A Preference Learning Approach to Develop Safe and Personalizable Autonomous Vehicles

Ruya Karagulle[1], Nikos Aréchiga[2], Andrew Best[2], Jonathan DeCastro[2], and Necmiye Ozay[1]

Abstract— **This work introduces a preference learning method that ensures adherence to traffic rules for autonomous vehicles. Our approach incorporates priority ordering of signal temporal logic (STL) formulas, describing traffic rules, into a learning framework. By leveraging the parametric weighted signal temporal logic (PWSTL), we formulate the problem of safety-guaranteed preference learning based on pairwise comparisons, and propose an approach to solve this learning problem. Our approach finds a feasible valuation for the weights of the given PWSTL formula such that, with these weights, preferred signals have weighted quantitative satisfaction measures greater than their non-preferred counterparts. The feasible valuation of weights given by our approach leads to a weighted STL formula which can be used in correct-and-custom-by-construction controller synthesis. We demonstrate the performance of our method with human subject studies in two different simulated driving scenarios involving a stop sign and a pedestrian crossing. Our approach yields competitive results compared to existing preference learning methods in terms of capturing preferences, and notably outperforms them when safety is considered.**

I. Introduction

Preferences are a fundamental aspect of human behavior and decision-making, and it is valuable to design autonomous systems that allow for personalization to better suit the needs and desires of users. Surveys have demonstrated that drivers have different comfort and performance preferences while driving in different scenarios and conditions [2]–[4]. Moreover, drivers tend to prefer different driving styles for autonomous vehicles than their own styles [5]. Customizing autonomous vehicles can increase user satisfaction in these vehicles, and customization according to preferences over different styles rather than based on driver's data can help more. However, autonomous systems often require satisfaction of a rule set for safe operation. Relying on human preferences only may result in unsafe behaviors. For instance, at an intersection with a stop sign, drivers may sometimes prefer a rolling stop, which is illegal, over a full stop. However, an autonomous vehicle should always stop completely at a stop sign to guarantee safety of all agents in the environment. Preference learning algorithms for safety critical operations must consider rule satisfaction. The main motivation of our work is the need for safe, trustworthy and customizable autonomous vehicle algorithms.

For safety-critical applications like driving, there are three desirable properties a preference learning method should satisfy to allow safe personalization: (i) *expressivity*: the model should be expressive enough to capture preferences, (ii) *safety*: it ensures safety by preferring a rule-following behavior against a rule-violating one (even in cases the latter is scarce in the training data), and (iii) *usability in control design*: the learned model should be easy to integrate into downstream correct-by-construction control synthesis tasks. In this work, we propose to incorporate personalization and safety in one framework by using Signal Temporal Logic (STL) in a way to satisfy all of these properties. STL is a variant of temporal logic that is tailored for reasoning about the temporal properties of time series data and commonly used in describing correct behaviors in autonomous systems. Its descriptive power on systems opens wide application areas, from controller synthesis, motion planning, to classification problems [6]–[11].

To develop a personalization framework with the STL formalism, we use a parametric extension to Weighted Signal Temporal Logic (WSTL), which is tailored for the ordering of preferences and priorities in STL formulas [12]. We introduce a learning framework that is based on this extension. The learning framework returns required parameters for WSTL formula, which can be used to synthesize a controller that yields preferred system behaviors as in [12], [13]. Starting with a parametric WSTL formula that specifies task objectives (traffic rules in autonomous vehicles) and a set of pairwise comparison preferences among a set of safe behaviors, the goal is to find suitable formula parameters such that preferred signals have greater satisfaction measure, namely *WSTL robustness*, than their non-preferred counterparts. We show how to cast this problem as an optimization problem. We propose two different approaches to solve the resulting optimization problem: a random sampling approach and gradient-based approach with a construction of a computation graph to calculate the WSTL robustness of signals.

To evaluate the performance of our framework, we simulate two different driving scenarios one with an autonomous vehicle navigating an intersection with a stop sign and one with an autonomous vehicle approaching to a pedestrian crossing while there is a pedestrian crossing the road. We generate two sets of trajectories that comply with traffic rules for these scenarios, and run human subject studies with four participants for both scenarios. We discuss the performance of our solution approach and compare them with baseline preference learning methods. Our results verify the need for safety-aware preference learning by showing that baseline methods usually lead to unsafe selections.

Toyota Research Institute provided funds to support this work. A poster abstract on our preliminary results was presented at HSCC [1].

[1]Electrical and Computer Engineering, University of Michigan, Ann Arbor, USA ruyakrgl@umich.edu

[2] Toyota Research Institute, Los Altos, USA

II. Literature Review

Preference learning aims to understand and predict individuals' preferences based on a set of their choices [14], [15]. This can be done through independent evaluations, such as rating, or comparisons with alternatives. While both evaluations open new research methods, learning from comparison pairs may help in terms of dividing the problem into smaller, more manageable batches [14]. While these methods capture and reason about preferences, for safety-critical scenarios such as capturing driving preferences and personalizing driving styles, they cannot ensure necessary safety guarantees.

Another use of preferences is preference-based learning for reward functions and task learning in robot systems [16]–[18]. For safety-aware applications, [19] combines preference-based learning with control barrier functions.

On the other hand, encoding safety rules in temporal logic is an eminent method for safety-critical applications [20], [21]. Specifications in temporal logic can be used for controller synthesis [6], [7], [21], motion planning [8], [22], [23] and also learning applications [9]–[11], [24]–[28] in many autonomous systems. In particular, works in [9], [10], [24], [25], [29] try to infer a temporal logic formula from the data used for classification. As a subset of learning applications, in robot learning, Chou et al. [26] tries to learn task specifications in linear temporal logic from demonstrations, Puranic et al. [11] scores demonstrations with the help of ordered specifications in the form of signal temporal logic, and works in [27], [28] use temporal logic for reward shaping and reinforcement learning.

Incorporating preferences and priorities with temporal logic is studied in [1], [12], [30]–[32]. The work in [12] introduces a weighted variant of the STL, called Weighted Signal Temporal Logic (WSTL), in which weights reflect the order of priority of preference. The work in [30] defines Weighted Truncated Linear Temporal Logic. Both works assumes that they have the knowledge for the formula and associated weights. For the end-user, it is hard to interpret the weights and define their preferences in temporal logic formalism, so there needs to be an intermediate step to infer the weights from the user. In [31], [32], a parametric extension of WSTL, which we call PWSTL, is used in a time series classification problem, where weights of the formula are learned using neural networks.

