A Safe Preference Learning Approach for Personalization With Applications to Autonomous Vehicles

Ruya Karagulle, Nikos Aréchiga, Andrew Best, Jonathan DeCastro, and Necmiye Ozay, *Senior Member, IEEE*

***Abstract*—This letter introduces a preference learning method that ensures adherence to given specifications, with an application to autonomous vehicles. Our approach incorporates the priority ordering of Signal Temporal Logic (STL) formulas describing traffic rules into a learning framework. By leveraging 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 that can be used in correct-and-custom-by-construction controller synthesis. We demonstrate the performance of our method with a pilot human subject study 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.**

***Index Terms*—Formal specifications, machine learning algorithms, vehicle safety.**

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. In particular for autonomous driving, surveys have demonstrated that drivers have different comfort and performance preferences while driving in different scenarios and conditions [2], [3], [4]. Moreover, drivers tend to prefer different driving styles for autonomous vehicles than their own styles [5]. Customizing autonomous vehicles based on user preferences can increase user satisfaction with these vehicles. However, autonomous systems often require the satisfaction of a set of rules for safe operation. Relying on human preferences alone 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 the safety of all agents in the environment. Preference learning algorithms for safety-critical operations must consider rule satisfaction. The main motivation for 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 where 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 letter, we propose an integrated framework for personalization and safety to satisfy all of these properties by using Signal Temporal Logic (STL). STL is a variant of temporal logic that is tailored for reasoning about the temporal properties of time series data and is commonly used in describing correct behaviors in a variety of autonomous systems [6], [7], [8], [9], [10], [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 the required parameters for the 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 a gradient-based approach, which utilizes computation graphs to calculate the WSTL robustness of signals.

Manuscript received 27 September 2023; accepted 13 February 2024. Date of publication 11 March 2024; date of current version 25 March 2024. This letter was recommended for publication by Associate Editor D. Brscic and Editor A. Peer upon evaluation of the reviewers' comments. This work was supported by Toyota Research Institute. A poster abstract on our preliminary results was presented at HSCC [1]. *(Corresponding author: Ruya Karagulle.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by IRB process by University of Michigan, under Application No. HUM00221976.

Ruya Karagulle and Necmiye Ozay are with the Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48103 USA (e-mail: ruyakrgl@umich.edu; necmiye@umich.edu).

Nikos Aréchiga, Andrew Best, and Jonathan DeCastro are with the Toyota Research Institute, Los Altos, CA 94043 USA (e-mail: nikos.arechiga@tri.global; andrew.best@tri.global; jonathan.decastro@tri.global).

Digital Object Identifier 10.1109/LRA.2024.3375626

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 a crosswalk while a pedestrian is crossing the road. We generate two sets of trajectories that comply with traffic rules for these scenarios and run a pilot human subject study with eight participants for both scenarios. Comparisons with baseline preference learning methods verify the need for safety-aware preference learning by showing that baseline methods usually lead to unsafe selections, whereas our method does not.

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 ratings 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], [17], [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 learning applications [9], [10], [11], [24], [25], [26], [27], [28] in many autonomous systems. In particular, works in [9], [10], [24], [25], [29] try to infer a temporal logic formula for classification from the data. As a subset of learning applications, in robot learning, Chou et al. [26] try to learn task specifications in linear temporal logic from demonstrations, Puranic et al. [11] score 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], [31], [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 priorities or preferences. The work in [30] defines Weighted Truncated Linear Temporal Logic. Both works assume that they have knowledge of the formula and associated weights. For the end-user, it is hard to interpret the weights and define their preferences in the 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 $\mathcal{S} \subseteq \mathbb{R}_e^m$ is a m dimensional extended 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 $f_\pi : \mathcal{S} \to \mathbb{R}_e$ 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 are defined as follows:

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

Derived operators have the following qualitative semantics:

