4C: Custom-and-Correct-by-Construction Controller Synthesis using Multi-modal Human Feedback

Ruya Karagulle * **Marco Antonio Valdez Calderon** *
Necmiye Ozay *

* *University of Michigan, Ann Arbor MI USA*
e-mail: {ruyakrgl, mvaldezc, necmiye}@umich.edu

Abstract: Autonomous vehicle technology has a critical role in enhancing road safety by reducing human errors. The adoption of such technologies is shown to be dependent on user satisfaction, which is influenced by diverse comfort and performance preferences. This paper addresses the challenge of generating custom and naturalistic autonomous vehicle behaviors to meet individual user expectations while ensuring safety. To achieve this goal, we incorporate multi-modal human feedback with pairwise preferences and driving demonstrations into a control synthesis framework. We write safety specifications and express preferences using Weighted Signal Temporal Logic (WSTL), and we employ the `PyTeLo` toolbox for controller synthesis. Our approach is validated through synthetic experiments and a pilot, small-sized human subject study, demonstrating the effectiveness of integrating multi-modal human feedback for customizing autonomous vehicle behaviors.

Keywords: autonomous vehicles, safety-critical applications, preference learning, learning from demonstrations

1. INTRODUCTION

Widespread adoption of autonomous vehicle (AV) technology is crucial to reducing accidents caused by human errors and ultimately enhancing road safety. The extent of adoption of such technologies highly depends on users' satisfaction with the output; if the system fails to meet their expectations, they may opt to disable the feature altogether (Hasenjäger and Wersing, 2017). On the other hand, surveys demonstrate that drivers have different comfort and performance preferences Hartwich et al. (2018), therefore it may not be possible to execute behaviors that fit all, which highlights the importance of customizable autonomous vehicle behaviors. However, driving is a safety-critical task, where rules of the road govern the safety standards. Any autonomous vehicle must consistently prioritize safety by following the traffic rules at all times.

Various approaches exist in the literature for the personalization of autonomous vehicles. Studies by Huang et al. (2022); Lefèvre et al. (2016); Kuderer et al. (2015) utilize driving data to mimic the driving style of users. Driving demonstrations are a valuable source to infer naturalistic driving information but surveys suggest that people have different preferences when they are in the driver's seat compared to when they are passengers (Basu et al., 2017). Given that everybody is essentially a passenger in fully autonomous vehicles, imitation learning methods may fall short of satisfying user expectations. Alternatively, Schrum et al. (2024) introduce tuning options to adjust the aggressiveness level of the autonomous vehicle on top of demonstrations from the user. While this approach offers some flexibility, a recent study from Haselberger et al. (2024) confirms that user preferences are dynamic and can vary depending on various factors such as weather conditions or oncoming traffic. Dynamic changes in preferences may require frequent adjustments in aggressiveness level, which may decrease user satisfaction.

Asking users comparison questions is a prominent technique in preference learning (Fürnkranz and Hüllermeier, 2003). Karagulle et al. (2024a,b) attempt the customization problem as a preference learning problem from pairwise comparison questions of behaviors. These works provide safety guarantees by incorporating formal methods with preference learning. Given a scenario and its specifications that describe scenario-specific traffic rules and comfort standards, they learn new semantics for the specifications that represent user preferences. While Karagulle et al. (2024a) pre-define a question set offline, Karagulle et al. (2024b) adaptively select the next question based on question-answer pairs so far. They leave the controller synthesis from the new semantics as a direct extension of their work. Although these works cover hidden and dynamic preferences that cannot be observed from demonstrations, scenario-specific rule descriptions do not include all aspects of the driving task. For instance, consider a scenario where the ego vehicle approaches a pedestrian crossing while a pedestrian crosses the road. Scenario specifications include components related to the longitudinal behavior but do not necessarily include specifications for the lateral behavior. Therefore, we need to consider specifications related to latent variables that help generate a naturalistic behavior in the controller synthesis problem from these specifications. Specifications related to latent variables

* This work was supported in part by NSF TI # 2303564.

may be hard-coded as additional rules or constraints. This task requires domain-expert knowledge and is prone to issues like overlooking some aspects.

In this work, we focus on generating safe, customized, and naturalistic behaviors that are one step closer to being approved and used by potential AV users. To this end, we propose a method for custom-and-correct-by-construction controller synthesis by combining human feedback from two modalities; preferences learned by asking pairwise questions (Karagulle et al., 2024b,a), and driving demonstrations from users. Our framework, called 4C, is summarized in Figure 1. To ensure safety and express preferences, we utilize Weighted Signal Temporal Logic (WSTL), introduced by Mehdipour et al. (2021). Weights in a WSTL formula are learned through “Safe Preference Learning” module. Then, these weights are fed into the controller synthesis problem. Synthesizing controllers that satisfy specifications expressed in temporal logic is an active research area with different temporal logic syntaxes and different control approaches. Wolff et al. (2013) introduce an automaton-guided controller for Linear Temporal Logic, Sadraddini and Belta (2015) present a robust Model Predictive Control (MPC) problem with Signal Temporal Logic constraints, and Saha and Julius (2016) introduce a Mixed-Integer Linear Programming (MILP) Approach for Metric Temporal Logic constraints. In this work, we leverage a recent toolbox `PyTeLo` developed by Cardona et al. (2023), where the MPC problem with WSTL is recast as a MILP with an efficient implementation.

