>0}$ hypothesis pLTL$_f$ formulas. The teacher knows a *target hypothesis* $\phi^* \in \Phi$ and intends to teach ϕ^* to a learner.

We define the *preference function* $\sigma : \Phi \times \Phi \to \mathbb{R}_{>0}$ as a function that encodes the learner's transition preferences. Specifically, given the current hypothesis ϕ and any two hypotheses ϕ' and ϕ'', ϕ' is preferred to ϕ'' from ϕ if and only

[1]Although other temporal operators such as "Until "($\mathcal{U}$) may also appear in the full syntax of pLTL, they are omitted from the syntax of (F,G)-fragment bounded pLTL as they can be hard to interpret and are not often used for the inference of pLTL formulas.

if $\sigma(\phi';\phi) < \sigma(\phi'';\phi)$. If for any ϕ', $\sigma(\phi';\phi)$ does not depend on ϕ, then σ is a *global* preference function; otherwise, σ is a *local* preference function. If $\sigma(\phi';\phi)$ is a constant for any ϕ and ϕ', then σ is a *uniform* preference function.

Definition 3. *Given a target hypothesis ϕ^* and a demonstration $[\rho_L, l]$, where $l = 1$ represents positive demonstration and $l = -1$ represents negative demonstration, $[\rho_L, l]$ is strongly inconsistent with a pLTL$_f$ formula $\phi \in \Phi$, if and only if the following condition is satisfied:*

$$\begin{cases}(\rho_L, 0) \models_{\mathrm{S}} \phi^*, \ (\rho_L, 0) \models_{\mathrm{S}} \neg\phi, & \text{if } l = 1;\\ (\rho_L, 0) \models_{\mathrm{S}} \neg\phi^*, \ (\rho_L, 0) \models_{\mathrm{S}} \phi, & \text{if } l = -1.\end{cases}$$

Starting from the teacher's first demonstration, the teaching stops as soon as the learner's current hypothesis reaches the target hypothesis, and we call it a *teaching session*. For a sequence D of demonstrations, we define the *version space* induced by D, denoted as $\Phi(D)$, as the subset of pLTL$_f$ formulas in Φ that are *not* strongly inconsistent with any demonstration in D. We use Θ to denote the random variable representing the randomness of the environment (e.g., learner's random choice of next hypothesis in the presence of ties) and θ to denote the *realization* of Θ in each teaching session.

Definition 4. *We define a teaching setting as a 4-tuple $\mathcal{S} = (\phi^*, \phi^0, \Phi, \mathrm{dom}(\theta))$, where ϕ^* is the target hypothesis, $\phi^0 \neq \phi^*$ is the learner's initial hypothesis, Φ is the hypothesis set and $\mathrm{dom}(\theta)$ is the domain of the realization θ for the randomness of the environment.*

Let $\mathcal{T}$ be a teacher that outputs a sequence of demonstrations based on the target hypothesis, the version space, and either the learner's initial hypothesis (non-adaptive teacher) or the learner's current hypothesis (adaptive teacher). For a learner with preference function σ, let $\mathcal{L}_\sigma : \Phi \times 2^\Phi \times \mathcal{Z} \times \mathrm{dom}(\theta) \rightarrow \Phi$ be the learner's mapping that maps the learner's current hypothesis, the current version space, the current demonstration and the randomness of the environment to the learner's next updated hypothesis. Given a teacher $\mathcal{T}$, a learner $\mathcal{L}_\sigma$ and the randomness θ in the teaching setting $\mathcal{S} = (\phi^*, \phi^0, \Phi, \mathrm{dom}(\theta))$, we use $D_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma, \theta)$ to denote the sequence of demonstrations provided by the teacher $\mathcal{T}$ before the learner's hypothesis reaches ϕ^*.

Definition 5. *The accumulated-number (AN) teaching cost and accumulated-length (AL) teaching cost are defined as*

$$\begin{aligned}&\textit{AN-Cost}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma, \theta) := |D_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma, \theta)|,\\ &\textit{AL-Cost}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma, \theta) := \sum_{(\rho^k_{L_k}, l_k) \in D_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma, \theta)} L_k.\end{aligned}$$

Intuitively, the AN teaching cost and the AL teaching cost are the number of demonstrations and the accumulated time lengths of the demonstrations for the learner's hypothesis to reach ϕ^* in a specific teaching session, respectively.

Definition 6. *Given a teacher $\mathcal{T}$ and a learner $\mathcal{L}_\sigma$ in the teaching setting $\mathcal{S} = (\phi^*, \phi^0, \Phi, \mathrm{dom}(\theta))$, we define the worst-case AN teaching cost and AL teaching cost as*

$$\begin{aligned}&\textit{AN-Cost}^{\textit{WC}}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma) := \max_{\theta \in \mathrm{dom}(\theta)} \textit{AN-Cost}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma, \theta),\\ &\textit{AL-Cost}^{\textit{WC}}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma) := \max_{\theta \in \mathrm{dom}(\theta)} \textit{AL-Cost}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma . \theta).\end{aligned}$$

Definition 7. *Given a learner $\mathcal{L}_\sigma$ in the teaching setting $\mathcal{S} = (\phi^*, \phi^0, \Phi, \mathrm{dom}(\theta))$, we define the AN teaching complexity and AL teaching complexity respectively as*

$$\begin{aligned}&\textit{AN-Complexity}_{\mathcal{S}}(\mathcal{L}_\sigma) := \min_{\mathcal{T}} \textit{AN-Cost}^{\textit{WC}}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma),\\ &\textit{AL-Complexity}_{\mathcal{S}}(\mathcal{L}_\sigma) := \min_{\mathcal{T}} \textit{AL-Cost}^{\textit{WC}}_{\mathcal{S}}(\mathcal{T}, \mathcal{L}_\sigma).\end{aligned}$$

Intuitively, the AN teaching complexity and the AL teaching complexity are the minimal number of demonstrations and the minimal accumulated time lengths of the demonstrations needed for the learner's hypothesis to reach ϕ^* despite the randomness of the environment, respectively.