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

For qualitative semantics at time instant $t = 0$, we omit t and write $s \models \phi$. STL also has quantitative semantics to measure how well the signal models the formula. There are different quantitative semantics, also known as robustness metrics [34], [35]. In this letter, we use the traditional robustness metric $\rho : \mathcal{S} \times \mathcal{F} \times \mathbb{T} \to \mathbb{R}_e$ from [34], defined recursively as[2]:

$$
\begin{array}{rcl}
\rho(s, \top, t) & = & \infty, \\
\rho(s, \pi, t) & = & f_\pi(s(t)), \\
\rho(s, \neg\phi, t) & = & -\rho(s, \phi, t), \\
\rho(s, \phi_1 \wedge \phi_2, t) & = & \min\left(\rho(s, \phi_1, t), \rho(s, \phi_2, t)\right), \\
\rho(s, \phi_1 \mathcal{U}_{[a,b]} \phi_2, t) & = & \max\limits_{t' \in [t+a, t+b]} \Big(\min\Big(\rho(s, \phi_2, t'), \\
 & & \min\limits_{t'' \in [t, t']} \rho(s, \phi_1, t'')\Big)\Big).
\end{array}
$$

The robustness for derived operators is given by

$$
\begin{array}{rcl}
\rho(s, \phi_1 \vee \phi_2, t) & = & \max\left(\rho(s, \phi_1, t), \rho(s, \phi_2, t)\right), \\
\rho(s, \Diamond_{[a,b]} \phi, t) & = & \max\limits_{t' \in [t+a, t+b]} \rho(s, \phi, t'), \\
\rho(s, \Box_{[a,b]} \phi, t) & = & \min\limits_{t' \in [t+a, t+b]} \rho(s, \phi, t').
\end{array}
$$

The robustness value 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]. The robustness metric ρ is sound,

[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 $\Box_{[a,b]}\phi = \neg(\Diamond_{[a,b]} \neg\phi)$.

[2] To represent Boolean-valued quantities, we use signals $p : \mathbb{T} \to \{-\infty, +\infty\}$ and simply write p as a predicate instead of $p \geq 0$. If such a signal is the i^{th} coordinate of s and $\pi(s(t)) := e_i^T s(t) \geq 0$ with e_i being the i^{th} natural basis vector, we have $\rho(s, \pi, t) = e_i^T s(t) = p(t)$.

i.e., $\rho(s,\phi,t) > 0 \Rightarrow (s,t) \models \phi$ and $\rho(s,\phi,t) < 0 \Rightarrow (s,t) \not\models \phi$ [37].

Example 1: Let $s = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^T = \begin{bmatrix} 1 & -1 & -2 & -2 \\ 1 & 1 & 1 & 2 \end{bmatrix} \in \mathbb{R}^{2\times 4}$ be a two-dimensional signal with the time length $t_{final} = 3$. Let $\phi_{\text{STL}} = \Diamond(-s_1 \geq 0 \wedge s_2 \geq 0)$ 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 at time $t = 0$ over ϕ_{STL} is

$$\rho(s,\phi_{\text{STL}}) = \max_{t'\in[0,3]} (\min(-s_1(t'), s_2(t'))) = 2.$$

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

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

Derived operators have WSTL robustness definitions as:

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

with $r(s,\phi)$ denoting the WSTL robustness at $t = 0$.

Note that since we have $\top$ when defining Eventually from Until, and since the WSTL robustness of $\top$ is ∞, we drop the set of weights w^2 in the WSTL robustness of Eventually because they do not affect the result of the `min` operation in the computation of the WSTL robustness of Until. That is the reason why Eventually (and hence Always) has fewer weights than Until. Moreover, the Boolean true, predicates, and negation operators do not have associated weights, i.e., these operators have weights equal to 1.

Example 1: (cont.) Let ϕ be the weighted version of ϕ_{STL} with weights $\{w^\Diamond_i\}^4_{i=1} = [1.5, 0.3, 3, 1.2]$ and $\{w^\wedge_i\}^2_{i=1} = [1, 2]$. The WSTL robustness of s at time $t = 0$ is

$$r(s,\phi) = \max_{t'\in[0,3]} (w^\Diamond_{t'+1} \min(-w^\wedge_1 s_1(t'), w^\wedge_2 s_2(t'))) = 6.$$

The following result is adapted from Theorem 2 in [12].