III. Preliminaries

A. Signal Temporal Logic (STL)

STL is a temporal logic formalism used to reason about signals $s : \mathbb{T} \to \mathcal{S}$, where $\mathbb{T}$ is a time domain and $\mathcal{S} \subseteq \mathbb{R}^n$ is a n dimensional real-valued signal domain [33]. We will consider $\mathbb{T}$ to be infinite $\mathbb{Z}_{\geq 0}$ or finite $[0, t_{final}] \subset \mathbb{Z}_{\geq 0}$. An STL formula ϕ is given by the grammar $\phi ::= \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_{[a,b]} \phi_2$. Boolean true is $\top$, and π is a predicate of the form $\pi(s(t)) := f_\pi(s(t)) \geq 0$ where $f_\pi : \mathcal{S} \to \mathbb{R}$ and $s(t)$ is the signal value at time instant t. The logical not is $\neg$, the conjunction is $\wedge$, and $\mathcal{U}_{[a,b]}$ is the "Until" operator. Additional operators, disjunction $\vee$, Always $\Box_{[a,b]}$, and Eventually $\Diamond_{[a,b]}$ can be derived from operators in the grammar[1]. Subscript $[a, b]$ defines the time interval. When the time interval is from 0 to ∞, the subscript is omitted. We will denote the set of all well-formed STL formulas with $\mathcal{F}$. If a signal s satisfies a formula ϕ at time t, it is shown as $(s, t) \models \phi$. If it violates at t, it is shown as $(s, t) \not\models \phi$. The qualitative semantics of STL is defined as follows:

$$\begin{array}{ll}
(s,t) \models \pi & \Leftrightarrow \pi(s(t)), \\
(s,t) \models \neg\phi & \Leftrightarrow (s,t) \not\models \phi, \\
(s,t) \models \phi_1 \wedge \phi_2 & \Leftrightarrow ((s,t) \models \phi_1 \text{ and } (s,t) \models \phi_2), \\
(s,t) \models \phi_1 \mathcal{U}_{[a,b]} \phi_2 & \Leftrightarrow \exists t' \in [t+a, t+b]((s,t') \models \phi_2 \\
& \quad \text{and } \forall t'' \in [t, t')\ (s,t'') \models \phi_1).
\end{array}$$

Derived operators have following qualitative semantics:

$$\begin{array}{lll}
(s,t) \models \phi_1 \vee \phi_2 & \Leftrightarrow & ((s,t) \models \phi_1 \text{ or } (s,t) \models \phi_2), \\
(s,t) \models \Box_{[a,b]} \phi & \Leftrightarrow & \forall t' \in [t+a, t+b]\ (s,t') \models \phi, \\
(s,t) \models \Diamond_{[a,b]} \phi & \Leftrightarrow & \exists t' \in [t+a, t+b]\ (s,t') \models \phi.
\end{array}$$

For qualitative semantics at time instant $t = 0$, we omit t and write $s \models \phi$. STL has quantitative semantics as well. It measures how well the formula models the signal. There are different quantitative semantics, also known as robustness metrics [34], [35]. In this paper we use the traditional robustness metric, as defined in [34]. Robustness metric $\rho : \mathcal{S} \times \mathcal{F} \times \mathbb{T} \to \mathbb{R}_e$ is defined recursively as:

$$\begin{aligned}
\rho(s, \top, t) &= \infty, \\
\rho(s, \pi, t) &= f_\pi(s(t)), \\
\rho(s, \neg\phi, t) &= -\rho(s, \phi, t), \\
\rho(s, \phi_1 \wedge \phi_2, t) &= \min\big(\rho(s, \phi_1, t), \rho(s, \phi_2, t)\big), \\
\rho(s, \phi_1 \mathcal{U}_{[a,b]} \phi_2, t) &= \max_{t' \in [t+a, t+b]} \big(\min\big(\rho(s, \phi_2, t'), \\
&\qquad \min_{t'' \in [t, t']} \rho(s, \phi_1, t'')\big)\big).
\end{aligned}$$

Robustness for derived operators are

$$\begin{aligned}
\rho(s, \phi_1 \vee \phi_2, t) &= \max\big(\rho(s, \phi_1, t), \rho(s, \phi_2, t)\big), \\
\rho(s, \Diamond_{[a,b]} \phi, t) &= \max_{t' \in [t+a, t+b]} \rho(s, \phi, t'), \\
\rho(s, \Box_{[a,b]} \phi, t) &= \min_{t' \in [t+a, t+b]} \rho(s, \phi, t').
\end{aligned}$$

Robustness at $t = 0$ is shown as $\rho(s, \phi)$. Note that for finite signals where $t_{final} < \infty$, time interval $[t+a, t+b]$ in temporal operators may exceed the time length of the signal. In this case, time interval can be taken as $[t+a, \min(t+b, t_{final})]$ assuming that $t + a \leq t_{final}$. For simplicity, we keep the semantics for infinite signals but we use STL for finite signals with necessary corrections [36]. Note that robustness metric of STL is sound, i.e., $\rho(s, \phi, t) > 0 \iff s(t) \models \phi$ and $\rho(s, \phi, t) < 0 \iff s(t) \not\models \phi$.

Example 1: Let $s = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T \in \mathbb{R}^{2 \times t_{final}}$ be a two-dimensional signal with length t_{final}. Let $\phi = \Diamond(s_1 \leq 0 \wedge s_2 \geq 0)$ be an STL formula. Satisfaction of ϕ by the signal s means that "There is a time $t^* \leq t_{final}$ such that $s_1(t^*) \leq 0$ and $s_2(t^*) \geq 0$". The robustness of s over ϕ is

$$\rho(s, \phi, t) = \max_{t' \in [t, t_{final}]} (\min(-s_1(t'), s_2(t'))).$$

[1] Disjunction is $\phi_1 \vee \phi_2 = \neg(\neg\phi_1 \wedge \neg\phi_2)$, Eventually is $\Diamond_{[a,b]}\phi = \top \mathcal{U}_{[a,b]} \phi$, and Always is $\Box_{[a,b]}\phi = \neg(\Diamond_{[a,b]} \neg\phi)$.

B. Weighted Signal Temporal Logic (WSTL)

WSTL is tailored to represent priorities and preferences in STL formulas [12]. Its syntax extends STL syntax as

$$\phi := \top \mid \pi \mid \neg\phi \mid \phi_1 \wedge^w \phi_2 \mid \phi_1 \mathcal{U}^{w^1,w^2}_{[a,b]} \phi_2,$$

where the weights are $w \in \mathbb{R}^2_+$ and $w^1, w^2 \in \mathbb{R}^{(b-a+1)}_+$. All operators are interpreted as in STL.

