Ensuring Safety of Learning-Based Motion Planners Using Control Barrier Functions

Xiao Wang

***Abstract*—Reinforcement learning (RL) has been successfully applied to sequential decision-making problems, e.g., playing computer games or solving robotic tasks in simulations. However, RL methods are not yet ready to be applied to real robotic systems if safety is a major concern. To address this issue, we propose a safety layer based on control barrier functions to ensure the safety for an RL-based motion planner for highway scenarios with a continuous action space. Our method ensures legal safety by following traffic rules. Moreover, we propose a relaxation mechanism so that safety is restored as soon as possible when other vehicles violate traffic rules and render our optimization problem infeasible. We evaluate our approach using a real-world highway dataset and a traffic simulator. Numerical experiments confirm that an agent equipped with our proposed safety layer does not cause any accidents during learning and yet reaches the goal as often as an agent without a safety layer.**

***Index Terms*—Intelligent transportation systems, reinforcement learning, robot safety.**

I. Introduction

REINFORCEMENT learning (RL) offers promising solutions for real-world problems, especially sequential decision-making tasks in robotics, such as motion planning for autonomous vehicles [1], [2] or controlling robot manipulators [3], [4]. However, since RL methods aim to learn an optimal policy through interaction with the environment, unsafe actions are likely to be taken, especially during the initial learning phase. Even for a trained agent, safety is often not verified for deep RL models due to their black-box property. To apply RL methods to real autonomous systems, safety has to be guaranteed during training and deployment.

Our previous work [5] has proposed a safe RL framework ensuring the safety of a high-level motion planner for autonomous lane changing on highways. However, a low-level trajectory planner is additionally required for our previous approach. Moreover, the solution space is limited by the definition of the high-level action space. In this work, we aim to develop an RL-based planner that directly calculates safe control inputs for an autonomous vehicle from its current state. Although there have been studies on ensuring the safety of low-level control inputs of a car-following agent [6]–[11], there is no existing method for guaranteeing the safety of RL-based low-level motion planners in general highway scenarios. Figure. 1 shows our framework.

Manuscript received: September 9, 2021; Revised: December 7, 2021; Accepted: January 3, 2022. This paper was recommended for publication by Editor Youngjin Choi upon evaluation of the Associate Editor and Reviewers' comments.

Xiao Wang is with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany. xiao.wang@tum.de

Digital Object Identifier (DOI): see top of this page.

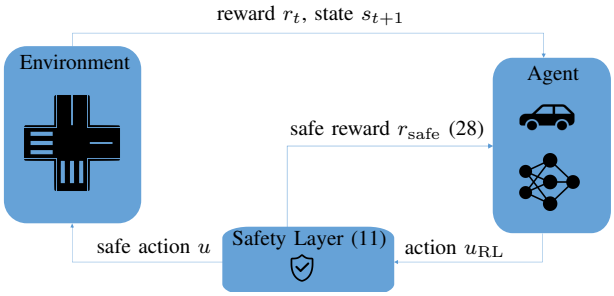

Fig. 1. Overview of the proposed approach. Our safety layer utilizes control barrier functions to correct the continuous actions of an RL agent u_{RL} to safe actions in a minimally invasive fashion (see II-C). In addition, our safety layer provides a reward term to improve the learning process (see IV-B).

A. Related Work

In the following literature review, we focus on existing work on safe RL methods and control barrier functions with an emphasis on motion planning for autonomous vehicles.

1) Safe reinforcement learning: García et al. [12] classified safe RL approaches into two main categories: by modification of the optimization criterion and by verification of the exploration processes. By modifying the optimization criterion, such as *the worst-case criterion* [13], [14] or *the risk-sensitive criterion* [15]–[19], as well as using *constrained policy optimization* methods [20], [21], the absence of unsafe actions is not guaranteed, even if risky behaviors are punished severely.

In contrast, safe actions can always be taken by verifying the exploration process using external guidance, e.g., using an additional safety layer to verify the safety of the proposed actions and correct unsafe actions using a fail-safe controller [5]. Therefore, the remainder of this literature review focuses on methods using a verification layer. One approach is to *provide initial knowledge*, i.e., initialize the agent using a verified safe policy and update the agent only if safety constraints are fulfilled [22], [23]. However, an initial verified safe policy for the task of autonomous driving is not always available. Another approach is to provide a *shield* for the agent [24], also called safety layer in [5] and safety filter in [25]. Alshiekh et al. [24] distinguish the shielding approaches between *preemptive shielding*, which removes unsafe actions from the action space before the learning agent [5], [26]–[29], and *post-posed shielding*, which corrects an unsafe action proposed by the agent [10], [25], [30], [31]. The preemptive shielding method is more suitable for problems with a discrete action space, since computing the regions of safe action spaces and incorporating them into the agent model can be intractable.

Therefore, we utilize the post-posed shielding framework in this paper.

Among the above-mentioned papers, the closest to ours is [10], since it also presents an approach to ensure the safety of an RL-based controller for continuous control tasks based on control barrier functions. However, it is assumed in [10] that the safe set is already given and has only one constraint. Moreover, the authors of [10] only demonstrated their approach for an inverted pendulum problem as well as a simple car-following scenario where the acceleration of other vehicles is known and Gaussian noise is added. In this work, we will show in detail how to define safe sets with multiple constraints for highway scenarios as well as how to relax the optimization problem when the problem becomes infeasible.