For example, we consider the hypothesis set $\Phi = \{F_{\leq i} s\}$, where $i \in \{0, \dots, 4\}$, and $s \in S = \{\clubsuit, \spadesuit, \blacklozenge\}$. If ϕ^* is $F_{\leq 2}\clubsuit$, and the learner has uniform preference, then for any ϕ^0, one sequence of demonstrations which can minimize both the (worst-case) AN and AL teaching costs are firstly a negative demonstration of $\spadesuit, \blacklozenge, \spadesuit, \clubsuit, \clubsuit$, and then a positive demonstration of $\spadesuit, \blacklozenge, \clubsuit$. In this teaching setting, the AN and AL teaching complexities are 2 and 8, respectively.

TLIP: Teaching of pLTL$_f$ Formulas with Integer Programming

Finding the optimal sequence of demonstrations with minimal AN or AL teaching cost has time complexity in the order of $2^{|D_{L_{\max}}|}$, where $D_{L_{\max}}$ is the set of all possible demonstrations with length at most $L_{\max}$. As $|D_{L_{\max}}| = \sum_{i=1}^{L_{\max}} |S|^{L_{\max}}$, $2^{|D_{L_{\max}}|}$ is double exponential with the maximal length of the demonstrations.

We resort to greedy methods for *myopic teaching* with near-optimal performance. To compute the greedy solution, we first derive a necessary condition through Definition 8 and Theorem 1 for the minimal time length of a demonstration so that a set of pLTL$_f$ formulas are strongly inconsistent with (thus can be eliminated by) this demonstration.

Definition 8. *We define the minimal time length $\zeta(\phi, l)$ of a pLTL$_f$ formula ϕ with respect to a label l recursively as*

$$\begin{aligned}\zeta(\pi, l) =& 0, \quad \zeta(\neg\phi, l) = \zeta(\phi, -l),\\ \zeta(\phi_1 \wedge \phi_2, l) =& \begin{cases}\max\{\zeta(\phi_1, l), \zeta(\phi_2, l)\}, & \text{if } l = 1;\\ \min\{\zeta(\phi_1, l), \zeta(\phi_2, l)\}, & \text{if } l = -1,\end{cases}\\ \zeta(F_{\leq\tau}\phi, l) =& \begin{cases}\zeta(\phi, l), & \text{if } l = 1;\\ \zeta(\phi, l) + \tau, & \text{if } l = -1,\end{cases}\\ \zeta(G_{\leq\tau}\phi, l) =& \begin{cases}\zeta(\phi, l) + \tau, & \text{if } l = 1;\\ \zeta(\phi, l), & \text{if } l = -1.\end{cases}\end{aligned}$$

Theorem 1. *Given a target hypothesis ϕ^* and the hypothesis set Φ, if a demonstration $[\rho_L, l]$ is strongly inconsistent with a subset $\hat{\Phi} = \{\phi_i\}_{i=1}^{\hat{N}} \subset \Phi$ of pLTL$_f$ formulas, then*

$$L \geq \max\{\zeta(\phi^*, l), \max_{1 \leq i \leq \hat{N}} \zeta(\phi_i, -l)\}.$$

Algorithm 1 shows the proposed TLIP approach for teaching pLTL$_f$ formulas to learners with preferences. Here

Algorithm 1: Teaching of pLTL$_f$ Formulas with Integer Programming (TLIP)

1 **Input**: hypothesis set Φ, initial hypothesis ϕ^0
2 Initialize $k \leftarrow 0$, $\tilde{\Phi} \leftarrow \emptyset$, $\Phi^0 \leftarrow \Phi$
3 **while** $\phi^k \neq \phi^*$ **do**
4   **if** *MyopicTeacher* $= 1$ **then** $\phi^{k*} = \phi^*$
5   **else** Compute $\phi^{k*} \leftarrow$ Oracle(ϕ^k, Φ^k, ϕ^*)
6   **if** σ *is global* **then**
7         $\tilde{\Phi} \leftarrow \{\phi \in \Phi^k : \sigma(\phi; \cdot) \leq \sigma(\phi^{k*}; \cdot)\}$
8   **else**
9         $\tilde{\Phi} \leftarrow \{\phi \in \Phi^k : \sigma(\phi; \phi') \leq \sigma(\phi^{k*}; \phi')$ for some $\phi' \in \Phi^k\}$
10     $(\rho^k, \ell^k), \hat{\Phi} \leftarrow$ComputeDemonstration($\Phi^k$, $\tilde{\Phi}$, ϕ^{k*})
11     $\Phi^{k+1} \leftarrow \Phi^k \setminus \hat{\Phi}$, $\tilde{\Phi} \leftarrow \tilde{\Phi} \setminus \hat{\Phi}$, $k \leftarrow k+1$
12   **if** *AdaptiveTeacher* $= 0$ **then**
13   **if** $\tilde{\Phi} = \{\phi^*\}$ **then** $\phi^k \leftarrow \phi^*$
14   **else** Randomly select $\phi^k \neq \phi^*$ from $\tilde{\Phi}$
15   **else** Observe the learner's next hypothesis ϕ^k
16 $K \leftarrow k - 1$
17 **return** $\{(\rho^0, \ell^0), (\rho^1, \ell^1), \ldots, (\rho^K, \ell^K)\}$