In [12], quantitative semantics of WSTL is called *WSTL robustness*, denoted as $r : \mathcal{S} \times \mathcal{F} \times \mathbb{T} \to \mathbb{R}$. We adopt WSTL formalism with the following quantitative semantics:

$$\begin{aligned}
r(s,\top,t) &= \infty\\
r(s,\pi,t) &= \rho(s,\pi,t)\\
r(s,\neg\phi,t) &= -r(s,\phi,t),\\
r(s,\phi_1 \wedge^w \phi_2,t) &= \min\big(w_1 r(s,\phi_1,t), w_2 r(s,\phi_2,t)\big),\\
r(s,\phi_1 \mathcal{U}^{w^1,w^2}_{[a,b]} \phi_2,t) &= \max_{t'\in[t+a,t+b]}\Big(\min\big(w^1_{t'-t-a+1} r(s,\phi_2,t'),\\
&\qquad w^2_{t'-t-a+1} \min_{t''\in[t,t')} r(s,\phi_1,t'')\big)\Big).
\end{aligned} \tag{1}$$

Derived operators have weighted robustness definitions as:

$$\begin{aligned}
r(s,\phi_1 \vee^w \phi_2,t) &= \max\big(w_1 r(s,\phi_1,t), w_2 r(s,\phi_2,t)\big),\\
r(s,\Box^w_{[a,b]}\phi,t) &= \min_{t'\in[t+a,t+b]}(w_{t'-t-a+1} r(s,\phi,t')),\\
r(s,\Diamond^w_{[a,b]}\phi,t) &= \max_{t'\in[t+a,t+b]}(w_{t'-t-a+1} r(s,\phi,t')).
\end{aligned}$$

Note that when deriving Eventually from Until, we have $\top$ in the second part of the robustness formula of $\mathcal{U}$. Since the weighted robustness of $\top$ is ∞, we can drop the set of weights w^2 when defining Eventually because it does not affect the result of `min` operation inside the computation of robustness of Until. That is the reason why Eventually (and hence Always) have fewer weights than Until. Moreover, Boolean truth, predicate and negation do not have associated weights, i.e., these operators have weights equal to 1.

Example 2: Let s and ϕ be the signal and WSTL formula constructed from STL formula in Example 1 with weights $\{w^{\Diamond}_i\}_{i=1}^{t_{final}-t+1}$ and $\{w^{\wedge}_i\}_{i=1}^{2}$. The WSTL robustness of s is

$$r(s,\phi,t) = \max_{t'\in[t,t_{final}]}(w^{\Diamond}_{t'-t+1} \min(-w^{\wedge}_1 s_1(t'), w^{\wedge}_2 s_2(t'))).$$

As in STL robustness, $r(s,\phi)$ is WSTL robustness at $t = 0$. The quantitative semantics is said to be *sign-consistent* if $\rho(s,\phi,t) r(s,\phi,t) > 0$, where $\rho(s,\phi,t) \neq 0$, for all operators in the syntax.

Lemma 1: (Theorem 2 in [12]) If a quantitative semantics is sign-consistent, robustness definition of WSTL is sound.

Theorem 1: Quantitative semantics in (1) is sound.

Proof: According to Lemma 1, it is sufficient to prove that quantitative semantics in (1) is sign-consistent. Since all weights are defined positive, multiplying a robustness value with a weight does not change its sign. Therefore, for each recursive operation in WSTL robustness calculation, the sign of the robustness value associated with this recursion step is preserved. Then, it is true that $\rho(s,\phi,t) r(s,\phi,t) > 0$ holds for every $(s,\phi,t) \in \mathcal{S} \times \mathcal{F} \times \mathbb{T}$ and all weights w. ■

In WSTL definition of [12], weights are pre-determined positive real values. In this work, we use an extension to WSTL that we call Parametric Weighted Signal Temporal Logic (PWSTL) in which weights are unknown parameters (cf., [31]). We denote the set of unknown parameters as $\mathcal{W}$ and denote PWSTL formulas as $\phi_{\mathcal{W}}$. A PWSTL formula results in a WSTL formula $\phi_{\mathcal{W}=w}$ with the valuation w of parameters. Note that in PWSTL, some weights may be known and $\mathcal{W}$ should be defined along with the formula.

IV. Problem Statement and Solution Method

As we focus on driving scenarios, inputs to our problem are signals. Preferences are given in pairs and preference data for signals is defined as follows.

Definition 1 (Preference Data): Preference data $\mathcal{P} := \{(s^+_i, s^-_i)\}_{i=1}^{P}$ is a set of P pairwise comparisons. In each pair (s^+_i, s^-_i), s^+_i represents the preferred signal and s^-_i represents non-preferred one.

The goal of this work is to select a weight valuation $\tilde{w}$, for the parameter set $\mathcal{W}$ of the PWSTL formula $\phi_{\mathcal{W}}$. Formula $\phi_{\mathcal{W}}$ is determined according to system rules, so that it captures the preferences. Formally, this paper aims to solve the following problem:

Problem 1: Given a PWSTL formula ϕ with a weight parameter set $\mathcal{W}$, and a preference data $\mathcal{P}$, find a valuation w of $\mathcal{W}$ such that

$$r(s^+_i, \phi_{\mathcal{W}=w}) > r(s^-_i, \phi_{\mathcal{W}=w}) \quad \forall (s^+_i, s^-_i) \in \mathcal{P}. \tag{2}$$

Problem 1 is a feasibility problem. We provide an analysis of the set of feasible weights using the syntax tree of formulas. STL formulas have an associated syntax tree, in which nodes represent Boolean and temporal operators, leaf nodes represent predicates, and edges represent the connection between operators and operands [37]. Let the *root weights* of a WSTL formula be the weights associated with the weighted operator closest to the root of its syntax tree. For instance, for $\varphi = \varphi_1 \mathcal{U}^{w^1,w^2}_{[a,b]} \varphi_2$, the root of the syntax tree of φ has the operator $\mathcal{U}^{w^1,w^2}_{[a,b]}$ and the root weights are $[w^1; w^2]$; but for $\varphi' = \neg(\varphi_1 \wedge^w \varphi_2)$, the root of the syntax tree of φ' has the operator $\neg$, which is not a weighted operator, so we look at its children until we find a weighted operator, which in this case turns about to be $\wedge^w$. Hence, root weights of φ' are w.