Lemma 1: Let $\tilde{r} : \mathcal{S} \times \mathcal{F} \times \mathbb{T} \to \mathbb{R}_e$ be a quantitative semantics. For a WSTL formula ϕ, let ϕ_{STL} be the STL formula obtained by removing the weights in ϕ. If $\text{sign}(\rho(s,\phi_{\text{STL}},t)) = \text{sign}(\tilde{r}(s,\phi,t))$ for all $(s,\phi,t) \in \mathcal{S} \times \mathcal{F} \times \mathbb{T}$ (i.e., $\tilde{r}$ is *sign-consistent*), then $\tilde{r}$ 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 as positive, multiplying a robustness value with a weight does not change its sign. Therefore, for each recursive operation in the WSTL robustness calculation, the sign of the robustness value associated with this recursion step is preserved. Then, we see that $\text{sign}(\rho(s,\phi_{\text{STL}},t)) = \text{sign}(\tilde{r}(s,\phi,t))$ for all $(s,\phi,t) \in \mathcal{S} \times \mathcal{F} \times \mathbb{T}$ and for all non-negative weights. ■

In the WSTL definition of [12], weights are pre-determined positive real values. In this letter, we use an extension to WSTL that we call Parametric Weighted Signal Temporal Logic (PWSTL) in which some of the weights are unknown parameters and the remaining weights are given constants (cf., [31]). We denote the set of unknown parameters as $\mathcal{W}$ and denote PWSTL formulas as $\phi_\mathcal{W}$, where we omit the known weights with slight abuse of notation since for most of the results in the letter $\mathcal{W}$ is the entire weight set. A PWSTL formula results in a WSTL formula $\phi_{\mathcal{W}=w}$ with the valuation w of the parameters.

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 $\mathcal{P} := \{(s^+_i, s^-_i)\}^P_{i=1}$ 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 reflects safety specifications. Formally, this letter aims to solve the following problem:

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

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

Problem 1 is a feasibility problem. In the next subsection, we reformulate it as an optimization problem to be able to handle infeasibility. Before doing so, we provide an analysis of the set of feasible weights using the syntax tree of STL formulas. An STL formula has 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 [38]. The syntax tree associated with the STL formula in Example 1 is given in Fig. 1. 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 ϕ of Example 1, the root of the syntax tree is $\Diamond$ operator and the root weights are $\{w^\Diamond_i\}^4_{i=1}$. Consider another example with $\tilde{\phi} = \neg\phi$, although the root operator is $\neg$ since it is not a weighted operator, we need to look at its children until we find a weighted operator. Hence, the root weights are again $\{w^\Diamond_i\}^4_{i=1}$.

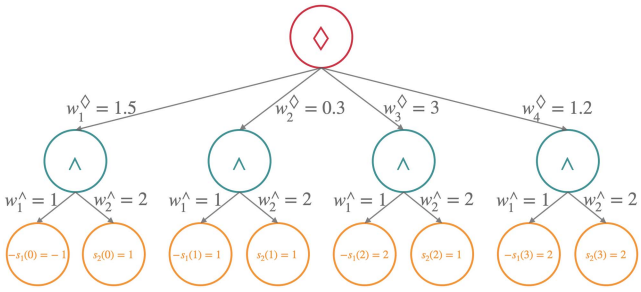


Fig. 1. Syntax tree of ϕ of Example 1.

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

Lemma 2: Let $\phi_{\mathcal{W}}$ be a PWSTL formula with weight set $\mathcal{W}$ containing only the weight parameters for the root weights of ϕ. Other weights of ϕ are fixed constants. 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-level 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 $\mathcal{D} = \mathcal{B}_\infty(0) \cap \mathbb{R}^n_+$, 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 $\bar{w}$ in the domain $\mathcal{D}$ such that $\phi_{\mathcal{W}=\bar{w}}$ 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, $\phi_{s,\bar{w}_s}$. Define $\bar{w}_s = w_s / \max(w_s)$. Clearly, $r(s, \phi_{s,w_s}, t) = \max(w_s) r(s, \phi_{s,\bar{w}_s}, t)$ and all weights of $\phi_{s,\bar{w}_s}$ 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 $\bar{w}_s$ are replaced by w_u and w_s, we achieve the same WSTL 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 $\mathcal{D}$. ■

We illustrate the proof with our running example.