2) Control barrier functions: Control barrier functions originated from the set invariance theory of Nagumo [32] and were proven and reformulated into a modern version after many years of research [33]. The core idea of barrier functions is to guarantee the forward invariance of a set, which can be chosen to also be a safe set so that safety is ensured for infinite time.

In the field of vehicle control, barrier functions have been utilized to solve adaptive cruise control and lane-keeping problems [6]–[10], which are formulated as quadratic programming problems. The objectives of the optimization problems contain soft constraints, such as holding the desired speed, whereas the hard constraints of the optimization problems are safety requirements, such as keeping a safe distance to other vehicles. Although the approaches in [6]–[10] have shown good performance in their experimental setups, they are based on simplified models, simplified assumptions, or simplified safety constraints, thus they cannot be applied to general highway scenarios directly.

More recent works have extended the application of control barrier functions to more general scenarios [34]–[37]. Choi et al. [34] used RL to learn the model uncertainty in the control barrier function and control Lyapunov function constraints, validated on a bipedal walking robot example. Chen et al. [35] combined imitation learning and control barrier functions to learn driving from recorded data with enhanced safety for generic urban scenarios. Notomista et al. [36] utilized control barrier functions to enhance safety for a game-theoretic approach for autonomous car racing. Zeng et al. [37] improved the previous works by formulating the car racing problem in a Frenet coordinate system. However, the above-mentioned works have defined rather simple safety constraints, i.e., based on relative position and velocity, which are only collision-free during the planning horizon and thus not invariably safe[1], which could result in an accident beyond the planning horizon. Instead, we provide a more comprehensive definition of provably safe constraints based on invariably safe sets [38] ensuring safety and feasibility under legal assumptions for an infinite time horizon.

[1]For the difference between collision-free states and invariably safe states, please refer to Fig.1 in [38].

B. Contributions

We develop a safe RL approach for highway motion planning based on control barrier functions [33]. Our main contributions are:

1) Our method guarantees safety for arbitrary low-level motion planners for highway driving while minimizing the interference of the safety layer.
2) We combine an RL-based motion planner with the proposed safety layer.
3) We improve the learning efficiency of the RL agent by providing a reward from the safety layer as feedback.
4) We evaluate the proposed approach using a real-world highway dataset and interactive scenarios in a traffic simulator.

The remainder of this paper is organized as follows: Section II presents the required preliminaries. Subsequently, Section III introduces the safety specifications and the required constraints including their relaxation when the optimization problem becomes infeasible due to the illegal behaviors of other vehicles. The proposed method is then evaluated in real-world and simulated highway scenarios in Sec. IV. Finally, we draw our conclusions in Sec. V.

II. Preliminaries

In this section, we introduce our vehicle model, control barrier functions, and our optimization problem formulation.

A. Vehicle Kinematics in a Frenet Frame

We use the commonly used kinematic single-track model, which assumes that the two wheels of the front and rear axle lumped together into a single wheel located at the center of each axle. The advantage of this model is its simplicity while still considering the non-holonomic vehicle behavior.

In this work, we formulate the safety constraints with regard to a Frenet frame aligned with the reference path Γ of orientation $\psi_\Gamma(s)$ (see Fig. 2). The configuration of the vehicle is described by the longitudinal position s, the lateral deviation of the reference path d, and the relative orientation e_ψ. Furthermore, let us introduce the velocity v, the orientation ψ, the acceleration in the longitudinal direction a_{long}, and the curvature along the reference path $\kappa(s)$. The differential equations of the kinematic single-track model in the Frenet frame are [39]

$$\begin{aligned} \dot{s} &= \frac{v\cos e_\psi}{1-\kappa(s)d}, \\ \dot{d} &= v\sin e_\psi, \\ \dot{e}_\psi &= \dot{\psi} - \frac{v\cos e_\psi \cdot \kappa(s)}{1-\kappa(s)d}, \\ \dot{v} &= a_{\text{long}} \end{aligned} \tag{1}$$

Furthermore, assuming the maximum absolute acceleration $a_{\max}$, we consider the friction circle as a constraint limiting absolute acceleration [39]

$$\sqrt{a_{\text{long}}^2 + (v\,\dot{\psi})^2} \leq a_{\max} \tag{2}$$

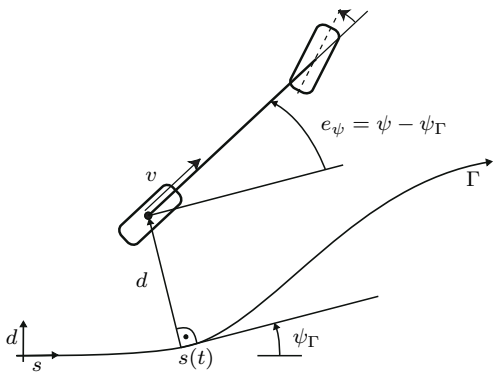

Fig. 2. Kinematic single-track model in a Frenet frame. We assume that Γ corresponds to the centerline of a lane occupied by the ego vehicle.