Next, we show that the feasible weight valuations of Problem 1 is unbounded, when non-empty, due to homogeneity with respect to the root weights of the formula.

Lemma 2: Let ϕ be an PWSTL formula with weight set $\mathcal{W}$ that contains only the weight parameters for the root weights of ϕ. If valuation w of $\mathcal{W}$ solves Problem 1, then $\tilde{w} = \alpha w$ also solves the problem for any $\alpha > 0$.

Proof: If the WSTL formula with valuation w is a feasible solution for Problem 1, we know that for all pairs in $\mathcal{P}$, $r(s^+_i, \phi_{\mathcal{W}=w}) > r(s^-_i, \phi_{\mathcal{W}=w})$ holds. We also have

$$r(s, \phi_{\mathcal{W}=\tilde{w}}) = \alpha r(s, \phi_{\mathcal{W}=w}).$$

This together with $\alpha > 0$ implies for all $(s^+_i, s^-_i) \in \mathcal{P}$, $r(s^+_i, \phi_{\mathcal{W}=\tilde{w}}) > r(s^-_i, \phi_{\mathcal{W}=\tilde{w}})$. Hence, the WSTL formula with valuation $\tilde{w}$ is a feasible solution for Problem 1. ■

Given the above property, namely *root-layer homogeneity*, we will show that it is possible to restrict the weight valuations to a bounded set $\mathcal{D}$ that is guaranteed to include at least one solution whenever a solution exists.

Theorem 2: Let $\mathcal{D} = \mathcal{B}_\infty(0) \cap \mathbb{R}_+$, i.e., the intersection of the n-dimensional closed unit ball in infinity-norm and the positive quadrant. If Problem 1 is feasible with weight valuation w, then there exists at least one weight valuation w' in the domain $\mathcal{D}$ such that $\phi_{\mathcal{W}=w'}$ solves the problem.

Proof: Let Problem 1 be feasible for the valuation w. If $w \in \mathcal{D}$, the proof is trivial. So, let us assume $w \notin \mathcal{D}$.

We will prove the theorem by induction on the depth d of the syntax tree of $\phi_{\mathcal{W}=w}$. For each subformula ϕ_s at level k ($k < d$) of the syntax tree of $\phi_{\mathcal{W}=w}$, assume that the root weights of ϕ_s are w_s and all the remaining weights of ϕ_s are already less than or equal to 1 (note that this trivially holds in the base case when $k = d$ where we pick $w^{(d)} = w$). Then, we will show that we can define a new set of weights $w^{(k)}$ for $\phi_{\mathcal{W}}$ such that $r(s, \phi_{\mathcal{W}=w^{(k)}}, t) = r(s, \phi_{\mathcal{W}=w^{(k+1)}}, t)$ such that the weights of each subformula at level $k-1$ except for their root weights are less than or equal to 1.

Consider an arbitrary subformula ϕ_s at level k with weights w_s satisfying the induction hypothesis. We use ϕ_{s,w_s} as a shorthand for such a pair to differentiate it from the same formula with updated weights, ϕ_{s,w'_s}. Define $w'_s = w_s / \max(w_s)$. Clearly, $r(s, \phi_{s,w_s}, t) = \max(w_s) r(s, \phi_{s,w'_s}, t)$ and all weights of ϕ_{s,w'_s} are less than or equal to 1. However, we can scale the weights w_u that multiply $r(s, \phi_{s,w_s}, t)$ at level $k-1$ with $\max(w_s)$ so that with valuation $w^{(k)}$, where the weights $\max(w_s) w_u$ and w'_s are replaced by w_u and w_s, we achieve the same weighted robustness value, establishing the induction hypothesis.

Finally, we can decrement k until we reach the root weights of $\phi_{\mathcal{W}=w}$ and invoke Lemma 2 to scale the root weights to be less than or equal to 1 while preserving feasibility. Therefore, the scaled valuation is in $\mathcal{D}$. ∎

Having a bounded feasible domain will be useful in our computational approach.

A. An Optimization Reformulation

Problem 1 can be formulated as an optimization problem.

Problem 2: Given preference data $\mathcal{P}$, PWSTL formula $\phi_{\mathcal{W}}$ and domain $\mathcal{D}$ described in Theorem 2, solve

$$w^* \in \arg\min_{w\in\mathcal{D}} \sum_{(s_i^+, s_i^-)\in\mathcal{P}} -\mathbb{1}(w)_{(r(s_i^+, \phi_{\mathcal{W}=w}) - r(s_i^-, \phi_{\mathcal{W}=w}) > 0)}, \quad (3)$$

where $\mathbb{1}(w)$ is the indicator function which takes $\mathbb{1}(w) = 1$ when the subscripted condition is satisfied and takes $\mathbb{1}(w) = 0$ otherwise.[2]

Next, we show that the solution of this optimization problem is a best possible solution in a specific sense.

Theorem 3: If Problem 1 is feasible, then a minimizer w^* of Problem 2 is a solution to Problem 1. Moreover, if Problem 1 is infeasible, Problem 2 finds a valuation for $\phi_{\mathcal{W}}$ that maximizes the number of pairs that satisfy Inequality (2).

Proof: For a preference pair (s_i^+, s_i^-), we have $\mathbb{1}(w)_{(r(s_i^+, \phi_{\mathcal{W}=w}) - r(s_i^-, \phi_{\mathcal{W}=w}) > 0)} \in \{0, 1\}$. Therefore, the objective takes values between $-|\mathcal{P}|$ and 0. By Theorem 2, the feasibility of Problem 1 implies the existence of a weight $w^* \in \mathcal{D}$ such that the objective is $-|\mathcal{P}|$. Since this is the minimum achievable value of the objective, w^* is a solution of the optimization Problem 2. Similarly, by definition of the indicator function, the objective $-|\mathcal{P}|$ implies that, for all preferences, Inequality (2) is satisfied.

Assume for any weight w at most $k < |\mathcal{P}|$ preference pairs satisfy the Inequality (2). This is the case when Problem 1 is infeasible. In this case, Problem 2 finds a valuation such that k pairs satisfy Inequality (2). The cost function is of the staircase form and obtaining $-k$ as a cost function value is only possible when k pairs satisfy the Inequality (2). Thus, the minimum cost value will always be obtained from the maximum amount of pairs that satisfy Inequality (2). ∎

Problem 2 not only transforms the feasibility Problem 1 into an optimization problem, but also returns a valuation that makes maximum number of pairs correctly ordered according to Inequality (2) when Problem 1 is infeasible.