Example 1: (cont.) The weight values w of ϕ are not in $\mathcal{D}$. We can construct a new weight valuation that preserves preference orders using Theorem 2. Let us denote ϕ as $\phi_{\mathcal{W}=w}$ and consider two signals x and y with robustness order $r(x, \phi_{\mathcal{W}=w}) > r(y, \phi_{\mathcal{W}=w})$. At level $k = 2$, we have $\max_i(w_i^{\wedge}) = 2$. Define $\tilde{w}^{\wedge} = w^{\wedge} / \max_i(w_i^{\wedge}) = [0.5, 1]$. Then, scale the weights in the upper level with $\max_i(w_i^{\wedge})$ and obtain $\tilde{w}^{\Diamond} = [3, 0.6, 6, 2.4]$. Note that $r(x, \phi_{\mathcal{W}=w}) = r(x, \phi_{\mathcal{W}=\tilde{w}})$ and $r(y, \phi_{\mathcal{W}=w}) = r(y, \phi_{\mathcal{W}=\tilde{w}})$. Now, let $k = 1$, which is the root level, and consider $\mathcal{W} = \tilde{w}$. Leave the lower levels as is: $\bar{w}^{\wedge} = \tilde{w}^{\wedge}$ and scale the root weights as $\bar{w}^{\Diamond} = \tilde{w}^{\Diamond} / \max_i(\tilde{w}_i^{\Diamond}) = [0.5, 0.1, 1, 0.4]$. By root-level homogeneity, we know that, if $r(x, \phi_{\mathcal{W}=\tilde{w}}) > r(y, \phi_{\mathcal{W}=\tilde{w}})$, which is the case by construction of $\tilde{w}$, then $r(x, \phi_{\mathcal{W}=\bar{w}}) > r(y, \phi_{\mathcal{W}=\bar{w}})$. Hence, $\bar{w}$ preserves the orders and it 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.[3]

By construction of the problem (3), we have the following result that states that relates the solution of this optimization problem to Problem 1.

Proposition 1: 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).

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 prefers 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 for certain STL formulas ϕ_{STL}, the corresponding WSTL robustness metric satisfies $r(s, \phi_{\mathcal{W}=w}) \leq 0$ for all signals s and all weights $w \in \mathbb{R}^n_+$. In this case, the

[3]Since the objective function takes only finitely many values, it always has a minimum. Therefore, searching for arg min is valid.

problem of learning weights only from rule-satisfying signals is not meaningful since these signals will have $r(s, \phi_{\mathcal{W}=w}) = 0$ for any weight w; and satisfaction cannot be deduced when $r(s, \phi_{\mathcal{W}=w}) = 0$. While such formulas can be uncommon in certain domains, we find them to be common in driving scenarios when the specification involves coming to a full stop or a Boolean indicator for pedestrians or traffic lights. Hence, we discuss a workaround to enable preference learning among satisfying signals for a class of such formulas but similar workarounds can be devised for other cases too. Consider a formula of the form $\phi = \phi_1 \wedge \square(f(s) \geq 0 \ \wedge \ -f(s) \geq 0)$, second part of which essentially represents an equality $f(s) = 0$. To enable preference learning in this case, we append s with a new coordinate b such that $b(t) = \infty$ if $f(s(t)) = 0$ and $b(t) = -\infty$ otherwise; and replace the formula with $\phi' = \phi_1 \wedge \square b$. With this transformation, if for all $t \in [0, t_{final}]$, $b(t) = \infty$, the WSTL robustness of ϕ' is determined by ϕ_1; otherwise the WSTL robustness of ϕ' becomes $-\infty$, which indicates a violation of ϕ. Applying our method to ϕ' allows us to rank rule-satisfying signals.