Algorithm 2: ComputeDemonstration

1 **Input** : $\Phi, \tilde{\Phi}, \phi^*$
2 Compute ρ^*_{pos} and $\kappa(\rho^*_{\text{pos}})$ for IP$_{\text{pos}}(\tilde{\Phi}, \phi^*)$
3 Compute ρ^*_{neg} and $\kappa(\rho^*_{\text{neg}})$ for IP$_{\text{neg}}(\tilde{\Phi}, \phi^*)$
4 **if** $\kappa(\rho^*_{pos}) \geq \kappa(\rho^*_{neg})$ **then**
5     $(\rho, \ell) \leftarrow (\rho^*_{\text{pos}}, 1)$, $\hat{\Phi} \leftarrow \{\phi \in \Phi : c(\phi, \rho) = -1\}$
6 **else**
7     $(\rho, \ell) \leftarrow (\rho^*_{\text{neg}}, -1)$, $\hat{\Phi} \leftarrow \{\phi \in \Phi : c(\phi, \rho) = 1\}$
8 **return** (ρ, ℓ), $\hat{\Phi}$

we focus on the myopic solution (*MyopicTeacher* $= 1$) under the *non-adaptive* setting (*AdaptiveTeacher* $= 0$) where the teacher does not observe the learner's current hypothesis and provides the sequence of demonstrations based on the learner's initial hypothesis. We compute $\tilde{\Phi}$ as the set of hypotheses that are preferred over the target hypothesis in the current version space if the learner has global preferences (Line 7) and the union of the sets of hypotheses that are preferred over the target hypothesis based on each hypothesis in the current version space if the learner has local preferences (Line 9). Then we call Algorithm 2 to compute the demonstrations that achieve the greedy myopic solution (Line 10).

Note that finding the greedy myopic solution via exhaustive search amounts to traversing the space of demonstrations, which is exponential with the maximal length of the demonstrations. We propose to find the greedy solution via a novel integer programming (IP) formulation. For a trajectory ρ_L and a pLTL$_f$ formula ϕ, we denote $c(\phi, \rho_L) = 1$ if $(\rho_L, 0) \models_{\text{S}} \phi$; $c(\phi, \rho_L) = -1$ if $(\rho_L, 0) \models_{\text{S}} \neg\phi$; and $c(\phi, \rho_L) = 0$ if $(\rho_L, 0) \not\models_{\text{S}} \phi$ and $(\rho_L, 0) \not\models_{\text{S}} \neg\phi$. For positive demonstrations, we compute the following integer programming problem IP$_{\text{pos}}(\tilde{\Phi}, \phi^*)$.

$$\max_{\rho_L} \ \kappa(\rho_L)$$

$$\text{subject to: } b_j \in \{0,1\}, \forall j, \text{ s.t. } \phi_j \in \tilde{\Phi}, \ \ c(\phi^*, \rho_L) = 1,$$
$$c(\phi_j, \rho_L) = 1 - 2b_j, \forall j, \text{ s.t. } \phi_j \in \tilde{\Phi},$$
$$L \geq \zeta(\phi^*, 1), L \geq b_j \zeta(\phi_j, -1), \forall j, \text{ s.t. } \phi_j \in \tilde{\Phi},$$

where $\kappa(\rho_L) = \left(\sum_{\phi_j \in \tilde{\Phi}} b_j\right)$ when optimizing for the AN teaching cost and $\kappa(\rho_L) = \left(\sum_{\phi_j \in \tilde{\Phi}} b_j\right)/L$ when optimizing for the AL teaching cost, the strong satisfaction or strong violation of a pLTL$_f$ formula ϕ by ρ_L can be encoded as integer linear constraints of ρ_L, and the constraints for L are obtained from Theorem 1. In practice, the problem IP$_{\text{pos}}(\tilde{\Phi}, \phi^*)$ can be efficiently solved by highly-optimized IP solvers (Gurobi 2019), which, as demonstrated in the case studies in later sections, is significantly more efficient than the exhaustive search method.

For negative demonstrations, the integer programming problem IP$_{\text{neg}}(\tilde{\Phi}, \phi^*)$ can be similarly formulated with the constraints $c(\phi^*, \rho_L) = -1$ and $c(\phi_j, \rho_L) = -1 + 2b_j, \forall j, \text{ s.t. } \phi_j \in \tilde{\Phi}$. We use ρ^*_{pos} and ρ^*_{neg} to denote the optimal positive and negative demonstrations computed from IP$_{\text{pos}}(\tilde{\Phi}, \phi^*)$ and IP$_{\text{neg}}(\tilde{\Phi}, \phi^*)$, respectively. We select ρ^*_{pos} or ρ^*_{neg} depending on whether $\kappa(\rho^*_{\text{pos}})$ is no less than $\kappa(\rho^*_{\text{neg}})$ or not. Then, we eliminate the hypothesis pLTL$_f$ formulas that are strongly inconsistent with the selected demonstration (Line 11). For non-adaptive teaching, we randomly select a pLTL$_f$ formula different from the target hypothesis (Line 14, as we consider the worst case) and perform another round of computation for the demonstration until the current hypothesis reaches the target hypothesis.