After introducing the state variables x_i as

$$x_1 = s, x_2 = d, x_3 = e_\psi, x_4 = v$$

and the input variables u_i as

$$u_1 = \dot{\psi}, u_2 = a_{\text{long}} \tag{3}$$

the kinematic single-track model in the Frenet frame can be written in state-space form:

$$\dot{x} = \underbrace{\begin{bmatrix} \frac{x_4 \cos x_3}{1-\kappa(x_1)\,x_2} \\ x_4 \sin x_3 \\ -\frac{x_4 \cos x_3\, \kappa(x_1)}{1-\kappa(x_1)\,x_2} \\ 0 \end{bmatrix}}_{f(x)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{g(x)} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{u} \tag{4}$$

B. Control Barrier Functions

Definition. (Control Barrier Function). Let us define the safe set $\mathcal{C}$ as the superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, i.e., $\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}$. Given a nonlinear control-affine system

$$\dot{x} = f(x) + g(x)\,u, \tag{5}$$

where f and g are locally Lipschitz continuous, $x \in X \subset \mathbb{R}^n$ is the set of admissible states and $u \in U \subset \mathbb{R}^m$ is the set of admissible inputs, h is a control barrier function if there exists an extended class $\mathcal{K}$ function α [40] such that:

$$\dot{h}(x,u) \geq -\alpha(h(x)) \quad \Leftrightarrow \quad \mathcal{C} \text{ is invariant.} \tag{6}$$

A simple form of an extended class $\mathcal{K}$ function is $\alpha(h(x)) = \gamma\, h(x)$, $\gamma > 0$, which we use in this paper. We introduce how to choose the value of γ in Sec. III-C. To prove the necessity and sufficiency of condition (6), interested readers are referred to [7]. To guarantee invariance of safety, we need to define a safe set $\mathcal{C}$ and find the set of control values that render $\mathcal{C}$ safe, i.e., if the system starts inside the safe set $\mathcal{C}$, it will never leave $\mathcal{C}$.

Let $*$ be a variable; we denote all variables related to the ego vehicle by $*_{\text{ego}}$ and all variables related to other vehicles by $*_{\text{obs}}$. Specifically, the state variables of all other vehicles are stacked together and are denoted by x_{obs}. For the task of safe motion planning on highways, we consider static constraints specified by $h(x_{\text{ego}})$, which only depend on the ego vehicle, as well as dynamic constraints, which also depend on other vehicles (see Sec. III-B) and thus can be separated into

$$h(x) = h(x_{\text{ego}}) + h(x_{\text{obs}}) \tag{7}$$

To obtain safety constraints on the inputs of the ego vehicle u_{ego}, we need to rewrite (6):

$$\begin{aligned} \dot{h}(x_{\text{ego}}) = \frac{\partial h}{\partial x_{\text{ego}}}\,\dot{x}_{\text{ego}} \overset{(5)}{=} \frac{\partial h}{\partial x_{\text{ego}}}(f(x_{\text{ego}}) + g(x_{\text{ego}})\,u_{\text{ego}}) \\ := L_f h(x_{\text{ego}}) + L_g h(x_{\text{ego}})\,u_{\text{ego}}, \end{aligned} \tag{8}$$

where $L_f h(x) = \frac{\partial h}{\partial x}\, f(x) \in \mathbb{R}^n$, $L_g h(x) = \frac{\partial h}{\partial x}\, g(x) \in \mathbb{R}^{n \times m}$. Combining (6), (7), and (8), we have

$$-L_g h(x_{\text{ego}})\,u_{\text{ego}} \leq \gamma h(x) + L_f h(x_{\text{ego}}) + \dot{h}(x_{\text{obs}}) \tag{9}$$

We calculate $\dot{h}(x_{\text{obs}})$ at each time step using worst-case assumptions.

In this work, we consider $\mathcal{C}$ as the intersection of half-spaces defined by k affine barrier functions with the index i. Stacking together all constraints in (9), the affine constraint on u then becomes

$$A\,u \leq b, \tag{10}$$

where

$$\begin{aligned} A &= [-L_g h_1(x), -L_g h_2(x), \cdots, -L_g h_k(x)]^\mathsf{T} \in \mathbb{R}^{k \cdot n \times m} \\ b &= [b_1, b_2, \cdots, b_k] \in \mathbb{R}^{k \cdot n}, \text{with} \\ b_i &= \begin{cases} \gamma_i\, h_i(x) + L_f h_i(x_{\text{ego}}), & \text{for static constraints} \\ \gamma_i\, h_i(x) + L_f h_i(x_{\text{ego}}) + \dot{h}(x_{\text{obs}}), & \\ & \text{for dynamic constraints.} \end{cases} \end{aligned}$$

C. Optimization-Based Minimally Invasive Control

Our goal is to modify the action proposed by the RL agent u_{RL} as little as possible to ensure safety. Considering the affine constraints on u in (10), we can formulate the optimization problem as [33]

$$\begin{aligned} u(x) = \underset{u \in \mathbb{R}^m}{\arg\min} \quad & \frac{1}{2}\,\|u - u_{\text{RL}}\|^2 \\ \text{s.t.} \quad & A\,u \leq b, \end{aligned} \tag{11}$$

which can be solved by quadratic programming [41].

III. Safe Highway Motion Planning Using Control Barrier Functions

To formulate safety constraints properly, we first introduce the safety specifications considered in this work to guarantee legal safety. Moreover, when other vehicles violate traffic rules and render our optimization problem infeasible, e.g., due to illegal cut-ins, we relax the original optimization problem while restoring safety as soon as possible.

A. Specifications

We consider the following specifications which are conformant with traffic laws [11], [27]:

1) The ego vehicle has to keep a safe distance to its leading vehicle; when another vehicle cuts-in in front of the ego vehicle, the ego vehicle has to recover the safe distance in a timely manner.