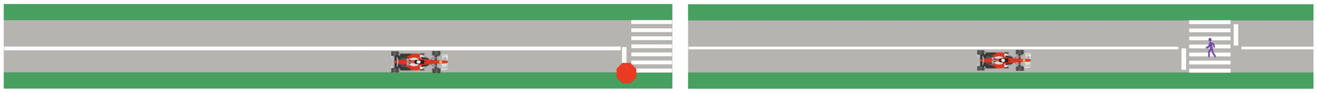
It is important to note that with Problem 2 and weights being positive, it is impossible to find weight valuations that result in a greater robustness value of a rule-violating behavior than the robustness value of a rule-satisfying one. Violating signals will always have negative robustness values. If there exists a pair in the preference dataset such that the person prefer a rule-violating behavior over a satisfying behavior, we cannot satisfy Inequality (2) for this pair, Problem 1 becomes infeasible and we will find a valuation that satisfies Inequality (2) for maximum number of pairs.

Remark 1: We note that equality predicates and Boolean signals may require special treatment. This is because equality predicates, i.e., $\pi(s(t)) = f_\pi(s(t)) \geq 0 \wedge -f_\pi(s(t)) \geq 0$, cannot hold strictly positive robustness values. When inserted into a conjunction, $\phi = \phi_1 \wedge \pi$, equality predicate dominates the robustness of ϕ and restricts it to non-positive values. As a result, we cannot observe the effect of ϕ_1 on robustness values, and weights in front of and under ϕ_1 does not appear in WSTL robustness of ϕ. In other words, ϕ_1 is shadowed. As observed earlier [35], shadowing is not ideal when we want to learn the importance order of different time instances and subformulas. A possible remedy to this issue, which is used in our experiments, is to introduce Boolean signal b as a substitute to equality predicates. That is, if $f_\pi(s(t)) = 0$, $b = \top$, otherwise $b = \bot$. When b is false, robustness of ϕ goes to $-\infty$, and when b is true, robustness of ϕ is determined by ϕ_1. Please note that this remedy only works over conjunction and Always operators.

B. Computational Approach

We note that Problem 2 is highly non-convex and non-differentiable. Even if we create a differentiable surrogate function and render the robustness definition differentiable with known methods [38], the loss function retains a highly

[2] Note that $\mathcal{D}$ is an open set, but the objective function takes only finitely many values, hence it always has a minimum. Therefore, searching for $\arg\min$ is valid.



(a) Stop Sign Scenario: Vehicle approaching to an intersection with a stop sign. The traffic rule says that vehicles should stop before the stop sign.

(b) Pedestrian Scenario: Vehicle approaching to a pedestrian crosswalk, while a pedestrian is crossing. Vehicle can come to a complete stop or slow down sufficiently to allow the pedestrian.

Fig. 1: Two scenarios that is used for experiments

non-convex nature. As a result, it is hard to solve this problem to a global minimum. In the following, we propose two approaches, one gradient-based, the other sampling based that aim to find an approximate solution.

a) Gradient-based optimization: Thanks to the prevalence and success of gradient-based methods and backpropagation in machine learning, many temporal logic learning algorithms using gradients have been proposed [38]. To be able to compute the gradient, we need a differentiable loss function. In the weighted robustness definition, we replace `max` and `min` functions with their soft differentiable versions `softmin/softmax` as in [38]. We also replace the indicator function with the logistic function with a shift. The shift helps for avoiding equality of robustness values in preference pairs. Overall, we propose the following surrogate loss

$$\mathcal{L} = \sum_{(s_i^+, s_i^-)\in\mathcal{P}} (1+\exp(M[r(s_i^+,\phi_{\mathcal{W}=w}) - r(s_i^-,\phi_{\mathcal{W}=w}) - \epsilon]))^{-1} + \log(1+\theta\exp(\|W_\phi\|_2^2 - \|W_\phi^{init}\|_2^2)),$$

where M is a large number, ϵ is a small shift, and θ is an optimization weight for the second term. Here, the first term is an approximation of the cost function in Equation (3) and the second term promotes the norm of the weights not to change too much compared to its initial value W_ϕ^{init}, where $W_\phi^{init} \in \mathcal{D}$. This second term is essentially a surrogate for the constraints in Equation (3); and due to Theorem 2 and the equivalence of the infinity norm and 2-norm in finite dimensions, does not change the validity of the solutions.
Implementation details: We can use readily available software tools like PyTorch which benefits from backpropagation for gradient computation, which requires graph-like structure for loss functions. Inspired from [38], we construct a computation graph for robustness of WSTL formulas from syntax trees. For each operator in the quantitative semantics, we construct a computation block. Since weighted robustness is computed recursively, we can connect computation blocks for each operator using the syntax tree of the formula and form the computation graph. Computation graph takes a signal as input and returns the weighted robustness value of that signal at time t as output. We use Adam [39] as the optimization method. Unfortunately, depending on the loss surface, gradient-based methods can become stuck in local minimum. The loss surface can have steep changes and flat surfaces depending on the formula and preference set, potentially leading to failure of gradient-based methods. One potential remedy might involve decreasing the softness coefficient β of `softmin/max`, but it would compromise the soundness guarantee. Another strategy could be decreasing the steepness of the logistic function, i.e., decrease M, but this will make the surrogate $\mathcal{L}$ less similar to the objective in Problem 2. Another strategy could be initializing the iteration from multiple random points to overcome bad local minima.