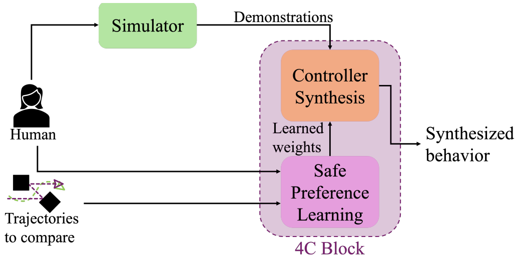

Fig. 1. The goal of this work is to design a custom-and-correct-by-construction controller based on multi-modal human feedback data; user demonstrations and indicated preferences to pairwise questions

We demonstrate the performance of the proposed approach by synthetic experiments and a pilot human subject study. Our synthetic experiments illustrate the distinct effects of demonstrations and learned WSTL semantics over the generated behavior. The human subject study is conducted on a simple simulator that consists of a monitor, a steering wheel, and throttle/brake pedals that implements the proposed framework represented in Figure 1. The pilot human subject study shows that integrating two modalities helps obtain more naturalistic trajectories while guaranteeing safety.

2. PRELIMINARIES AND PROBLEM STATEMENT

We model an autonomous vehicle with a discrete-time system of the form

$$x_{t+1} = f_t(x_t, u_t),\ s_t = h(x_t),\ x_0 = x_o, \tag{1}$$

where $x_t \in \mathcal{X} \subseteq \mathbb{R}^n$ is the $n-$dimensional state at time t, $u_t \in \mathcal{U} \subseteq \mathbb{R}^p$ is the control input, $s_t \in \mathcal{S} \subseteq \mathbb{R}^k$ is the output with $h : \mathcal{X} \mapsto \mathcal{S}$, and $x_0 \in \mathcal{X}$ is the initial state. We are interested in controlling this system over a horizon $[0, T]$. Then, x maps $t \in \mathbb{T} = [0, T] \subseteq \mathbb{Z}_+$ to $\mathcal{X}$. We assume for a given scenario, we have specifications that are related to some of the states. Information related to specifications at time t is represented with s_t. A sequence $x = x_0 x_1 \dots x_T$ is a state trajectory of System (1) if $x_0 = x_o$ and there exists a sequence of control inputs $u = u_0 u_1 \dots u_{T-1}$ that generates x (i.e., $x_{t+1} \in \mathcal{X}$, $u_t \in \mathcal{U}$, and $x_{t+1} = f_t(x_t, u_t)$ for all $t \in [0, T-1]$). A sequence $s = h(x_0)h(x_1)\dots h(x_T)$ is an output trajectory of System (1) if $x = x_0 x_1 \dots x_T$ is a state trajectory.

We define task specifications using Weighted Signal Temporal Logic (WSTL) formulas given by grammar $\phi = \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge^w \phi_2 \mid \phi_1 \mathcal{U}^{w_1,w_2}_{[a,b]} \phi_2$, where $\top$ is the Boolean true, μ is a predicate of the form $g_\mu(s_t) \geq 0$, where $g_\mu : \mathbb{R}^k \mapsto \mathbb{R}$, $\neg$ is the negation operator, $\wedge$ is conjunction, and $\mathcal{U}$ is the until operator, where $a, b \in \mathbb{Z}_+$.[1] Until is a temporal operator with the time interval $[a, b]$, and semantically means that ϕ_1 should hold until ϕ_2 starts to hold in the time interval $[a, b]$. If the time interval is omitted, it reads as $[0, \infty)$. The set of all well-formed STL formulas is denoted as $\mathcal{F}$. WSTL has qualitative and quantitative semantics. 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 $s_t \not\models \phi$. When time t is omitted, i.e. $s \models \phi$, it reads for $t = 0$. We say ϕ is *realizable* by the system (1) if there exists an output trajectory s such that $s \models \phi$. The quantitative semantics of WSTL formulas are denoted as $r : \mathcal{S} \times \mathcal{F} \times \mathbb{T} \mapsto \mathbb{R}_e$ and defined recursively as follows:

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

where $\mathbb{R}_e$ is the extended real domain. Weights of WSTL formulas are in the positive quadrant, that is, $w \in \mathbb{R}_+$ and $w_1, w_2 \in \mathbb{R}_+^{b-a+1}$. Restricting weights to the positive quadrant guarantees the *soundness* property. Soundness is defined as follows: $r(s, \phi, t) > 0 \implies s_t \models \phi$ and $r(s, \phi, t) < 0 \implies s_t \not\models \phi$. Similarly to the qualitative semantics, $r(s, \phi)$ reads for $t = 0$. We say ϕ is *strictly realizable* by the system in (1) if there exists an output trajectory s such that $r(s, \phi) > 0$. The STL and WSTL syntaxes are defined for infinite signals, slight modifications are done for the finite signals along with time interval adjustments (De Giacomo and Vardi, 2013).

An extension to WSTL is Parametric Weighted Signal Temporal Logic (PWSTL), in which a subset of weights are unknown. We denote the PWSTL formulas as $\phi_\mathcal{W}$ where $\mathcal{W}$ is the set of unknown weights. A valuation w for $\mathcal{W}$ makes a WSTL formula. With a slight abuse of notation, we denote WSTL formulas constructed with valuation w for $\mathcal{W}$ as $\phi_{\mathcal{W}=w}$. The valuation w can be inferred using

[1] Other common predicates with $\leq, <, >$ and $=$ relations, as well as other operators such as $\vee, \Box$ and $\Diamond$ can be expressed in the usual way. Details can be found in Mehdipour et al. (2021).

works like Karagulle et al. (2024a,b). Both works require a pre-generated signal set $\mathcal{S}_Q = \{s_1, \dots, s_K\}$ to choose pairwise comparison questions of the form $q = (s_i, s_j)$. In this work, we want to solve the following problem:

Problem 1. Given a scenario with specifications ϕ, a set of demonstrations $(\bar{x}, \bar{u})$, and a set of example trajectories $\mathcal{S}_Q$ to ask pairwise preference questions from, find a controller that generates output trajectories that represent preferences while satisfying ϕ.

3. METHODOLOGY

We propose a two-step approach to solve Problem 1 as illustrated in Figure 1. In the first step, we leverage the method developed by Karagulle et al. (2024b) to learn the most likely weight valuation w^* by adaptively asking questions to the user. They utilize the Bayesian inference method and represent the set of weight valuations as a discrete random variable with M discretization points. In the beginning, all weight valuations have a uniform probability of being the suitable weight valuations for the participant. With each question-answer pair, they update the posterior probability of the weight valuations. Then, the following proposition forms the basis for correctness in synthesis:

Proposition 1. Given w learned by the methods in Karagulle et al. (2024a,b), let x^*, u^*, s^* be the solution of the following problem:

$$\begin{aligned} x^*, u^*, s^* = \operatorname*{arg\,min}_{\substack{x \in \mathcal{X}^T, u \in \mathcal{U}^T, \\ s \in \mathcal{S}^T}} \quad & -r(s, \phi_{\mathcal{W}=w}) \\ \text{s.t.} \quad & (1) \text{ holds for all } t \in [0, T] \end{aligned} \tag{3}$$

where T is the time horizon of the problem. If ϕ is strictly realizable, then the solution $s^* \models \phi$.

Proof. It follows from the safety guarantees in Karagulle et al. (2024a) and the soundness of WSTL.

Proposition 1 shows that the solution to Problem (3) guarantees a custom-and-correct-by-construction behavior when started from x_o and the control sequence u^* is applied. However, since this behavior only follows an objective based on s, the latent dimensions in x may result in unnaturalistic behavior. To address this, we incorporate driving demonstrations. Demonstrations and comparison signals enter the 4C block and output the desired behavior. The second step of the approach solves

$$\begin{aligned} x^*, u^*, s^* = \operatorname*{arg\,min}_{\substack{x \in \mathcal{X}^T, u \in \mathcal{U}^T, \\ s \in \mathcal{S}^T}} \quad & J(x, \bar{x}) - \lambda r(s, \phi_{\mathcal{W}=w^H}) \\ \text{s.t.} \quad & (1) \text{ holds for all } t \in [0, T] \\ & r(s, \phi) > 0. \end{aligned} \tag{4}$$

where $J(x, \bar{x})$ is the cost function related to demonstrations, and $\lambda \geq 0$ is a hyperparameter to tune the effect of preferences over the generated trajectory.

By constraining the weighted robustness as positive, we ensure that specifications are satisfied. We choose the following cost function: $J(x, \bar{x}) = \|x - \bar{x}\|_1$, where $\|x\|_1 = \sum_{i \in \{1,\dots,n\}} |x_i|$ is the L_1 norm and $\bar{x}$ is the most preferred demonstration, i.e., $\bar{x} \in \arg\max_{x \in \bar{\mathcal{X}}} r(l(x), \phi_{\mathcal{W}=w})$, where $\bar{\mathcal{X}}$ denotes the set of demonstrations.