2) The ego vehicle is allowed to change its current lane only when its longitudinal distance to the leading and following vehicles in the target lane is larger than the safe distance.

3) If safety is harmed due to the illegal behavior of other vehicles, the ego vehicle should try to restore safety as soon as possible[2].

Besides traffic laws, we consider an additional specification for approaching following vehicles:

4) The ego vehicle has to keep a safe distance to its following vehicle whenever feasible; when no feasible solution can be obtained, the ego vehicle is allowed to violate the safe distance to its following vehicle but should not collide with it.

In addition, the ego vehicle should satisfy the speed limit and control limit constraints.

B. Safe Sets

We define the safety constraints based on the invariably safe sets developed in our previous work [38] as the control barrier functions.

1) Longitudinal Constraints: We formulate longitudinal constraints according to specifications 1 and 4. For the leading vehicle of the ego vehicle, we define

$$h_1 = \Delta s - s_{\text{safe}}, \tag{12}$$

where Δs and s_{safe} are the longitudinal distance and safe distance between the leading vehicle and the ego vehicle, respectively. Let $*_{\text{lead}}$ and $*_{\text{f}}$ denote the states of a leading and following vehicle, respectively; and let us define

$$s_{\text{brake}} = \frac{v_{\text{lead}}^2}{-2a_{\text{max,lead}}} - \frac{v_{\text{f}}^2}{-2a_{\text{max,f}}} + \delta v_{\text{f}}, \tag{13}$$

where δ is the reaction time of the following vehicle, then the safe distance can be calculated as [42]

$$s_{\text{safe}} = \max(s_{\text{brake}}, 0) \tag{14}$$

For simplification, we assume $\delta = 0$ and $a_{\text{max,lead}} = a_{\text{max,f}}$ in this work. Since a control barrier function has to be continuously differentiable, we have to eliminate the $\max$ operator in (14). We use $s_{\text{safe}} = s_{\text{brake}}$ in (12), since $v_{\text{lead}} > v_{\text{f}}$ if $s_{\text{safe}} < 0$ and thus Δs will become larger and safer at the next time step, no matter what actions the ego vehicle executes. Therefore, we obtain

$$h_1 = \Delta s - s_{\text{brake}} \overset{(13)}{=} \tag{15}$$
$$\underbrace{s_{\text{lead}} - l_{\text{lead,rear}} + \frac{v_{\text{lead}}^2}{2a_{\text{max,lead}}}}_{h_1(x_{\text{lead}})} \underbrace{-x_1 - l_{\text{ego,front}} - \frac{x_4^2}{2a_{\text{max,ego}}}}_{h_1(x_{\text{ego}})},$$

where $l_{*,\text{front}}$ and $l_{*,\text{rear}}$ denote the length between the reference point and the front/rear end of a vehicle.

For the constraint to a following vehicle, we define two versions of control barrier functions: one is invariably safe, namely considering the safe distance, which ensures legal safety for infinite time; when the optimization problem with the invariably safe constraint becomes infeasible, we relax this constraint to emergency mode. The invariably safe version can be obtained similarly to (15):

$$h_{2,\text{IS}} = \underbrace{x_1 - l_{\text{ego,rear}} + \frac{x_4^2}{2a_{\text{max,ego}}}}_{h_2(x_{\text{ego}})} \underbrace{-s_{\text{f}} - l_{\text{f,front}} - \frac{v_{\text{f}}^2}{2a_{\text{max,f}}}}_{h_2(x_{\text{f}})} \tag{16}$$

To prevent collision within the planning horizon when legal assumptions are violated by other vehicles, the emergency version is defined as

$$h_{2,\text{CF}}^0 = \Delta s = x_1 - l_{\text{ego,rear}} - s_{\text{f}} - l_{\text{f,front}} \tag{17}$$

Since the input relative degree of (17) is two, i.e., we need to derive (17) twice to obtain inputs, we convert (17) using high-order control barrier functions [43] to

$$\begin{aligned} h_{2,\text{CF}}^1 &= \dot{h}_{2,\text{CF}}^0 + \gamma\, h_{2,\text{CF}}^0 \geq 0 \\ &= (x_4 \cos x_3 - v_{\text{f}} \cos e_{\psi,\text{f}}) + \\ &\quad \gamma\,(x_1 - l_{\text{ego,rear}} - s_{\text{f}} - l_{\text{f,front}}) \geq 0 \end{aligned} \tag{18}$$

Note that γ here is the coefficient of the zero-order control barrier function. We show the effect of γ in Sec. III-C1.

2) Lateral Constraints: We consider two kinds of lateral constraints: one dynamic constraint for other vehicles merging illegally from other lanes into the current lane of the ego vehicle; and one static constraint for lane or road boundaries. The dynamic lateral constraints can be obtained similarly to (18) as (see Appendix)

$$\begin{aligned} h_3 &= (v_{\text{obs}} \sin e_{\psi,\text{obs}} - v_{\text{ego}} \sin e_{\psi,\text{ego}}) + \gamma\,(d_{\text{obs}} - d_{\text{ego}}), \\ &\qquad \text{for vehicles from the left,} \\ h_4 &= (v_{\text{ego}} \sin e_{\psi,\text{ego}} - v_{\text{obs}} \sin e_{\psi,\text{obs}}) + \gamma\,(d_{\text{ego}} - d_{\text{obs}}), \\ &\qquad \text{for vehicles from the right.} \end{aligned} \tag{19}$$

Note that it is not trivial to obtain a worst-case behavior for lateral dynamics, since it is non-monotonic [44]. For simplicity, we use $\psi_{\text{obs}} = \psi_{\max}$ and $a_{\text{long,obs}} = 0$ as the control inputs of other vehicles of interest. Although this bound is not formally correct, our method still ensures legal safety since constraints (19) are only considered for illegally merging obstacles. To obtain a more precise bound of the lateral dynamics, reachability analysis can be applied [45].