Teaching with Positive Demonstrations Only

Learning temporal logic formulas from positive demonstrations is a typical problem in temporal logic inference (Xu et al. 2019a; Vazquez-Chanlatte et al. 2018).

The algorithm for teaching pLTL$_f$ formulas with positive demonstrations only can be modified from Algorithms 1 and 2, by deleting Lines 3-7 of Algorithm 2 and obtaining (ρ, l) as $(\rho^*_{\text{pos}}, 1)$. The following theorem provides a necessary condition for teaching a pLTL$_f$ formula with positive demonstrations to a learner with global preferences.

Theorem 2. *Given a hypothesis set Φ and a sequence of positive demonstrations D_p, if a target hypothesis $\phi^* \in \Phi$ is teachable from D_p to a learner with global preference function σ, i.e., $\forall \phi \in \Phi(D_p) \setminus \{\phi^*\}, \sigma(\phi; \cdot) > \sigma(\phi^*; \cdot)$, then $\max_{1 \leq k \leq |D_p|} L_k \geq \max_{\phi_i \in \tilde{\Phi} \setminus \{\phi^*\}} \zeta(\phi_i, -1)$. Furthermore, there does not exist a pLTL$_f$ formula ϕ' with $\sigma(\phi'; \cdot) \leq \sigma(\phi^*; \cdot)$ s.t. $\phi^* \Rightarrow \phi'$. Here, $\tilde{\Phi} := \{\phi \in \Phi : \sigma(\phi; \cdot) \leq \sigma(\phi^*; \cdot)\}$, L_k is the time length of the k-th demonstration in D_p.*

4 Adaptive Teaching of pLTL$_f$ Formulas

We now explore the theoretical aspects of machine teaching for pLTL$_f$ formulas under the adaptive setting.

Teaching Complexity

Different from the non-adaptive teaching, an adaptive teacher observes the learner's current hypothesis and provides the next demonstration according to the target hypothesis, the current version space and the learner's current hypothesis. Algorithm 1 with *AdaptiveTeacher* = 1 shows the procedure for adaptive teaching using TLIP.

The following theorem, as adapted from Chen et al. (2018b), provides near-optimality guarantees under the adaptive myopic setting.

Theorem 3. *We denote the myopic adaptive teacher in TLIP as $\mathcal{T}^m$. Given a target hypothesis ϕ^* and the hypothesis set Φ, then*

$$\textit{AN-Cost}^{WC}_{\mathcal{S}}(\mathcal{T}^m, \mathcal{L}_\sigma) \leq \lambda(\log|\tilde{\Phi}_{\mathcal{S}}| + 1)\textit{AN-Complexity}_{\mathcal{S}}(\mathcal{L}_\sigma),$$
$$\textit{AL-Cost}^{WC}_{\mathcal{S}}(\mathcal{T}^m, \mathcal{L}_\sigma) \leq \lambda(\log|\tilde{\Phi}_{\mathcal{S}}| + 1)\textit{AL-Complexity}_{\mathcal{S}}(\mathcal{L}_\sigma),$$

where $\tilde{\Phi}_{\mathcal{S}} := \{\phi \in \Phi : \sigma(\phi;\phi^0) \leq \sigma(\phi^*;\phi^0)\}$, $\lambda = 1$ if σ is global, and $\lambda = 2$ if σ is local and the following two conditions are satisfied for both σ and the sequence D of demonstrations.

$$\begin{cases} 1.\forall \phi', \phi'' \in \Phi, \sigma(\phi';\phi) \leq \sigma(\phi'';\phi) \leq \sigma(\phi^*;\phi) \\ \qquad\qquad \Rightarrow \sigma(\phi'';\phi') \leq \sigma(\phi^*;\phi'); \\ 2.\forall \Phi' \subset \bar{\Phi}(\{[\rho_L, l]\}), \exists [\rho'_{L'}, l'] \in D, s.t., \bar{\Phi}(\{[\rho'_{L'}, l']\}) = \Phi'. \end{cases}$$

In Condition 2, $\bar{\Phi}(\{[\rho_L, l]\})$ denotes the set of hypotheses in Φ which are strongly inconsistent with demonstration $[\rho_L, l]$.

Theorem 3 shows that, under the adaptive myopic setting, TLIP can achieve near-optimal performance for global preferences and certain local preferences that satisfy Conditions 1 and 2. This motivates us to design intermediate target hypotheses as shown in the case studies.

Teaching with Oracles

For learners with local preferences, we can design intermediate target hypotheses so that Condition 1 of Theorem 3 can be satisfied for learning each target hypothesis. In Algorithm 1, we assume that we have access to an *oracle*, i.e., Oracle(ϕ^k, Φ^k, ϕ^*), which outputs an intermediate target hypothesis ϕ^{k*} at each step.

As an example, we consider the following local preference of the learner: the learner prefers formulas with the same temporal operator as that in ϕ^c if the learner's current hypothesis ϕ^c does not contain ♣; the learner prefers G-formulas to F-formulas if ϕ^c contains ♣; and with the same temporal operator, the learner prefers formulas with ♠ to formulas with ♣, and prefers formulas with ♦ the least. Then, for $\phi^* = G_{\leq 2}♣$, $\phi^1 = F_{\leq 3}♠$, $\phi^2 = F_{\leq 5}♣$, $\phi^3 = F_{\leq 3}♦$, we have $\sigma(\phi^2;\phi^1) < \sigma(\phi^3;\phi^1) < \sigma(\phi^*;\phi^1)$, but $\sigma(\phi^*;\phi^2) < \sigma(\phi^3;\phi^2)$. Therefore, Condition 1 of Theorem 3 does not hold. If the oracle outputs ϕ^2 as an intermediate target hypothesis before the learner's hypothesis reaches ϕ^2, then the teaching problem is decomposed into two subproblems, i.e., teaching before ϕ^2 and after ϕ^2, and both subproblems satisfy Condition 1 of Theorem 3.