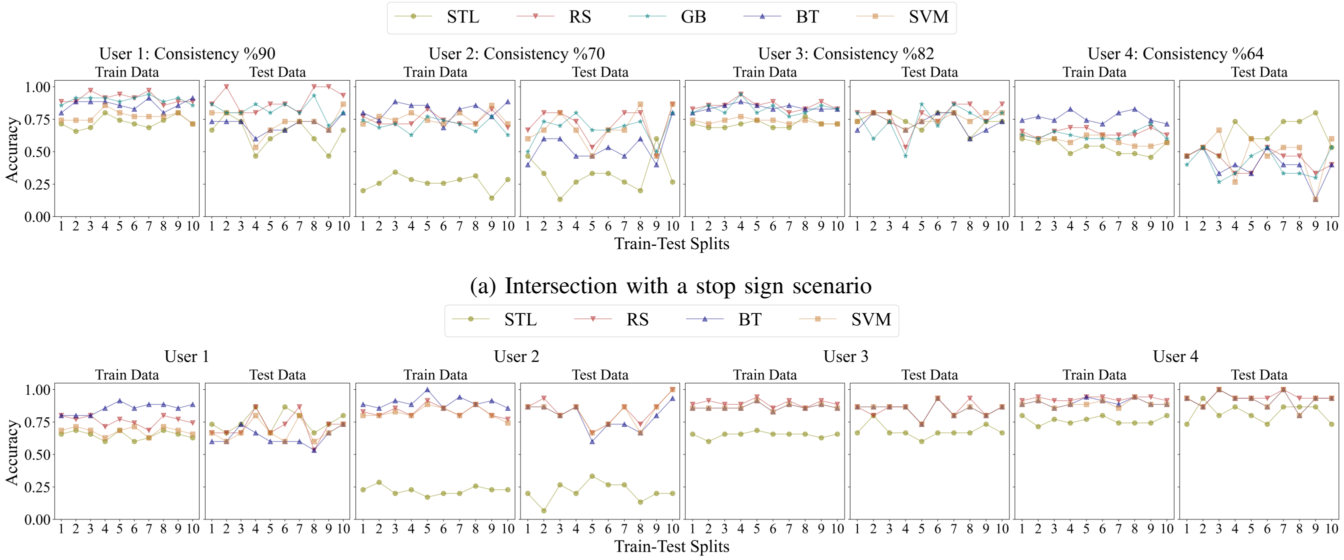
b) Random Sampling: Randomized methods have shown some success in temporal logic planning problems [40], especially when there is a multitude of feasible solutions. Similarly in [41], it is shown that simple random search can give not only competitive but also faster results compared to gradient methods. This inspires our attempt to solve Problem 2 through random sampling in the region $\mathcal{D} = \mathcal{B}(0)_\infty \cap \mathbb{R}_+$. We uniformly sample weight valuations in $\mathcal{D}$. *Implementation details:* We want the weight valuations to meet the following criterion: the absolute difference in robustness between signals within a pair should exceed 5% of the range between the maximum and minimum robustness values among all signals. While this condition is not required for the random sampling approach alone, it can be useful for two downstream tasks: (i) when using the best performing of these weights as initialization of gradient-based approaches[3], this separation helps start the iterations at a part of the weight space where logistic function well-approximates the indicator function; (ii) when using the learned formula in controller synthesis, weights that well-separates the preferences lead to controllers that more robustly reflect the preferences.

V. Experiments

In this section, we provide a comparison of solution approaches with baseline methods along with demonstrating the need for a safety-guaranteed preference learning framework. We also showcase framework's performance on capturing personal preferences of different participants of a human subject study. For these purposes, we use two different driving scenarios.

Driving Scenarios: We can use temporal logics to specify traffic rules in driving scenarios. The first scenario is a simple intersection with a stop sign, a screenshot is shown in Figure 1a. The vehicle must stop before the stop sign, but there is some flexibility in the approach and final position. The traffic rule can be expressed in STL as follows: $\phi^{stop} = \Diamond\Box(x - x_{stop} \geq 0 \wedge v = 0) \wedge \Box(v \geq 0)$ where x and v are the position and speed signals of a vehicle, respectively, and x_{stop} is the stop sign position. Note that $\rho(s, \phi^{stop}) \leq 0$ for any signal due to equality predicate. We substitute $v = 0$ with its Boolean form as discussed in Remark 1. We construct

[3] We tried this combination in our experiments, however, the performance improvement was not significant. Therefore, due to space constraints, we do not report these results further.



(a) Intersection with a stop sign scenario

(b) Approaching to a pedestrian scenario

Fig. 2: Human subject study results for the two scenarios for all four users. "STL" denotes the traditional (unweighted) robustness when it is used directly,"RS" denotes our method with random sampling, "GB" denotes our method with gradient-based optimization, "BT" denotes SGD with Bradley-Terry model, and "SVM" represents SVM classification.

PWSTL formula $\phi^{stop}_{\mathcal{W}}$ with a weight parameter set $\mathcal{W}$ that contains all weights in the formula. In the second scenario, we observe an ego vehicle approaching to a pedestrian while she is crossing the road, as illustrated in Figure 1b. The traffic regulation in this case is expressed in STL form as $\phi^{pedes} = \Box[(p \wedge \neg(x - x_{cross} \le 0)) \implies (x - x_{cross} \le 0 \; \mathcal{U} \; \neg p) \wedge (v \le v_{lim})]$ where x, v represent position and velocity signals, respectively. Signal p is a Boolean signal that indicates the presence of a pedestrian.

Human Subject Studies: Studies are completed under IRB study no HUM00221976. For each scenario, we collaborated with four participants. We simulate hundred trajectories that satisfy their temporal logic formula. We compose fifty pairs such that the Euclidean distance between each pair is greater than a threshold. This threshold value is determined manually as the point at which the difference between signals becomes difficult to discern. These pairs are shown to participants who then choose their preferred behavior. As human decisions can vary in consistency, for the first scenario, we repeat the same question set twice to have a measure of participant's decisiveness. The consistency of the answers of each participant is reported in Figure 2a.

A. Baseline Methods

One well-known approach to pairwise preference learning problem is to recast it as a supervised learning problem [42]. Let $\psi(s)$ be the feature vector of item s. We use Fourier transform of the signal s for $\psi(\cdot)$. To set up the supervised learning problem, for a given preference pair (s_i^+, s_i^-), we construct a new feature vector as the difference of feature vectors as $\psi(s_i^+) - \psi(s_i^-)$. All signals pairs in $\mathcal{P}$ belongs to Class 0. We generate the data for Class 1 by reversing the signal order and defining the feature vector $\psi(s_i^-) - \psi(s_i^+)$. This process gives us binary labels for all comparison pairs and their reverse orders. Then, we use Support Vector Machines (SVM) to learn a binary classifier. In the SVM, we use radial basis functions as kernels. For a test pair (s_1, s_2), if $\psi(s_1) - \psi(s_2)$ is classified in Class 0, we say s_1 is preferred over s_2; and we say s_2 is preferred over s_1 otherwise.

The second baseline method is based on a representation of pairwise user preferences with the likelihood of selecting one item over another. In particular, Bradley-Terry model is a common example of such likelihood function model used in preference learning applications [43]. Bradley-Terry model [43] uses the following likelihood function:

$$P_v(s_i^+, s_i^-) = \frac{e^{<v, \psi(s_i^+)>}}{e^{<v, \psi(s_i^-)>} + e^{<v, \psi(s_i^-)>}},$$

where $\psi(\cdot)$ again represents the feature vector (Fourier transform in our case). Then, we solve for the weights v to maximize the log-likelihood as follows:

$$v^* = \arg\min - \sum_{i=1}^{P} \log(P_v(s_i^+, s_i^-)). \qquad (4)$$

In particular, we use stochastic gradient decent (SGD) for solving this problem. Finally, for a test pair (s_1, s_2), if $e^{<v^*, \psi(s_1)>} > e^{<v^*, \psi(s_2)>}$, we say s_1 is preferred over s_2; and we say s_2 is preferred over s_1 otherwise.