B. Computational Approach

We note that Problem 2 is highly non-convex and non-differentiable. 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 back-propagation in machine learning, many temporal logic learning algorithms using gradients have been proposed [39]. To be able to compute the gradient, we need a differentiable loss function. In the WSTL robustness definition, we replace `max` and `min` functions with their soft differentiable versions `softmin/softmax` as in [39]. We also replace the indicator function with the logistic function with a shift. The shift helps avoid 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 (3) and the second term promotes the norm of the weights W_ϕ 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 (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: Inspired by [39], we construct a computation graph for the robustness of WSTL formulas from syntax trees. This computation graph takes a signal as input and returns the WSTL robustness value of that signal at all times as output. We use PyTorch along with Adam [40] optimizer. Several strategies are investigated to mitigate convergence issues of the gradient-based method: (i) decreasing the softness coefficient β of `softmin/max`, possibly compromising the soundness guarantee, (ii) decreasing the steepness of the logistic function, i.e., decreasing M, but this makes the surrogate $\mathcal{L}$ less similar to the objective in Problem 2, (iii) 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 [41], especially when there is a multitude of feasible solutions. Similarly in [42], 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 $\mathcal{D} = \mathcal{B}(0)_\infty \cap \mathbb{R}_+^n$. We uniformly sample weight valuations in $\mathcal{D}$.

Implementation details: We want weight valuations such that the absolute difference in robustness of 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,[4] this separation helps start the iterations at a part of the weight space where the 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

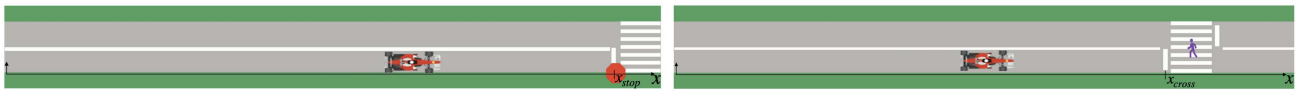
In this section, we provide a comparison of solution approaches with baseline methods, along with demonstrating the need for a safety-guaranteed preference learning framework.[5] We also showcase the framework's performance in capturing the personal preferences of different participants in a human subject study. For these purposes, we use two different driving scenarios.

Driving Scenarios: We use STL to specify traffic rules in driving scenarios. The first scenario is a simple intersection with a stop sign, a screenshot is shown in Fig. 2(a). 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 as follows: $\phi^{stop} = \lozenge\square(x - x_{stop} \geq 0 \wedge v = 0) \wedge \square(v \geq 0)$ 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 condition. We substitute $v = 0$ with an indicator variable as discussed in Remark 1. We construct the PWSTL formula $\phi_{\mathcal{W}}^{stop}$ with a weight parameter set $\mathcal{W}$ that contains all weights in the formula. In the second scenario, we observe an ego vehicle approaching a pedestrian while she is crossing the road, as illustrated in Fig. 2(b). The traffic regulation, in this case, is expressed in STL form as $\phi^{pedes} = \square[(p \wedge (x - x_{cross} \leq 0)) \Rightarrow (x - x_{cross} \leq 0 \mathcal{U} \neg p) \wedge (v \leq v_{lim})]$ where x, v represent position and velocity signals, respectively. Boolean signal p indicates the presence of a pedestrian, and v_{lim} and x_{cross} are constants denoting the speed limit and the crosswalk position, respectively.

Human Subject Studies: Studies are completed under IRB Study No. HUM00221976. For each scenario, we collaborate with eight participants with a 75–25% male-female ratio from the 25–35 age group. We simulate a hundred trajectories that satisfy the temporal logic formula per scenario. We compose

[4]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.