As it can be deduced from Equation (2), the terms with WSTL specifications are not linear. However, it is possible to make them mixed-integer linear constraints by using `PyTeLo` developed by Cardona et al. (2023). Additionally, the L_1 norm can be written in linear constraints. Therefore, as long as the model is linear, Problem 3 is a MILP. In the case where the model is not linear, linearizing the $f(\cdot)$ with respect to a state helps obtain linear constraints. The time horizon of this problem is the full length of the generated trajectory. In other words, this problem can be seen as a trajectory generation problem rather than receding horizon control, as we do not have a time horizon to shift over the timeline. The generated trajectory can be used as a reference trajectory for the online control problem. When it is used as a trajectory to follow, we should consider potential shifts in the behavior due to model mismatch, and ensure the satisfaction of temporal logic formulas in the presence of this mismatch. Various works address this problem for different temporal logics (Deshmukh et al., 2017; Fainekos et al., 2009).

4. EXPERIMENTS

In this section, we present a synthetic analysis showing the effect of demonstrations and varying preferences on generating naturalistic behaviors. Following this, we will discuss the results of the pilot human subject study. Both experiments share key design elements, including the driving scenario, and the state-space model.

Driving scenario: We generate a simple scenario in which the ego vehicle approaches an intersection while a pedestrian crosses the street. The formula the ego vehicle should satisfy depends on the distance to the pedestrian d, the position of the ego vehicle $p = [p_x \; p_y]^T$, the acceleration of the vehicle a, and the jerk of the vehicle $\dot{a}$. The specification is $\phi = \phi_{rule} \wedge \phi_{destination} \wedge \phi_{comfort}$, where $\phi_{rule} = \Box(d \geq 2)$ represents the traffic rule for this scenario, that is the distance to the pedestrian should always be greater than 2 meters, $\phi_{destination} = \Diamond\Box_{[0,10]}(p \in \mathcal{P})$ represents the goal specification for the vehicle where $\mathcal{P}$ represents the goal region, and $\phi_{comfort} = \Box(a \leq 10 \wedge \dot{a} \leq 30)$ represents additional comfort specifications. Note that the comfort specifications are trivial to satisfy in daily driving situations (de Winkel et al., 2023).

Implementation details: We use the bicycle model for the ego vehicle. The simple diagram of variables related to bicycle kinematics can be found in Figure 2, and the details of the model can be found in (Polack et al., 2017).

The state variables are $x = [p_x \; p_y \; v \; \theta]^T$, where v is the speed and θ is the orientation of the vehicle. The control inputs are $u = [a \; \delta]^T$, where δ is the steering angle. The continuous kinematics of the center of mass of the vehicle are

$$f(x, u) = \dot{x} = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos(\theta + \beta) \\ v\sin(\theta + \beta) \\ a \\ v/L * \tan(\delta) * \cos(\beta) \end{bmatrix}, \tag{5}$$

where $\beta = \arctan(L_R \tan(\delta)/L)$ is the slip angle of the vehicle, L is the length of the vehicle from the rear wheel to the front wheel, and L_R is the distance from the rear wheel to the center of mass. The model is continuous-time

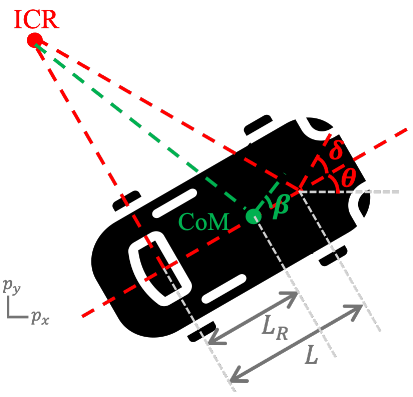

Fig. 2. The bicycle kinematics model. ICR represents the Instantaneous Center of Rotation and CoM represents the Center of Mass.

nonlinear, and we need to approximate it as linear-time discrete model to fit into Problem (3). The linearization point depends on the demonstration's initial state, as an example, with $x' = [0, 0, 0.5, 0]$, $u' = [0, 0]$, and $\Delta t = 0.1$, the discrete-time linear model becomes $x_{t+1} = A_d x_t + B_d u_t + f_d$ with matrices

$$A_d = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B_d = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.025 \\ 0.1 & 0 \\ 0 & 0.017 \end{bmatrix}, f_d = \begin{bmatrix} 0.05 \\ 0 \\ 0.5 \\ 0 \end{bmatrix}.$$