B. Comparison of Solution Approaches

In this section, we compare the performance of the proposed solution approaches with baseline preference learning methods listed in Section V-A. We use the percentage of trains (test) pairs a model accurately predicts as the metric for comparison.

For each participant, we randomly split the preference set by $70\% - 30\%$ as train-test data ten times, i.e., 35 pairs for training set and 15 for test set. For each train-test split, we compute accuracy of train-test datasets with respect to traditional STL, and compare two proposed computational approaches with two baseline methods. The first method solves Problem 2 using ϕ^{stop} an ϕ^{pedes} for respective scenarios, via random sampling with a threshold condition, where we sample 1000 weight valuations per split. For stop sign scenario, the second method solves Problem 2 with gradient-based optimization over loss function $\mathcal{L}$ initialized from ten random weight valuations and from the traditional STL valuation. We report the best training/test accuracy pair among these 11 as the result. Learning rate is 10^{-5}, $\epsilon = 0.01$, and $\theta = 0.01$. Softness coefficient for `softmax/min` is $\beta = 10^{10}$. We terminate the optimization when the cost value difference drops below 10^{-6}. We divide the training set into batches of five pairs. Batch selection is random at each iteration. We skip the gradient-based method for the pedestrian scenario since it was too slow to converge with each iteration taking a long time due to formula complexity. The third method is SVM classification baseline. For the last method, we solve Equation (4) via SGD with learning rate 0.1. Results for both scenarios are summarized in Figures 2a and 2b.

The results indicate that simple random sampling effectively identifies promising weight valuations that improve the traditional STL accuracy, and gives comparable results to gradient-based optimization for the intersection with stop sign scenario. We see that for almost all scenarios, proposed solutions give competitive results to baseline solutions to the least. The average performance of methods for all users and all splits are shown in Table I. While Bradley-Terry performs better in average training accuracy, it generalizes significantly worse than others. Random sampling gives results close to Bradley-Terry on average training set accuracy and generalizes better than all other methods in both scenarios.

TABLE I: Average accuracy results for different methods on human subject studies. Values represent the average accuracy over all splits and all users for each method.

Method	RS (ours)		GB (ours)		BT		SVM	
Accuracy	Train	Test	Train	Test	Train	Test	Train	Test
Stop sign	79%	**71%**	77%	67%	**82%**	58%	71%	66%
Pedestrian	83%	**81%**	N/A	N/A	**86%**	78%	79%	80%

Finally, when we look at Figure 2a, we see that with decreasing consistency, generalizability of all methods decreases, i.e., they perform poorly on test data.

Now we turn our attention to safety of different approaches. Ideally, when presented with a pair of signals where one is violating the traffic rules and the other one satisfying, an approach should give preference to the satisfying one. Our method satisfies this nice property by construction. To test how the baselines do in this case, we simulate hundred violating pairs for the intersection with stop sign scenario, and pair them with satisfying signals. Now, we have fifty satisfying-satisfying signal pairs that we use in human subject studies and hundred satisfying-violating signal pairs. We create two different training sets: (i) one with all satisfying-satisfying pairs only, and (ii) one with all satisfying pairs and fifty satisfying-violating pairs. Test sets are hundred satisfying-violating pairs, and fifty satisfying-violating pairs, respectively. Table II shows the safety performance of two baseline methods, and Random Sampling. As we can see, baseline methods trained with satisfying signals only performs poorly when encountered with violating signals. However, it is not always feasible to generate violating real-life behaviors for safety-critical scenarios. When training baseline methods, we rely on simulators to generate violating signals, which may lack of reality. When we look at results with training set (ii), test performance of both baseline methods increases but can never reach (let alone ensure) 100%. This shows that there is at least one pair that the method chooses the violating signal over satisfying one.

TABLE II: Safety-critical selection comparison with baseline methods. The test values indicate the percentage of test cases for which the learned model prefers a rule-following (safe) behavior to a rule-violating (unsafe) one.

Method	RS (ours)		BT		SVM	
Trained with	(i)	(ii)	(i)	(ii)	(i)	(ii)
Training Accuracy	92%	96%	84%	83%	76%	83%
Test Accuracy	**100%**	**100%**	17%	94%	15%	98%

VI. Conclusion, Limitations, and Future Work

In this work, we introduce a safe preference learning approach and evaluate its performance in two different driving scenarios. Considering three desirable properties of preference learning for safe personalization mentioned in the Introduction, our results show that our method gives competitive results with the baselines in terms of expressivity but significantly outperforms them in terms of safety. Moreover, it is not clear how models learned by generic preference learning methods can be used in control design, whereas our STL-based method can be readily integrated into control synthesis.

We note that neither random-sampling nor gradient-based method guarantee finding an optimal value. We also observe the gradient-based method to have difficulties in convergence for certain formulas. It would be interesting to study different smooth robustness metrics to see if they can mitigate this issue. While preference data in our experiments appears to be on a smaller scale, expecting humans to select preferences for hundreds of signal pairs all at once is impractical. Our experience shows that even dealing with fifty pairs could be overwhelming. To this end, our upcoming focus is on an active learning scheme that maximizes inference using minimum amount of question pairs.

References

[1] R. Karagulle, N. Arechiga, A. Best, J. Decastro, and N. Ozay, "Poster abstract: Safety guaranteed preference learning approach for autonomous vehicles," ser. HSCC '23. New York, NY, USA: ACM, 2023.

[2] M. Hasenjäger and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *2017 IEEE 20th Intl. Conf. on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–7.

[3] S. Y. Park, D. J. Moore, and D. Sirkin, "What a driver wants: User preferences in semi-autonomous vehicle decision-making," in *Proc. of the 2020 CHI Conf. on Human Factors in Computing Systems*. ACM, 2020, p. 1–13.

[4] H. Bellem, B. Thiel, M. Schrauf, and J. F. Krems, "Comfort in automated driving: An analysis of preferences for different automated driving styles and their dependence on personality traits," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 55, pp. 90–100, 2018.