5 Case Study: Numerical Example

In this section, we evaluate the proposed approach under different teaching settings. The hypothesis set of pLTL$_f$ formulas are listed in Table 1. The set S of states is $\{0, 1, \ldots, 10\}$. We randomly select both the initial hypothesis and the target hypothesis from the hypothesis set. The results are averaged over 10 teaching sessions, with the standard deviations listed in the supplementary material.

Teaching pLTL$_f$ Formulas under Global Preferences

We first compare TLIP with the exhaustive search method for myopic teaching (ESMT). Table 2 shows the computation time for TLIP using myopic teaching and ESMT (minimizing the AL teaching costs) for learners with global (uniform) preferences, where timeout (TO) is 300 minutes (on a MacBook with 1.40-GHz Core i5 CPU and 16-GB RAM). ESMT becomes intractable when the maximal length $L_{\max}$ of the demonstrations reaches 10, while TLIP maintains relatively short computation time with increasing $L_{\max}$.

As ESMT is not scalable, we implement the following 4 methods for comparison for myopic teaching performances.

- **AN-TLIP**: TLIP for minimizing AN teaching cost.
- **AL-TLIP**: TLIP for minimizing AL teaching cost.
- **AN-RG**: randomized greedy algorithm for minimizing AN teaching cost. At each iteration, we pick the demonstration with minimal AN teaching cost among a randomly selected subset of demonstrations (of size 10000).
- **AL-RG**: same with AN-RG except that here we minimize the AL teaching cost.

Fig. 2 shows that AN-TLIP (resp. AL-TLIP) outperforms the other three methods when minimizing the AN teaching costs (resp. the AL teaching costs). Specifically, the AN teaching costs using AN-TLIP are up to 33.33%, 75.9% and 80.39% less than those using AL-TLIP, AN-RG and AL-RG, respectively. The AL teaching costs using AL-TLIP are 52.63%, 89.7% and 87.87% less than those using AN-TLIP, AN-RG and AL-RG, respectively. Fig. 2 also shows that the growth of the AL teaching cost is more significant with the increasing size of the hypothesis set than that of the AN teaching costs.

Teaching with Positive Demonstrations Only We test the machine teaching algorithm with positive demonstrations only. We consider global preferences, where the learner prefers F-formulas to G-formulas, and with the same temporal operator the learner prefers formula ϕ_1 to ϕ_2 if and only

Category	Hypothesis pLTL$_f$ Formulas
F-formulas	$F_{\leq 1}(x \leq 1), \ldots, F_{\leq 1}(x \leq 9)$, $\ldots$ $F_{\leq a}(x \leq 1), \ldots, F_{\leq a}(x \leq 9)$
G-formulas	$G_{\leq 1}(x \leq 1), \ldots, G_{\leq 1}(x \leq 9)$, $\ldots$ $G_{\leq a}(x \leq 1), \ldots, G_{\leq a}(x \leq 9)$

Table 1: Hypothesis set of pLTL$_f$ formulas ($a = 5, 10, 15$ correspond to hypothesis sets of sizes 90, 180 and 270).

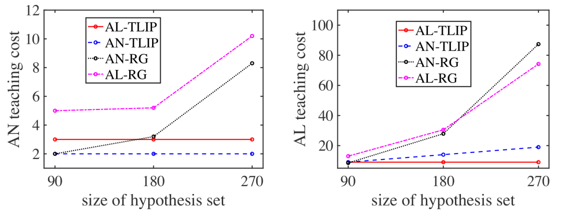

Figure 2: AN and AL teaching costs under global (uniform) preference with increasing sizes of the hypothesis set.

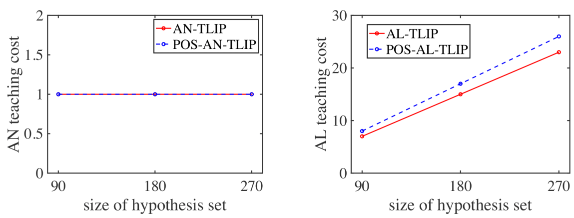

Figure 3: AN and AL teaching costs with positive demonstrations only and with both positive and negative demonstrations with increasing sizes of the hypothesis set.

if ϕ_1 implies ϕ_2. It can be shown that this preference function satisfies the condition of Theorem 2.

The corresponding algorithms for AN-TLIP and AL-TLIP with positive demonstrations only are referred to as POS-AN-TLIP and POS-AL-TLIP, respectively. Fig. 3 shows that POS-AN-TLIP and POS-AL-TLIP do not incur much additional teaching cost (up to 20% more) when restricted to only positive demonstrations.