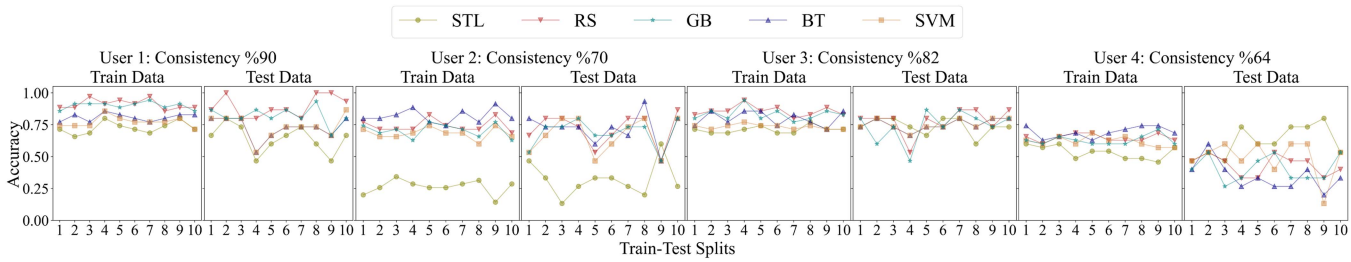
[5]The code and the data can be accessed from https://github.com/ruyakrgl/SPL-WSTL



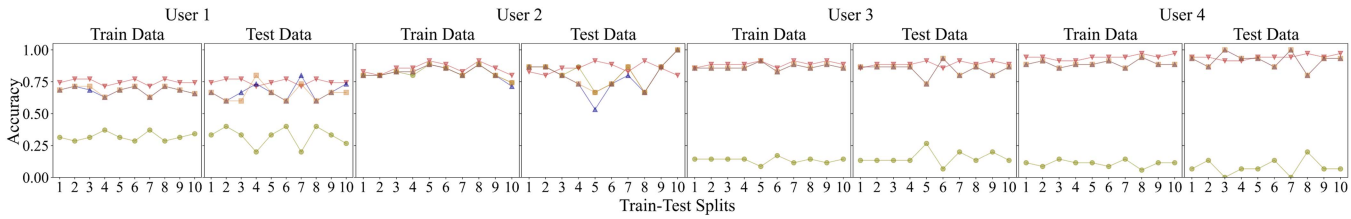
(a) Stop Sign Scenario: Vehicle approaching to an intersection with a stop sign. The traffic rule says that vehicles should stop before the stop sign.

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

Fig. 2. Two scenarios that are used for experiments.



(a) Intersection with a stop sign scenario



(b) Approaching to a pedestrian scenario

Fig. 3. Human subject study results for the two scenarios for four of the 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.

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 get a measure of the participant's decisiveness. The consistency levels of four participants are reported in Fig. 3(a).

A. Baseline Methods

One well-known approach to pairwise preference learning problem is to recast it as a supervised learning problem [43]. Let $\psi(s)$ be the feature vector of item s. We construct $\psi(\cdot)$ by dividing the Fourier transform of s into five frequency bins and adding the traditional robustness metric as the final feature. 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 signal pairs in $\mathcal{P}$ belong 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), with a radial basis function kernel, to learn a binary classifier. 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, the Bradley-Terry model is a common likelihood function model in preference learning applications [44]. The Bradley-Terry model [44] 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 described earlier. Then, we solve for weights v to maximize the log-likelihood as follows:

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

In particular, we use stochastic gradient descent (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 the baseline methods listed in Section V-A. We use the percentage of training (test) pairs that a model accurately predicts as the metric for comparison.

TABLE I
AVERAGE ACCURACY RESULTS FOR DIFFERENT METHODS ON HUMAN SUBJECT STUDIES

Method	RS (ours)		GB (ours)		BT		SVM	
Accuracy	Train	Test	Train	Test	Train	Test	Train	Test
Stop sign	81.2%	**77.0%**	80.2%	72.4%	**82.7%**	75.7%	78.9%	76.7%
Pedestrian	**91.5%**	**91.4%**	N/A	N/A	88.4%	88.4%	88.3%	88.7%

Values represent the average accuracy over all splits and all eight users for each method.
Bold values represent best accuracies in the train and test data for both scenarios.

TABLE II
SAFETY-CRITICAL SELECTION COMPARISON WITH BASELINE METHODS

Method	RS (ours)		BT		SVM	
Trained with	(i)	(ii)	(i)	(ii)	(i)	(ii)
Training Accuracy	92%	96%	78%	82%	76%	85.33%
Test Accuracy	100%	100%	40%	87%	31%	**100%**

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.
Bold values represent best accuracies in the train and test data for both scenarios.