[5] C. Basu, Q. Yang, D. Hungerman, M. Sinahal, and A. D. Draqan, "Do you want your autonomous car to drive like you?" in *2017 12th ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI*, 2017, pp. 417–425.

[6] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2019.

[7] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conf. on Decision and Control*, 2014, pp. 81–87.

[8] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *IEEE Trans. on Robotics*, vol. 23, no. 2, pp. 320–330, 2007.

[9] G. Bombara and C. Belta, "Online learning of temporal logic formulae for signal classification," in *2018 European Control Conf. (ECC)*, 2018, pp. 2057–2062.

[10] D. Li, M. Cai, C.-I. Vasile, and R. Tron, "Learning signal temporal logic through neural network for interpretable classification," in *2023 American Control Conf. (ACC)*, 2023, pp. 1907–1914.

[11] A. G. Puranic, J. V. Deshmukh, and S. Nikolaidis, "Learning performance graphs from demonstrations via task-based evaluations," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 336–343, 2023.

[12] N. Mehdipour, C.-I. Vasile, and C. Belta, "Specifying user preferences using weighted signal temporal logic," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2006–2011, 2021.

[13] G. A. Cardona, D. Kamale, and C.-I. Vasile, "Mixed integer linear programming approach for control synthesis with weighted signal temporal logic," in *Proc. of the 26th ACM Intl. Conf. on Hybrid Systems: Computation and Control*. New York, NY, USA: ACM, 2023.

[14] J. Fürnkranz and E. Hüllermeier, *Preference learning*. Springer Berlin Heidelberg, 2011.

[15] K. Martyn and M. Kadziński, "Deep preference learning for multiple criteria decision analysis," *European Journal of Operational Research*, vol. 305, no. 2, pp. 781–805, 2023.

[16] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions," *Robotics: Science and Systems*, vol. 13, 2017.

[17] E. Biyik and D. Sadigh, "Batch active preference-based learning of reward functions," in *Proc. of The 2nd Conf. on Robot Learning*, vol. 87. PMLR, 29–31 Oct 2018, pp. 519–528.

[18] M. Tucker, N. Csomay-Shanklin, W.-L. Ma, and A. D. Ames, "Preference-based learning for user-guided hzd gait generation on bipedal walking robots," in *2021 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021, pp. 2804–2810.

[19] R. Cosner, M. Tucker, A. Taylor, K. Li, T. Molnar, W. Ubelacker, A. Alan, G. Orosz, Y. Yue, and A. Ames, "Safety-aware preference-based learning for safety-critical control," in *Proc. of The 4th Annual Learning for Dynamics and Control Conf.*, vol. 168. PMLR, 2022, pp. 1020–1033.

[20] E. Plaku and S. Karaman, "Motion planning with temporal-logic specifications: Progress and challenges," *AI Communications*, vol. 29, pp. 151–162, 2016.

[21] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. D. Ames, J. W. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Trans. on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2016.

[22] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[23] A. Linard, I. Torre, B. Ermanno, A. Sleat, I. Leite, and J. Tumova, "Real-time rrt* with signal temporal logic preferences," in *Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[24] D. Neider and I. Gavran, "Learning linear temporal properties," in *2018 Formal Methods in Computer Aided Design (FMCAD)*, 2018, pp. 1–10.

[25] Z. Xu, M. Ornik, A. A. Julius, and U. Topcu, "Information-guided temporal logic inference with prior knowledge," in *2019 American Control Conf. (ACC)*, 2019, pp. 1891–1897.

[26] G. Chou, N. Ozay, and D. Berenson, "Explaining multi-stage tasks by learning temporal logic formulas from suboptimal demonstrations," in *Robotics: Science and Systems (RSS)*, 2020.

[27] Y. Jiang, S. Bharadwaj, B. Wu, R. Shah, U. Topcu, and P. Stone, "Temporal-logic-based reward shaping for continuing reinforcement learning tasks," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 35, no. 9, 2021, pp. 7995–8003.

[28] X. Li, Y. Ma, and C. Belta, "A policy search method for temporal logic specified reinforcement learning tasks," in *2018 Annual American Control Conf. (ACC)*, 2018, pp. 240–245.

[29] R. Karagulle, N. Aréchiga, J. DeCastro, and N. Ozay, "Classification of driving behaviors using stl formulas: A comparative study," in *Formal Modeling and Analysis of Timed Systems*. Springer Intl. Publishing, 2022, p. 153–162.

[30] H. Wang, H. He, W. Shang, and Z. Kan, "Temporal logic guided motion primitives for complex manipulation tasks with user preferences," in *2022 Intl. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 4305–4311.

[31] R. Yan, A. Julius, M. Chang, A. Fokoue, T. Ma, and R. Uceda-Sosa, "Stone: Signal temporal logic neural network for time series classification," in *2021 Intl. Conf. on Data Mining Workshops (ICDMW)*, 2021, pp. 778–787.

[32] N. Fronda and H. Abbas, "Differentiable inference of temporal logic formulas," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 11, pp. 4193–4204, 2022.

[33] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, Y. Lakhnech and S. Yovine, Eds. Springer Berlin Heidelberg, 2004, pp. 152–166.

[34] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Formal Modeling and Analysis of Timed Systems*. Springer Berlin Heidelberg, 2010, pp. 92–106.

[35] P. Varnai and D. V. Dimarogonas, "On robustness metrics for learning stl tasks," in *2020 American Control Conf. (ACC)*, 2020, pp. 5394–5399.

[36] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Proc. of the Twenty-Third Intl. Joint Conf. on Artificial Intelligence*, 2013, p. 854–860.

[37] X. Li, G. Rosman, I. Gilitschenski, C.-I. Vasile, J. A. DeCastro, S. Karaman, and D. Rus, "Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3459–3466, 2021.

[38] K. Leung, N. Arechiga, and M. Pavone, "Back-propagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods," in *Algorithmic Foundations of Robotics XIV*. Springer Intl. Publishing, 2021, pp. 432–449.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Intl. Conf. on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conf. Track Proceedings*, 2015.

[40] Y. Kantaros and M. M. Zavlanos, "Sampling-based optimal control synthesis for multirobot systems under global temporal tasks," *IEEE Trans. on Automatic Control*, vol. 64, no. 5, pp. 1916–1931, 2019.

[41] H. Mania, A. Guy, and B. Recht, "Simple random search of static linear policies is competitive for reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[42] F. Aiolli and A. Sperduti, *A Preference Optimization Based Unifying Framework for Supervised Learning Problems*. Springer Berlin Heidelberg, 2011, pp. 19–42.

[43] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.