Teaching pLTL$_f$ Formulas under Local Preferences

Local Preferences For two pLTL$_f$ formulas $\phi_1 = F_{\leq i_1}(x \leq v_1)$ and $\phi_2 = F_{\leq i_2}(x \leq v_2)$, where $i_1, i_2 \in \{1, \dots, 9\}$, $v_1, v_2 \in \{1, \dots, a\}$, we define the Manhattan distance between ϕ_1 and ϕ_2 as $|i_1 - i_2| + |v_1 - v_2|$. We consider the following local preference:
(1) The learner prefers formulas with the same temporal operator as that in the learner's previous hypothesis;
(2) With the same temporal operator the learner prefers formulas that are "closer" to the learner's previous hypothesis in terms of the Manhattan distance;
(3) The learner prefers G-formulas to F-formulas if the learner's current hypothesis are in the form of $F_{\leq i}(x \leq 1)$ or $F_{\leq i}(x \leq 9)$ $(i = 1, \dots, a)$. This is intuitively consistent with human's preferences to switch categories when the values reach certain boundary values.

	$L_{\max} = 5$	$L_{\max} = 10$	$L_{\max} = 15$
TLIP	3.67 s	5.29 s	7.65 s
ESMT	4.57 s	TO	TO

Table 2: Computation time for the myopic solutions.

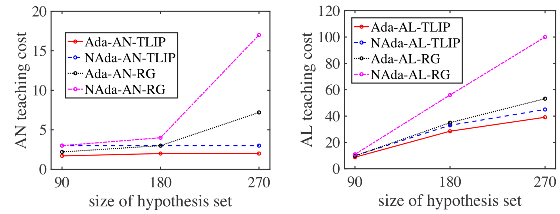

Figure 4: AN and AL teaching costs for adaptive and non-adaptive teaching with increasing sizes of the hypothesis set.

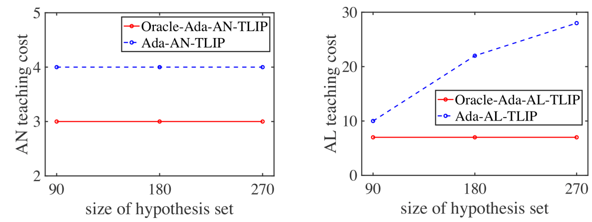

Figure 5: AN and AL teaching costs for adaptive teaching with oracles and without oracles with increasing sizes of the hypothesis set.

Adaptive Teaching To test the advantage of adaptive teaching, we compare it with non-adaptive teaching in the presence of uncertainties. We consider local preferences with added uncertainty noises. Specifically, the learner has equal preference of selecting the formulas in the version space that have the least Manhattan distance from the current hypothesis and also any formula that can be perturbed from these formulas in the version space (here "perturb" means adding or subtracting the parameters i or v by 1, e.g., as in $F_{\leq i}(x \leq v)$).

Fig. 4 shows that Ada-AN-TLIP (i.e., adaptive AN-TLIP) can reduce the AN teaching costs by up to 43.33% compared with NAda-AN-TLIP (i.e., non-adaptive AN-TLIP), Ada-AN-RG (i.e., adaptive AN-RG) can reduce the AN teaching costs by up to 57.65% compared with NAda-AN-RG (i.e., non-adaptive AN-RG), Ada-AL-TLIP (i.e., adaptive AL-TLIP) can reduce the AN teaching costs by up to 13.64% compared with NAda-AL-TLIP (i.e., non-adaptive AL-TLIP), and Ada-AL-RG (i.e., adaptive AL-RG) can reduce the AN teaching costs by up to 46.8% compared with NAda-AL-RG (i.e., non-adaptive AL-RG).

Adaptive Teaching with Oracles To decompose the teaching problem into subproblems that satisfy Condition 1 of Theorem 3, we design the oracle which outputs an intermediate target hypothesis $F_{\leq 1}(x \leq 10)$[2]. Fig. 5 shows that Oracle-Ada-AN-TLIP (adaptive AN-TLIP with oracles) can reduce the AN teaching costs by 25% compared with Ada-AN-TLIP and Oracle-Ada-AL-TLIP (adaptive AL-TLIP with oracles) can reduce the AL teaching costs by up to 75% compared with Ada-AL-TLIP.

[2]Theoretically, the oracle can be designed to output any formula of the form $F_{\leq i}(x \leq 0)$ or $F_{\leq i}(x \leq 10)$ $(i = 1, \dots, a)$.

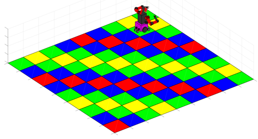

Figure 6: Simulated space in the robotic navigation scenario.

6 Case Study: Robot Navigation

In this case study, we consider a robotic navigation scenario in a simulated space partitioned into 81 cells as shown in Fig. 6. Each cell is associated with a color. A robot can be only at one cell at any time. The robot has five possible actions at each time step: stay still, go north, go south, go east or go west. For simplicity, we assume that each action is deterministic, i.e., there is zero slip rate. For example, when the robot takes action to go north at the cell located at (1, 1), it will land in the cell located at (2, 1). However, if the robot hit the boundaries, it will remain at the same position.

As shown in the gridworld map, if the robot is at a red cell, then at the next time step, it can be at a red cell or a blue cell, but cannot be at a green cell or a yellow cell; and if the robot is at a green cell, then at the next time step, it can be at a green cell, a yellow cell or a blue cell, but cannot be at a red cell. We add such transition constraints to the integer programming formulation in computing the demonstrations. The hypothesis set of pLTL$_f$ formulas are listed in Table 3. The set S of states is $\{$Red, Blue, Green, Yellow$\}$.

Global Preferences We implement four methods for comparison for myopic teaching performances for learners with global (uniform) preferences: AN-TLIP, AL-TLIP, AN-RG, and AL-RG.