The details about the derivation of A_d, B_d and f_d can be found in Appendix A. We choose the time horizon T of the problem as the time length of the demonstration. For the cost function $J(x, \bar{\mathcal{X}}) = \|x - \bar{x}\|_1$, a weighted L_1 norm is used to scale the state elements in $J(x, \bar{\mathcal{X}}) = \|x - \bar{x}\|_{1,\alpha} = \sum_{i \in \{1,\ldots,4\}} \sum_{t \in T} \alpha_i |x_t^i - \bar{x}_t^i|$, where x_t^i represents the i^{th} entry of x_t. We define the weight multiplier as $\alpha = [0.2\ 0.4\ 0.05\ 0.5]$. The state and input domains are set as $x \in [-5, 140]\ m$, $y \in [1, 6.5]\ m$, $v \in [0, 30]\ m/s$, $\theta \in [-\pi, \pi]\ rad$, $a \in [-10, 15]\ m/s^2$, $\dot{a} \in [-30, 30]\ m/s^3$ and $\delta \in [-1, 1] rad$. Finally, for the safe preference learning, we use the set of existing demonstrations for the same scenario, generated by professional drivers using the simulator in Karagulle et al. (2024b).

4.1 Synthetic Experiments

In this section, we run our analysis on the effect of modalities in feedback with two different setups: the influence of preferences over the generated trajectory when (i) the demonstration satisfies ϕ, (ii) when the demonstration violates ϕ. These demonstrations are generated using CARLA (Dosovitskiy et al., 2017), following the physics of the CARLA vehicles. We adjust the λ value to modify the effect of the weighted robustness value on the cost function.

First, we examine the case (i) for four different instances with three different weight valuations. Instance (a) occurs when the cost function does not include the weighted robustness term, that is $\lambda = 0$, instance (b) and (c) is when $\lambda = 100$ and $\lambda = 10^3$, respectively. Instance (d) represents the case where the demonstration is excluded from the controller synthesis problem, meaning $J(x, \bar{x}) = 0$. Weight valuation w_1 represents the case where all weights are 1, corresponding to the traditional STL robustness. Weight valuations w_2 and w_3 are synthetically obtained through the safe preference learning module from Karagulle et al. (2024b). For w_2, we randomly pick a weight valuation w_H that is not among $M = 1000$ discretization points and assume it represents the weight valuation of a hypothetical person. This hypothetical person answers each question based on the order induced by the weighted robustness value. Specifically, when asked to compare s_i and s_j, they choose s_i if $r(s_i, \phi_{\mathcal{W}=w_H}) > r(s_j, \phi_{\mathcal{W}=w_H})$ and s_j otherwise. For w_3, the hypothetical person selects the answer with the smaller weighted robustness value, choosing s_i if $r(s_j, \phi_{\mathcal{W}=w_H}) > r(s_i, \phi_{\mathcal{W}=w_H})$ and vice versa.

Figure 3 shows results for case (i). In instance (a), we observe the same behavior for all weight valuations. Since the weighted robustness does not affect the cost, this observation is expected. The slight shift of the generated trajectories from the demonstration can be attributed to the differences in the vehicle models of the demonstration and the controller synthesis problem. As λ increases, we see more deviation from the demonstration to maximize the weighted robustness value. Notably, although the demonstration ends before the pedestrian, the trajectory generated with w_3 passes the intersection even before the pedestrian arrives. The flexibility in the specifications allows for such diversity in behaviors. In instance (d), where we exclude the demonstration from the controller synthesis, vehicles maximize their weighted robustness values by either slowing down or steering toward the edges of the lane to maximize their distance from the pedestrian. These behaviors are less likely to be seen on the road compared to ones in instances (b) and (c).

Second, we study case (ii) for the same instances and same weight valuations. The 4C block guarantees that the generated trajectories satisfy the formula even though the demonstration violates it. Figure 4 shows the violating demonstration and generated trajectories for all weight valuations when $\lambda = 10^3$. It is clear that at $t = 12$, the demonstration is violating ϕ_{rule}. Generated trajectories do not follow the same pattern and satisfy ϕ.

4.2 Pilot Human Subject Study

In this set of experiments, we conduct a pilot study with four participants to test if the generated trajectories reflect the preferences of the users. Our supplementary goal is to gather observational data and feedback from participants for future studies to be used in potential extensions. Therefore, the population size and the diversity of the participants are kept intentionally limited, and the results of this study are not representative of broad conclusions.

The simulator used in the study is shown in Figure 5. It consists of a wide monitor and a set of steering wheel and pedals, Logitech G920. The simulation environment is based on CARLA.