For the static lateral constraints, we consider specification 2, namely if the ego vehicle is legally allowed to change to

[2]We show exemplary scenarios where other vehicles violate traffic rules and how agents equipped with our safety layer response in the video attachment.

adjacent lanes, we restrict the lateral distance between the ego vehicle and road boundaries; otherwise, we restrict the lateral distance between the ego vehicle and lane boundaries. We denote this distance by d_{boundary}. We focus on highway scenarios where d_{boundary} is constant. Therefore, we obtain the static lateral constraints using high-order control barrier functions as

$$\begin{aligned} h_5 &= -v_{\text{ego}} \sin e_{\psi,\text{ego}} + \gamma\,(d_{\text{boundary}} - d_{\text{ego}}), \\ &\qquad \text{for the left road/lane boundary,} \\ h_6 &= v_{\text{ego}} \sin e_{\psi,\text{ego}} + \gamma\,(d_{\text{ego}} - d_{\text{boundary}}), \\ &\qquad \text{for the right road/lane boundary.} \end{aligned} \tag{20}$$

3) Speed Limit Constraints: We restrict the velocity of the ego vehicle according to the speed limit of the current lane. Furthermore, the ego vehicle is not allowed to drive backward on a highway. Therefore, we have the following constraints:

$$\begin{aligned} h_7 &= v_{\text{speed\_limit}} - v \\ h_8 &= v \end{aligned} \tag{21}$$

4) Control Limit Constraints: We obtain N control limit constraints by under-approximating the friction circle (2) and linearizing it into N segments. Here, we choose $N = 16$ empirically. We denote the coefficients obtained in the linearization by $a_\mu, b_\mu \in \mathbb{R}^N$ and the range of the indices of the constraints by $9 : 9 + N$. Then the corresponding control barrier functions can be formulated as

$$h_{9:9+N} = a_\mu\, v_{\text{ego}}\, u_1 + u_2 - b_\mu \tag{22}$$

C. Relaxation of Constraints

Since the constraints in (11) depend on the value of γ, the optimization problem could become infeasible if γ is not chosen properly. Therefore, before introducing a relaxation method when (11) becomes infeasible, we first discuss the effect of γ to derive its bounds.

1) Effect of γ: We show the effect of γ on safety and feasibility of the optimization problem (11) in two cases[3] and derive its bounds:

- When the state of the agent is within $\mathcal{C}$, we allow the agent to move in any direction. However, the step size of the agent should be restricted such that the agent still stays within $\mathcal{C}$ at the next time step, i.e.,

$$\begin{aligned} &h(x(t+1)) \approx h(x(t)) + \dot{h}(x(t))\Delta t \geq 0 \implies \\ &\dot{h}(x) \geq -\gamma h(x) \geq -\frac{h(x)}{\Delta t} \implies \\ &\qquad 0 < \gamma \leq \frac{1}{\Delta t}, \text{ for } h(x) > 0. \end{aligned} \tag{23}$$

  The smaller γ is, the safer the agent stays, but the constraints in (11) become stricter, which could render (11) infeasible. We choose $\gamma = 3$ for $h > 0$ as the default value.

- When the state of the agent is outside or at the boundary of the safe set $\mathcal{C}$, the agent should only move towards $\mathcal{C}$, i.e.,

$$\dot{h}(x) \geq -\gamma h(x) > 0 \implies \gamma > 0, \text{ for } h(x) \leq 0. \tag{24}$$

  The larger γ is, the further the agent moves towards $\mathcal{C}$, thus the safer the agent stays, but the constraints become stricter, which could render (11) infeasible. We choose $\gamma = \frac{1}{\Delta t}$ for $h \leq 0$ as the default value, such that $h(x(t+1)) \geq 0$.

To summarize, the choice of the value of γ is a tradeoff between safety and the feasibility of (11).

2) Relaxation of the Optimization Problem: When the original optimization problem (11) with default γ becomes infeasible, i.e., when the intersection of constraints (10) is an empty set, we relax the constraint to the following vehicle from invariably safe mode (see (16)) to emergency mode (see (18)). Moreover, we observe that in all cases, infeasibility is caused by constraints with $h \leq 0$ since its γ has a much larger default value ($\frac{1}{\Delta t}$), which means that it is infeasible for the agent to come back to safe sets within one time step. We aim to relax constraints with $h \leq 0$ (i.e., decrease γ) while minimizing the time steps for the agent to come back to safe sets (i.e., maximizing γ within $\frac{1}{\Delta t}$). We introduce a new variable $y := \frac{1}{\Delta t} - \gamma$ and convert (11) and (9) into

$$\begin{aligned} &\underset{u\in\mathbb{R}^m, y}{\arg\min} \quad \frac{1}{2}\left\|u - u_{\text{RL}}\right\|^2 + \frac{1}{2}y^2, \\ &\text{s.t.} \qquad\qquad y \leq \frac{1}{\Delta t}, \\ &-L_g h(x_{\text{ego}}) u_{\text{ego}} + h(x) y \leq \frac{h(x)}{\Delta t} + L_f h(x_{\text{ego}}) + \dot{h}(x_{\text{obs}}) \end{aligned} \tag{25}$$