Fig. 7 shows that AN-TLIP and AL-TLIP are the best for minimizing the AN teaching costs and the AL teaching costs, respectively. The AN teaching costs using AN-TLIP are the same as the AN teaching costs using AL-TLIP, which are up to 57.75% less than those using AN-RG and AL-RG. The AL teaching costs using AL-TLIP are 17.78%, 42.99% and 37.07% less than those using AN-TLIP, AN-RG and AL-RG, respectively.

Local Preferences We use the mapping ϱ that maps each color to an integer. Specifically, $\varrho(\text{Red}) = 1$, $\varrho(\text{Blue}) = 2$, $\varrho(\text{Green}) = 3$, and $\varrho(\text{Yellow}) = 4$. For two pLTL$_f$ formulas $\phi_1 = F_{\leq i_1}\text{Red}$ and $\phi_2 = F_{\leq i_2}\text{Green}$, we define the Manhattan distance between ϕ_1 and ϕ_2 as $|i_1 - i_2| + |\varrho(\text{Red}) - \varrho(\text{Green})|$. We consider the following local preference:

Example	Hypothesis pLTL$_f$ Formulas
F-formulas	$F_{\leq 1}$Red, $F_{\leq 1}$Blue, $F_{\leq 1}$Green, $F_{\leq 1}$Yellow, … $F_{\leq a}$Red, $F_{\leq a}$Blue, $F_{\leq a}$Green, $F_{\leq a}$Yellow
G-formulas	$G_{\leq 1}$Red, $G_{\leq 1}$Blue, $G_{\leq 1}$Green, $G_{\leq 1}$Yellow, … $G_{\leq a}$Red, $G_{\leq a}$Blue, $G_{\leq a}$Green, $G_{\leq a}$Yellow

Table 3: Hypothesis set of pLTL$_f$ formulas ($a = 5, 10, 15$ correspond to hypothesis sets of sizes 90, 180 and 270).

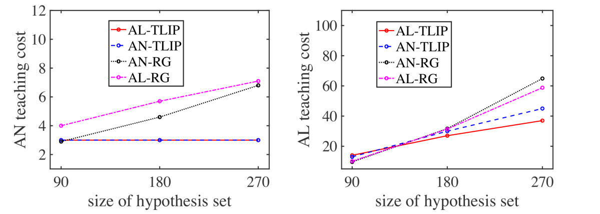

Figure 7: AN and AL teaching costs under global (uniform) preferences with increasing sizes of the hypothesis set in the robotic navigation scenario.

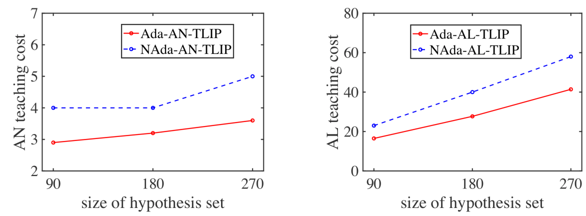

Figure 8: AN and AL teaching costs for adaptive and non-adaptive teaching with increasing sizes of the hypothesis set in the robotic navigation scenario.

(1) The learner prefers formulas with the same temporal operator as that in the learner's previous hypothesis;
(2) With the same temporal operator the learner prefers formulas that are "closer" to the learner's previous hypothesis in terms of the Manhattan distance.

To test the advantage of adaptive teaching, we compare it with non-adaptive teaching in the presence of uncertainties. We consider local preferences with uncertainty noises added in the same way as in the previous case study.

Fig. 8 shows that Ada-AN-TLIP (i.e., adaptive AN-TLIP) can reduce the AN teaching costs by 28% compared with NAda-AN-TLIP (i.e., non-adaptive AN-TLIP), and Ada-AL-TLIP (i.e., adaptive AL-TLIP) can reduce the AN teaching costs by 30.75% compared with NAda-AL-TLIP (i.e., non-adaptive AL-TLIP).

7 Conclusion

We presented the first attempt for teaching parametric linear temporal logic formulas to a learner with preferences. We also explored how to more efficiently teach the learner utilizing adaptivity and oracles. The results show the effectiveness of the proposed approach. We believe this is an important step towards practical algorithms for teaching more complex concept classes in the real-world scenarios. For future work, we will explore teaching methods for more general forms of temporal logic formulas, with more specific learning algorithms for inferring temporal logic formulas.

Acknowledgments

This work was supported by NSF CNS-1836900, ARL ACC-APG-RTP W911NF, NSF 1652113 and NSF 2040989.

References

Alur, R.; Etessami, K.; La Torre, S.; and Peled, D. 2001. Parametric Temporal Logic for "Model Measuring". *ACM Trans. Comput. Logic* 2(3): 388–407. ISSN 1529-3785. doi: 10.1145/377978.377990.

Bombara, G.; Vasile, C.-I.; Penedo, F.; Yasuoka, H.; and Belta, C. 2016. A Decision Tree Approach to Data Classification Using Signal Temporal Logic. In *Proc. HSCC'16*, 1–10. ISBN 978-1-4503-3955-1. doi:10.1145/2883817.2883843.

Brown, D. S.; and Niekum, S. 2019. Machine teaching for inverse reinforcement learning: Algorithms and applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7749–7758.