The study is conducted under HUM#00221976, and the protocol is as follows: (i) we record multiple demonstrations for the scenario, and allow participants to discard the ones they do not like. By the end of this step, we have a set of demonstrations $\bar{\mathcal{X}}$, that consists of at least one demonstration that the subject is satisfied with, (ii) we

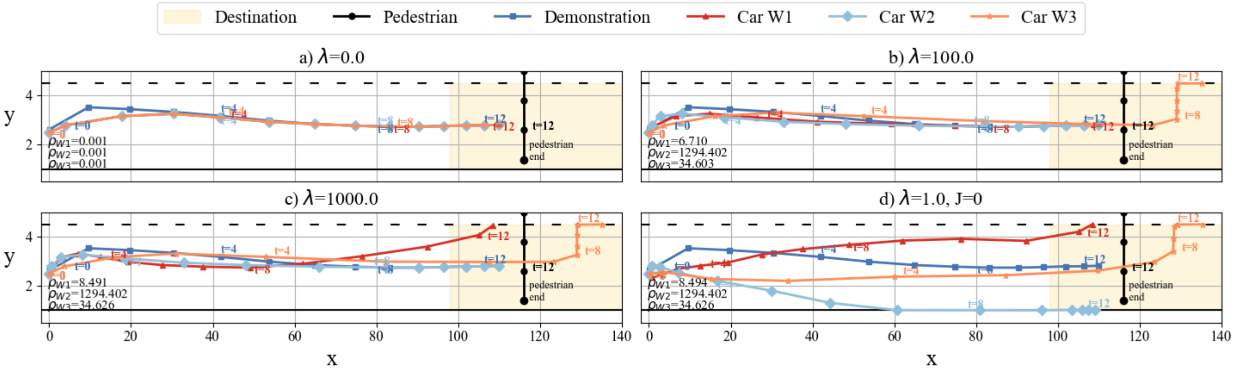

Fig. 3. Synthesised trajectories for case (i): the case where demonstration satisfies ϕ for different instances: The pedestrian is shown in black with circle markers, the demonstration is shown in blue with square markers. Generated trajectories with w_1, w_2, and w_3 are shown in red with triangle markers, pale blue with diamond markers, and orange with star markers, respectively. Markers denote positions at each second and we also every four seconds. The solid line represents the start of the sidewalk and the dashed line represents the lane limit.

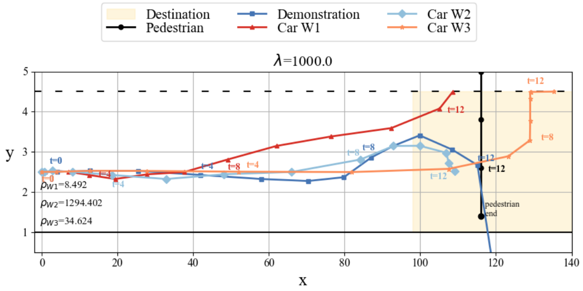

Fig. 4. Synthesised trajectories for case (ii): the case where demonstration violates ϕ for different instances.

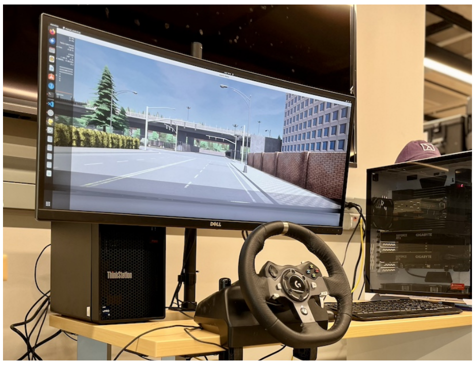

Fig. 5. The simulator used in the pilot human subject study. The throttle/brake pedals are on the floor.

show pairs of videos of different behaviors for the same scenario. We instruct them to imagine they are passengers in a vehicle driving as the shown behaviors and ask them to choose their preferred behavior. We continue to show new pairs until either the question budget is reached, which is 20, or the threshold maximum a-posteriori probability is achieved, which is 0.99. The output of this step is the learned weight valuation w^* for $\phi_{\mathcal{W}}$, (iii) we find their most preferred demonstration x^* based on the learned weight valuation, as $x^* = \arg\max_{x \in \bar{\mathcal{X}}} r(x, \phi_{\mathcal{W}=w^*})$, and generate two trajectories based on x^* and w^*: x_g, which is generated with $\lambda = 100$, and x_r, which is generated with $\lambda = 1$ with $J(x, \bar{x})$. (iv) The generated trajectories and their demonstration are replayed using a low-level PID controller, and we ask them to compare: (a) x_g to x_r and (b) x_g to x^*. Subjects are not informed which trajectory is their generated trajectories or their demonstrations.

Generated trajectories for subjects can be seen in Figure 6. Results of step (iv) show that three out of four subjects prefer x_g over x_r, as they find x_r unnatural, and three out of four subjects prefer their demonstration over x_g, as they note that the demonstration seems smoother. One potential reason for these preferences is the model mismatch and the performance of the low-level controller. The mismatch between the linear model and CARLA physics means the CARLA car cannot fully follow the generated trajectory, leading to a jerkier motion. Since the demonstrations are recorded using CARLA vehicles, they naturally follow those trajectories smoothly.