IV. Evaluation

A. Simulation Environment

1) Dataset: We evaluate the proposed approach on the real-world highway drone dataset (highD) of naturalistic vehicle trajectories recorded at six locations with two-lane or three-lane roads on German highways [46]. The highD dataset contains $16.5\,\text{h}$ (over $45\,000\,\text{km}$) of vehicle trajectories with a time size of $\Delta t = 0.04\,\text{s}$. We convert the dataset into 3000 scenarios with a duration of $40\,\text{s}$ for each scenario. For each scenario, we randomly choose a vehicle, create a planning problem using its initial and final states, and remove this vehicle from the scenario. Furthermore, we randomly split the scenarios into 70% training set and 30% test set.

2) Training Settings: We build the training environment on top of CommonRoad-RL [47]. The state space definition is adopted from [5] and we add five additional features as listed in Tab. I. The action of the agent is the control inputs of the vehicle model (3). We terminate the episode when one of the following binary variables becomes true:

- $\mathbf{1}_{\text{reach\_goal}} = 1$ if the ego vehicle reaches the goal area.
- $\mathbf{1}_{\text{collision}} = 1$ if the ego vehicle collides with others.
- $\mathbf{1}_{\text{off\_road}} = 1$ if the ego vehicle drives offroad.
- $\mathbf{1}_{\text{time\_out}} = 1$ if the duration of the scenario is reached.

In addition, we define $\mathbf{1}_{\text{safe\_dist}} = 1$ if the safe distance between the ego vehicle and its leading vehicle is violated.

[3]Note that we assume $\dot{h}(x(t))$ stays constant for one time step. To obtain more precise bounds of γ assuming time-variant $\dot{h}(x(t))$, reachability analysis can be applied [45] in future work.

TABLE I
ADDITIONAL FEATURES IN THE STATE SPACE COMPARED TO [5]

Dim.	State	Description
1-16	-	Same as Tab. I in [5]
17	ψ_{ego}	Orientation of the ego vehicle
18	$d_{\text{left\_lane}}$	Lateral distance from ego vehicle to left lane
19	$d_{\text{right\_lane}}$	Lateral distance from ego vehicle to right lane
20	$d_{\text{left\_road}}$	Lateral distance from ego vehicle to left of road
21	$d_{\text{right\_road}}$	Lateral distance from ego vehicle to right of road

With the help of these binary variables, we define a reward function as follows:

$$r_{\text{RL}} = r_{\text{reach\_goal}} + r_{\text{collision}} + r_{\text{off\_road}} + r_{\text{time\_out}} + r_{\text{closer}} + r_{\text{safe\_dist}}, \tag{26}$$

where each term is further specified as

$$\begin{aligned} r_{\text{reach\_goal}} &= 1000 \cdot \mathbf{1}_{\text{reach\_goal}}, \\ r_{\text{collision}} &= -1000 \cdot \mathbf{1}_{\text{collision}}, \\ r_{\text{off\_road}} &= -1000 \cdot \mathbf{1}_{\text{off\_road}}, \\ r_{\text{time\_out}} &= -200 \cdot \mathbf{1}_{\text{time\_out}}, \\ r_{\text{closer}} &= 5\left[s_{\text{goal}}(k-1) - s_{\text{goal}}(k)\right] \\ &\quad + 5\left[d_{\text{goal}}(k-1) - d_{\text{goal}}(k)\right], \\ r_{\text{safe\_dist}} &= -\exp\left(\frac{s_{\text{lead}}}{s_{\text{safe}}}\right) \mathbf{1}_{\text{safe\_dist}}, \end{aligned}$$

where $s_{\text{goal}}(k)$ and $d_{\text{goal}}(k)$ denote the longitudinal and lateral distance between the ego vehicle and the goal region at time $k \in \mathbb{N}$, respectively. Furthermore, s_{lead} denotes the current distance between the ego vehicle and its leading vehicle. The coefficients of each reward term are chosen empirically using grid-search over a defined set to maximize the goal-reaching and minimize the collision and off-road rate.

We optimize the policies using proximal policy optimization (PPO) [48] due to its superior performance for continuous control tasks compared to other state-of-the-art algorithms. Moreover, we use an actor-critic architecture [49] to approximate both the policy and the value function with a shared neural network to reduce variance. The shared policy and value network has two hidden layers with 64 neurons each and the hyperbolic tangent function as its activation function.

B. Results and Discussions

To demonstrate the effect of our safety layer, we train one agent without control barrier functions (i.e., *unsafe agent*) and one agent with control barrier functions (i.e., *safe agent*) on the same traffic scenarios. The results are shown in the learning curves in Fig. 3.

Is the proposed approach safe? Does the safe agent perform too conservatively?

The learning curves of the collision rate and off-road rate in Fig. 3 show that the safe agent does not have any accidents during the entire training, whereas the unsafe agent still had a collision rate of around 5% and a small variance around zero for off-road rate after convergence. In addition, the safe agent converged to a slightly higher goal-reaching rate than the unsafe agent. Therefore, we can conclude that with our safety layer, the agent performed much more safely, yet not too conservatively.