For each participant, we use ten random 70%–30% splits of the preference set in to train-test data, i.e., 35 pairs for the training set and 15 for the test set. For each split, we compute the train-test accuracy with respect to traditional STL and compare two proposed approaches with two baseline methods. Our first method solves Problem 2 using ϕ^{stop} and ϕ^{pedes} for respective scenarios, via random sampling with a threshold condition, where we sample 1000 weight valuations per split. For the stop sign scenario, our second method solves Problem 2 with gradient-based optimization over the loss function $\mathcal{L}$, initialized eleven times, ten from random weight valuations and one from the traditional STL valuation. We report the best training/test accuracy pair among these 11 as a result. The learning rate is 10^{-5}, $\epsilon = 0.01$, and $\theta = 0.01$. The softness coefficient for $\texttt{softmax}$ 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. The third method is the SVM classification baseline. For the last method, we solve (4) via SGD with the learning rate of 0.1.

Some representative results are shown in Figs. 3(a) and (b). The average performance of all methods for all users and splits is shown in Table I. Random sampling gives competitive results with the baseline methods in the stop sign scenario, and outperforms all methods in the pedestrian. Although the gradient-based method gives equally good results as random sampling for the stop sign scenario, it is much slower. Indeed, it was too slow to converge for the pedestrian scenario, due to formula complexity, and is skipped. We conclude that simple random sampling effectively identifies promising weight valuations that improve the traditional STL accuracy, and give comparable results to other methods.

Finally, when we look at Fig. 3(a), we see that with decreasing consistency, the generalizability of all methods decreases, i.e., they perform poorly on test data.

Now we turn our attention to the safety of different approaches. Ideally, when presented with a pair of signals where one is violating the traffic rules and the other is 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 a hundred violating pairs for the intersection with a 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 a hundred satisfying-violating signal pairs. We create two different training sets: (i) one with fifty satisfying-satisfying pairs, and (ii) one with fifty satisfying-violating pairs in addition to satisfying-satisfying 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 for the stop sign scenario and one participant. As we can see, baseline methods trained with only satisfying signals perform poorly when encountered with violating signals. However, it is not always feasible to generate real-life behaviors that violate a rule for safety-critical scenarios. When training baseline methods, we rely on simulators to generate violating signals, which may not be realistic. When we look at the results with training set (ii), the test performance of both baseline methods increases, and SVM reaches 100% accuracy. However, none of the baseline methods ensure the safety of arbitrary safe-unsafe pairs.

VI. CONCLUSION, LIMITATIONS, AND FUTURE WORK

This letter introduced a safe preference learning approach and evaluated 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 guarantees 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 a minimum amount of question pairs. In addition, we aim to integrate the final WSTL formula into a downstream control synthesis algorithm as in [13] to demonstrate its use in control design and to run further validation studies.