5. CONCLUSION, LIMITATIONS AND FUTURE WORK

This work addresses generating safe and natural AV behaviors that meet user preferences, by combining two modalities from human feedback: demonstrations and pairwise preferences. Pairwise preferences help construct personalized temporal logic semantics, which defines scenario-specific rules, to be used in the control synthesis step, and demonstrations include latent behaviors that are not described in the temporal logic formula. The safety guarantees of the personalized temporal logic semantics result in safe-and-custom-by-construction behaviors, while demonstrations add a layer of naturality. We conduct a set of synthetic experiments to show the effect of different feedbacks and also include a pilot study to show how human subjects perceive the generated trajectories. This work is limited in terms of the number of scenarios and number of subjects, and results show that the method is promising. Finally, preference learning with pairwise questions reduces reasonings behind choices into a boolean answer, which is useful from a computational perspective. On the other hand, reasoning is a rich source of information. As a next step, we plan to incorporate feedback in the form of natural language by leveraging large language models.

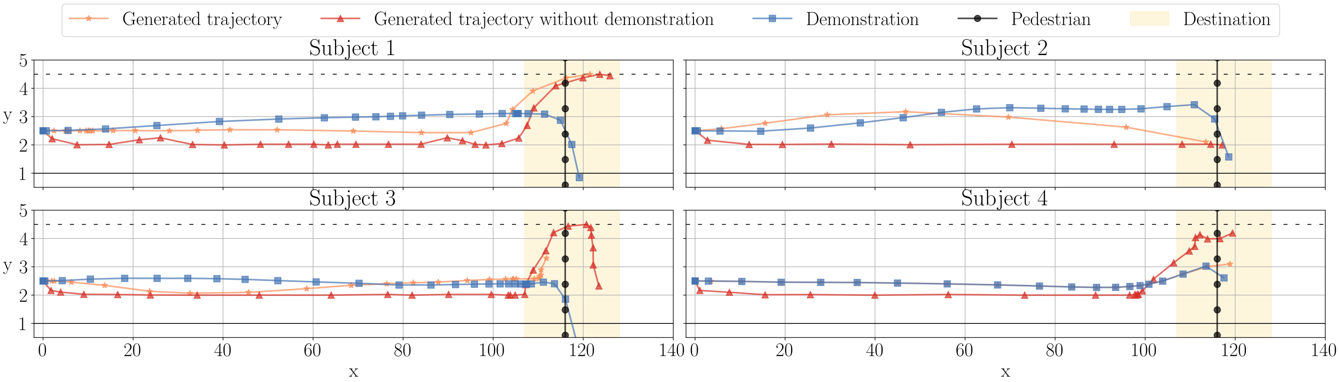

Fig. 6. Demonstrations and generated trajectories of human subjects in the study.

REFERENCES

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

Cardona, G.A., Leahy, K., Mann, M., and Vasile, C.I. (2023). A flexible and efficient temporal logic tool for python: Pytelo.

De Giacomo, G. and Vardi, M.Y. (2013). Linear temporal logic and linear dynamic logic on finite traces. In *Proceedings of the 23 Intl. Joint Conf. on AI*, 854–860. AAAI Press.

de Winkel, K.N., Irmak, T., Happee, R., and Shyrokau, B. (2023). Standards for passenger comfort in automated vehicles: Acceleration and jerk. *Applied Ergonomics*, 106.

Deshmukh, J.V., Donzé, A., Ghosh, S., Jin, X., Juniwal, G., and Seshia, S.A. (2017). Robust online monitoring of signal temporal logic. *Formal Methods in System Design*, 51, 5–30.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 1–16.

Fainekos, G.E., Girard, A., Kress-Gazit, H., and Pappas, G.J. (2009). Temporal logic motion planning for dynamic robots. *Automatica*, 45(2), 343–352.

Fürnkranz, J. and Hüllermeier, E. (2003). Pairwise preference learning and ranking. In *Machine Learning: ECML 2003*, 145–156. Springer Berlin Heidelberg.

Hartwich, F., Beggiato, M., and Krems, J.F. (2018). Driving comfort, enjoyment and acceptance of automated driving – effects of drivers' age and driving style familiarity. *Ergonomics*, 61(8), 1017–1032.

Haselberger, J., Böhle, M., Schick, B., and Müller, S. (2024). Exploring the influence of driving context on lateral driving style preferences: A simulator-based study. *arXiv preprint arXiv:2402.14432*.

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

Huang, Z., Wu, J., and Lv, C. (2022). Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 10239–10251.