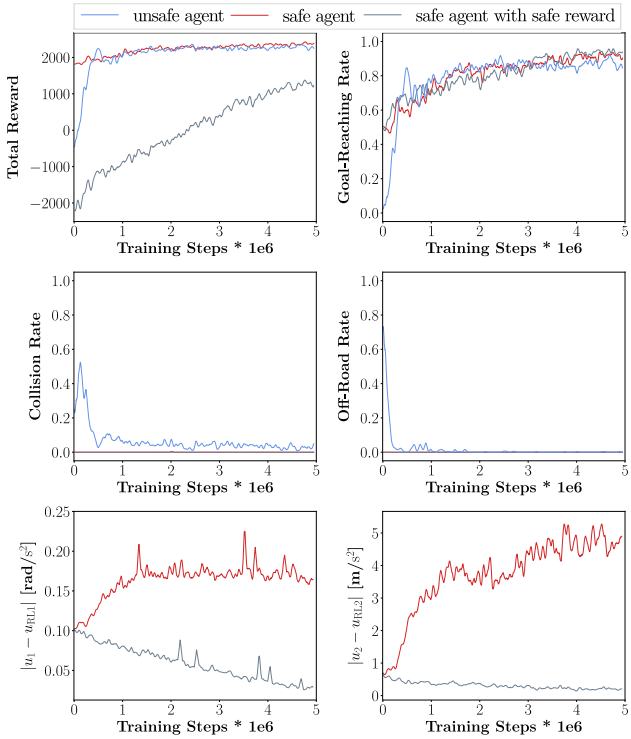

Fig. 3. Learning curves of the unsafe agent, safe agent, and safe agent with safe reward(28).

Furthermore, the safe agent increased its goal-reaching rate from 50% to 90% after training. Since the safety layer constrained the action to the safe area, the RL agent was able to learn the goal-reaching behavior better without having to consider safety, which reduces the learning complexity.

In addition, we compare the performance of the safe and unsafe agents in an example scenario in Fig. 4, where the unsafe agent collides with its leading vehicle, while the safe agent brakes in time and reaches the goal at the end. We use Time-To-Collision (TTC) to measure criticality since it considers the distance and velocity differences between vehicles:

$$t_{\text{TTC}} = \begin{cases} \frac{s_{\text{lead}} - s_{\text{f}}}{v_{\text{f}} - v_{\text{lead}}} & \text{if} \quad v_{\text{f}} > v_{\text{lead}}, \\ \infty & \text{else.} \end{cases} \tag{27}$$

How does the safety layer affect the performance of the agent itself during training?

As observed in [5], an agent trained with a safety layer performed more recklessly when the safety layer is removed compared to the agent trained without a safety layer, since it has never experienced a risky situation during learning. Therefore, to increase the safety of the agent itself as well as accelerate convergence during learning, we train a third agent with additional feedback from the safety layer to the agent through a reward function (i.e., *safe agent with safe reward*). The additional reward term of the safety layer is defined as

$$r_{\text{safe}} = -50\frac{|u_1 - u_{1,\text{RL}}|}{\dot{\psi}_{\max}} - 50\frac{|u_2 - u_{2,\text{RL}}|}{a_{\max}} \tag{28}$$

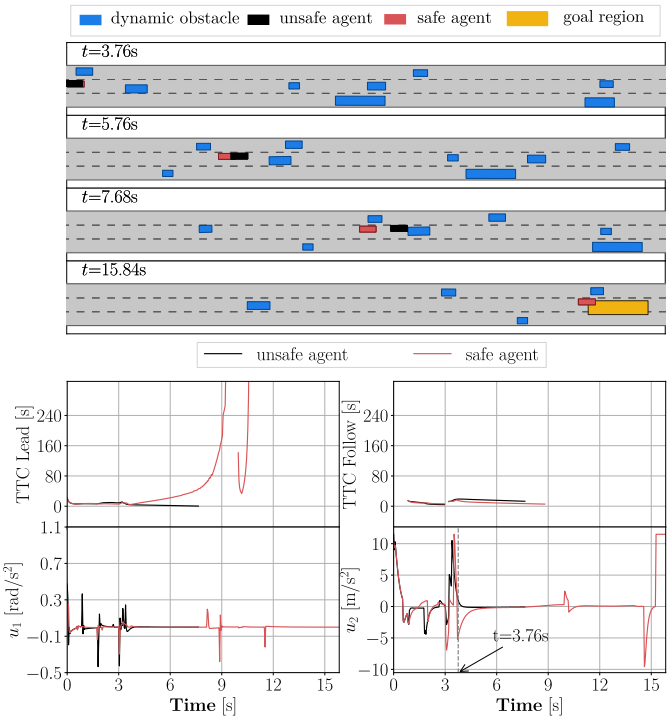

Fig. 4. An example scenario where the unsafe agent collides with its leading vehicle whereas the safe agent brakes in time and reaches the goal at the end.

Figure. 3 shows the learning curves of the safe agents trained with and without r_{safe} (28). Both agents performed equally in terms of reaching the goal area. The safe agent trained without the safe reward achieved a higher reward value since it was not penalized with an extra negative reward. However, the learning curves of the absolute value of the correction term of the actions show that the safe agent with the safe reward needed much less correction/intervention from the safety layer compared to the safe agent without the safe reward. Therefore, the safety of the agent itself was improved through feedback from the safety layer.

How do the trained agents perform in test scenarios?