Chakraborty, S.; and Katoen, J.-P. 2014. Parametric LTL on Markov Chains. In *Theoretical Computer Science*, 207–221. Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, Y.; Aodha, O. M.; Su, S.; Perona, P.; and Yue, Y. 2018a. Near-Optimal Machine Teaching via Explanatory Teaching Sets. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Chen, Y.; Singla, A.; Aodha, O. M.; Perona, P.; and Yue, Y. 2018b. Understanding the Role of Adaptivity in Machine Teaching: The Case of Version Space Learners. In *NeurIPS*.

Dasgupta, S.; Hsu, D.; Poulis, S.; and Zhu, X. 2019. Teaching a black-box learner. In *International Conference on Machine Learning*, 1547–1555.

Eisner, C.; Fisman, D.; Havlicek, J.; Lustig, Y.; McIsaac, A.; and Van Campenhout, D. 2003. *Reasoning with Temporal Logic on Truncated Paths*, 27–39. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gao, Z.; Ries, C.; Simon, H. U.; and Zilles, S. 2017. Preference-based teaching. *The Journal of Machine Learning Research* 18(1): 1012–1043.

Goldman, S. A.; and Kearns, M. J. 1995. On the complexity of teaching. *Journal of Computer and System Sciences* 50(1): 20–31.

Gurobi. 2019. Gurobi Optimizer Reference Manual. URL http://www.gurobi.com. Last accessed on February 1, 2021.

Ho, H.-M.; Ouaknine, J.; and Worrell, J. 2014. *Online Monitoring of Metric Temporal Logic*, 178–192. Cham: Springer International Publishing. ISBN 978-3-319-11164-3.

Hoxha, B.; Dokhanchi, A.; and Fainekos, G. 2017. Mining parametric temporal logic properties in model-based design for cyber-physical systems. *International Journal on Software Tools for Technology Transfer* 79–93.

Kong, Z.; Jones, A.; and Belta, C. 2017. Temporal Logics for Learning and Detection of Anomalous Behavior. *IEEE TAC* 62(3): 1210–1222. ISSN 0018-9286. doi: 10.1109/TAC.2016.2585083.

Kress-Gazit, H.; Wongpiromsarn, T.; and Topcu, U. 2011. Correct, Reactive, High-Level Robot Control. *IEEE Robotics Automation Magazine* 18(3): 65–74. ISSN 1070-9932. doi:10.1109/MRA.2011.942116.

Liu, W.; Dai, B.; Humayun, A.; Tay, C.; Yu, C.; Smith, L. B.; Rehg, J. M.; and Song, L. 2017. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2149–2158. JMLR. org.

Ma, Y.; Zhang, X.; Sun, W.; and Zhu, J. 2019. Policy poisoning in batch reinforcement learning and control. In *NeurIPS*, 14543–14553.

Mac Aodha, O.; Su, S.; Chen, Y.; Perona, P.; and Yue, Y. 2018. Teaching categories to human learners with visual explanations. In *CVPR*, 3820–3828.

Mansouri, F.; Chen, Y.; Vartanian, A.; Zhu, J.; and Singla, A. 2019. Preference-Based Batch and Sequential Teaching: Towards a Unified View of Models. In *Advances in Neural Information Processing Systems*, 9195–9205.

Neider, D.; and Gavran, I. 2018. Learning Linear Temporal Properties. In *Formal Methods in Computer Aided Design (FMCAD)*, 1–10.

Pnueli, A. 1977. The Temporal Logic of Programs. In *Proc. 18th Annu. Symp. Found. Computer Sci.*, 46–57. Washington, D.C., USA. doi:10.1109/SFCS.1977.32.

Shah, A.; Kamath, P.; Shah, J. A.; and Li, S. 2018. Bayesian Inference of Temporal Task Specifications from Demonstrations. In *NeurIPS*, 3808–3817. Curran Associates, Inc.

To, S. T.; Roberts, M.; Apker, T.; Johnson, B.; and Aha, D. W. 2015. Mixed propositional metric temporal logic: A new formalism for temporal planning. In *AAAI Workshop on Planning for Hybrid Systems*. Phoenix, AZ: AAAI Press.

Vazquez-Chanlatte, M.; Jha, S.; Tiwari, A.; Ho, M. K.; and Seshia, S. A. 2018. Learning Task Specifications from Demonstrations. In *NeurIPS*, 5372–5382.

Xu, Z.; and Julius, A. 2018. Census Signal Temporal Logic Inference for Multiagent Group Behavior Analysis. *IEEE Trans. Autom. Sci. Eng.* 15(1): 264–277. ISSN 1545-5955. doi:10.1109/TASE.2016.2611536.

Xu, Z.; Nettekoven, A. J.; Agung Julius, A.; and Topcu, U. 2019. Graph Temporal Logic Inference for Classification and Identification. In *IEEE CDC'19*, 4761–4768.

Xu, Z.; Ornik, M.; Julius, A.; and Topcu, U. 2019a. Information-Guided Temporal Logic Inference with Prior Knowledge. In *ACC*. URL https://arxiv.org/abs/1811.08846.

Xu, Z.; Saha, S.; Hu, B.; Mishra, S.; and Julius, A. 2019b. Advisory Temporal Logic Inference and Controller Design for Semi-autonomous Robots. *IEEE T-ASE* 16(1): 459–477.

Yan, R.; Xu, Z.; and Julius, A. 2019. Swarm Signal Temporal Logic Inference for Swarm Behavior Analysis. *IEEE Robotics and Automation Letters* 4(3): 3021–3028.

Zhu, J. 2013. Machine teaching for Bayesian learners in the exponential family. In *NeurIPS*, 1905–1913.

Zhu, X. 2015. Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education. AAAI, 4083–4087. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2888116.2888288.