Karagulle, R., Aréchiga, N., Best, A., DeCastro, J., and Ozay, N. (2024a). A safe preference learning approach for personalization with applications to autonomous vehicles. *IEEE Robotics and Automation Letters*, 9(5), 4226–4233. doi:10.1109/LRA.2024.3375626.

Karagulle, R., Ozay, N., Aréchiga, N., DeCastro, J., and Best, A. (2024b). Incorporating logic in online preference learning for safe personalization of autonomous vehicles. In *Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '24. Association for Computing Machinery, New York, NY, USA.

Kuderer, M., Gulati, S., and Burgard, W. (2015). Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2641–2646.

Lefèvre, S., Carvalho, A., and Borrelli, F. (2016). A learning-based framework for velocity control in autonomous driving. *IEEE Trans. on Automation Science and Engineering*, 13(1), 32–42.

Mehdipour, N., Vasile, C.I., and Belta, C. (2021). Specifying user preferences using weighted signal temporal logic. *IEEE Control Systems Letters*, 5(6), 2006–2011.

Polack, P., Altché, F., d'Andréa Novel, B., and de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 812–818.

Sadraddini, S. and Belta, C. (2015). Robust temporal logic model predictive control. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing*, 772–779.

Saha, S. and Julius, A.A. (2016). An milp approach for real-time optimal controller synthesis with metric temporal logic specifications. In *2016 American Control Conf. (ACC)*, 1105–1110.

Schrum, M.L., Sumner, E., Gombolay, M.C., and Best, A. (2024). Maveric: A data-driven approach to personalized autonomous driving. *IEEE Trans. on Robotics*, 40, 1952–1965.

Veliov, V. (1997). On the time-discretization of control systems. *SIAM Journal on Control and Optimization*, 35(5), 1470–1486.

Wolff, E.M., Topcu, U., and Murray, R.M. (2013). Automaton-guided controller synthesis for nonlinear systems with temporal logic. In *2013 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 4332–4339.

Appendix A. LINEARIZATION AND DISCRETIZATION OF BICYCLE MODEL

The local linear discretization around a desired state-input pair (x', u') is obtained by performing a first-order Taylor expansion

$$f(x,u) \simeq f(x',u') + A(x-x') + B(u-u'),$$

where $A = \frac{\partial f(x,u)}{\partial x}|_{(x',u')}$ and $B = \frac{\partial f(x,u)}{\partial u}\;|_{(x',u')}$. Now that we have the linearized model, we discretize it assuming A and B matrices are approximately constant for a discrete-time instant Δt. Then, discretized matrices can be computed by $A_d = \exp(A\Delta t)$, $B_d = A^{-1}(A_d - I)B$,$f_d = \int_0^{\Delta t} e^{A\tau} f(x',u') d\tau$, and we obtain

$$A_d = \begin{bmatrix} 1 & 0 & a_{13} & a_{14} \\ 0 & 1 & a_{23} & a_{24} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & a_{43} & 1 \end{bmatrix}, B_d = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & 0 \\ b_{41} & b_{42} \end{bmatrix},$$

where

$$\begin{aligned}
a_{13} &= \Delta t \cos(\theta+\beta) - \frac{\Delta t^2 v \sin(\beta)}{2L_R} \sin(\theta+\beta), \\
a_{14} &= -\Delta t v \sin(\theta+\beta), \\
a_{23} &= \Delta t \sin(\theta+\beta) + \frac{\Delta t^2 v \sin(\beta)}{2L_R} \cos(\theta+\beta), \\
a_{24} &= \Delta t v \cos(\theta+\beta), \\
a_{43} &= \frac{\Delta t \sin(\beta)}{L_R}, \\
b_{11} &= \frac{\Delta t^2}{2} \cos(\theta+\beta) - \frac{\Delta t^3 v \sin(\beta)}{6L_R} \sin(\theta+\beta), \\
b_{12} &= -\frac{\Delta t v \cos^2(\beta)(2L_R + \Delta t v \cos(\beta))}{2L \cos^2(\delta)} \sin(\theta+\beta), \cdot \\
b_{21} &= \frac{\Delta t^2}{2} \sin(\theta+\beta) + \frac{\Delta t^3 v \sin(\beta)}{6L_R} \cos(\theta+\beta), \\
b_{22} &= \frac{\Delta t v \cos^2(\beta)(2L_R + \Delta t v \cos(\beta))}{2L \cos^2(\delta)} \cos(\theta+\beta), \\
b_{31} &= \Delta t, \\
b_{41} &= \frac{\Delta t^2 \sin(\beta)}{2L_R}, \\
b_{42} &= \frac{\Delta t v \cos^3(\beta)}{L \cos^2(\delta)},
\end{aligned}$$

The f_d term is integrated by applying 4^{th} order Runge-Kutta (Veliov, 1997).