REFERENCES

[1] R. Karagulle, N. Arechiga, A. Best, J. Decastro, and N. Ozay, "Poster abstract: Safety guaranteed preference learning approach for autonomous vehicles," in *Proc. 26th ACM Int. Conf. Hybrid Syst.: Computation Control*, 2023, pp. 1–2.
[2] M. Hasenjäger and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 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. Conf. Hum. Factors Comput. Syst.*, 2020, pp. 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," *Transp. Res. F: Traffic Psychol. Behav.*, 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 *Proc. ACM/IEEE Intl. Conf. Hum.-Robot Interact.*, 2017, pp. 417–425.
[6] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Contr. Syst. Lett.*, vol. 3, no. 1, pp. 96–101, Jan. 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 *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 81–87.
[8] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 320–330, Apr. 2007.
[9] G. Bombara and C. Belta, "Online learning of temporal logic formulae for signal classification," in *Proc. Eur. Control Conf.*, 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 *Proc. Amer. Control Conf.*, 2023, pp. 1907–1914.
[11] A. G. Puranic, J. V. Deshmukh, and S. Nikolaidis, "Learning performance graphs from demonstrations via task-based evaluations," *IEEE Robot. Automat. Lett.*, vol. 8, no. 1, pp. 336–343, Jan. 2023.
[12] N. Mehdipour, C.-I. Vasile, and C. Belta, "Specifying user preferences using weighted signal temporal logic," *IEEE Control Syst. Lett.*, vol. 5, no. 6, pp. 2006–2011, Dec. 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. 26th ACM Int. Conf. Hybrid Syst.: Computation Control*, 2023, pp. 1–12.
[14] J. Fürnkranz and E. Hüllermeier, *Preference Learning*. Berlin/Heidelberg, Germany: Springer, 2011.
[15] K. Martyn and M. Kadziński, "Deep preference learning for multiple criteria decision analysis," *Eur. J. Oper. Res.*, 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," in *Proc. Robot.: Sci. Syst.*, 2017, vol. 13, doi: 10.15607/RSS.2017.XIII.053.
[17] E. Biyik and D. Sadigh, "Batch active preference-based learning of reward functions," in *Proc. Conf. Robot Learn.*, 2018, vol. 87, 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 *Proc. IEEE Intl. Conf. Robot. Automat.*, 2021, pp. 2804–2810.
[19] R. Cosner et al., "Safety-aware preference-based learning for safety-critical control," in *Proc. Learn. Dyn. Control Conf.*, 2022, vol. 168, pp. 1020–1033.
[20] E. Plaku and S. Karaman, "Motion planning with temporal-logic specifications: Progress and challenges," *AI Commun.*, vol. 29, pp. 151–162, 2016.
[21] P. Nilsson et al., "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1294–1307, Jul. 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 *Proc. Intl. Conf. Intell. Robots Syst.*, 2023, pp. 8621–8627.
[24] D. Neider and I. Gavran, "Learning linear temporal properties," in *Proc. Formal Methods Comput. Aided Des.*, 2018, pp. 1–10.
[25] Z. Xu, M. Ornik, A. A. Julius, and U. Topcu, "Information-guided temporal logic inference with prior knowledge," in *Proc. Amer. Control Conf.*, 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 *Proc. Robot.: Sci. Syst.*, 2020, doi: 10.15607/RSS.2020.XVI.097.
[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. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 9, pp. 7995–8003.
[28] X. Li, Y. Ma, and C. Belta, "A policy search method for temporal logic specified reinforcement learning tasks," in *Proc. Amer. Control Conf.*, 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*. Berlin, Germany: Springer, 2022, pp. 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 *Proc. Int. Conf. Robot. Automat.*, 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 *Proc. Intl. Conf. Data Mining Workshops*, 2021, pp. 778–787.
[32] N. Fronda and H. Abbas, "Differentiable inference of temporal logic formulas," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4193–4204, Nov. 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. Berlin Heidelberg: Springer, 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*. Berlin Heidelberg: Springer, 2010, pp. 92–106.
[35] P. Varnai and D. V. Dimarogonas, "On robustness metrics for learning STL tasks," in *Proc. Amer. Control Conf.*, 2020, pp. 5394–5399.
[36] G. De Giacomo and M. Y. Vardi, "Linear temporal logic and linear dynamic logic on finite traces," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 854–860.
[37] A. Donzé, T. Ferrère, and O. Maler, "Efficient robust monitoring for STL," in *Proc. Comput. Aided Verification*, 2013, pp. 264–279.
[38] X. Li et al., "Vehicle trajectory prediction using generative adversarial network with temporal logic syntax tree features," *IEEE Robot. Automat. Lett.*, vol. 6, pp. 3459–3466, Apr. 2021.
[39] 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*. Berlin, Germany: Springer, 2021, pp. 432–449.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
[41] Y. Kantaros and M. M. Zavlanos, "Sampling-based optimal control synthesis for multirobot systems under global temporal tasks," *IEEE Trans. Autom. Control*, vol. 64, no. 5, pp. 1916–1931, May 2019.
[42] H. Mania, A. Guy, and B. Recht, "Simple random search of static linear policies is competitive for reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31.
[43] F. Aiolli and A. Sperduti, *A Preference Optimization Based Unifying Framework for Supervised Learning Problems*. Berlin, Heidelberg: Springer, 2011.
[44] 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.