To further verify the effect of the proposed method, we evaluate the best model obtained in each training on the test scenarios, including the highD test dataset and interactive scenarios in the SUMO simulator [50]. In addition, we evaluate the performance of two safe agents when the safety layer is removed. Table II shows the collision rate, off-road rate, and goal-reaching rate of all best models on the highD test set and interactive scenarios in SUMO, respectively. Neither safe agent has an accident on the test dataset, while still reaching the goal more often than the unsafe agent in the highD scenarios. Furthermore, when the safety layer is removed, the safe agent performed worse than the unsafe agent, whereas the safe agent with safe reward achieved similar performance to the unsafe agent, demonstrating the effectiveness of the additional safe reward.

V. Conclusions

This paper presents an approach to ensure the safety of RL-based low-level motion planners for autonomous vehicles in highway scenarios. We define the safety constraints based on our previously-developed invariably safe sets and utilize control barrier functions to enforce the invariance of the safe sets. In addition, safety interference is minimized by formulating the problem as a quadratic programming problem. Furthermore, we propose a relaxation mechanism when the optimization problem becomes infeasible due to the illegal behavior of other vehicles. We demonstrate our approach in real-world highway scenarios and show that the RL agent does not violate any safety constraints during learning. Future work will integrate our previously-developed online verification framework [44], [51] to provide a safety guarantee for more general scenarios.

TABLE II
Performance on the Test Scenarios

Agent	Collision rate	Off-road rate	Goal-reaching rate
highD test set			
unsafe agent	3.07%	0.70%	83.36%
safe agent	**0%**	**0%**	**93.22%**
safe agent with safe reward	**0%**	**0%**	88.11
safe agent in [5]	**0%**	**0%**	85.7%
when safety layer is removed			
safe agent	29.27%	18.85%	47.74%
safe agent with safe reward	4.33%	0.34%	89.95%
safe agent in [5]	23%	0%	60.7%
interactive scenarios in SUMO			
unsafe agent	0.89%	0.04%	**88.01%**
safe agent	**0%**	**0%**	68.78%
safe agent with safe reward	**0%**	**0%**	82.42%
when safety layer is removed			
safe agent	4.49%	34.16%	35.33%
safe agent with safe reward	0.77%	0.07%	81.75%

Acknowledgments

The author thanks Zhenyu Li for his help with the numerical experiments. This work is funded by the German Research Foundation Grant AL 1185/3-2.

Appendix
Derivation of (19)

$$
\begin{aligned}
h_3^0 &= \Delta d = d_{\text{obs}} - d_{\text{ego}} \\
h_3^1 &= \dot{h}_3^0 + \gamma\, h_3^0 \\
&= (v_{\text{obs}} \sin e_{\psi,\text{obs}} - v_{\text{ego}} \sin e_{\psi,\text{ego}}) + \gamma\,(d_{\text{obs}} - d_{\text{ego}}) \\
h_4^0 &= \Delta d = d_{\text{ego}} - d_{\text{obs}} \\
h_4^1 &= \dot{h}_4^0 + \gamma\, h_4^0 \\
&= (v_{\text{ego}} \sin e_{\psi,\text{ego}} - v_{\text{obs}} \sin e_{\psi,\text{obs}}) + \gamma\,(d_{\text{ego}} - d_{\text{obs}}).
\end{aligned}
$$

References

[1] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[2] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2019, pp. 8248–8254.

[3] Z. Miljković, M. Mitić, M. Lazarević, and B. Babić, "Neural network reinforcement learning for visual control of robot manipulators," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1721–1736, 2013.

[4] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 3389–3396.

[5] H. Krasowski, X. Wang, and M. Althoff, "Safe reinforcement learning for autonomous lane changing using set-based prediction," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2020, pp. 1–7.

[6] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. of the IEEE Conf. on Decision and Control*, 2014, pp. 6271–6278.

[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[8] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2017.

[9] X. Xu, T. Waters, D. Pickem, P. Glotfelter, M. Egerstedt, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds," in *Proc. of the IEEE Conf. on Control Technology and Applications*, pp. 1769–1775.

[10] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3387–3395.

[11] M. Althoff, S. Maierhofer, and C. Pek, "Provably-correct and comfortable adaptive cruise control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 159–174, 2021.

[12] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 42, pp. 1437–1480, 2015.

[13] C. Gaskett, "Reinforcement learning under circumstances beyond its control," in *Proc. of the Int. Conf. on Computational Intelligence for Modelling Control and Automation*, 2003.

[14] A. Nilim and L. El Ghaoui, "Robust control of markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[15] R. A. Howard and J. E. Matheson, "Risk-sensitive markov decision processes," *Management science*, vol. 18, no. 7, pp. 356–369, 1972.

[16] T. Osogami, "Robustness and risk-sensitivity in markov decision processes," in *Advances in Neural Information Processing Systems*, 2012, pp. 233–241.

[17] T. M. Moldovan and P. Abbeel, "Safe exploration in markov decision processes," *arXiv preprint arXiv:1205.4810*, 2012.

[18] P. Geibel and F. Wysotzki, "Risk-sensitive reinforcement learning applied to control under constraints," *Journal of Artificial Intelligence Research*, vol. 24, pp. 81–108, 2005.

[19] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine learning*, vol. 49, no. 2-3, pp. 267–290, 2002.

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. of the Int. Conf. on Machine Learning*, 2017.

[21] Y. Zhang, Q. Vuong, and K. Ross, "First order constrained optimization in policy space," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[22] S. Pathak, L. Pulina, and A. Tacchella, "Verification and repair of control policies for safe reinforcement learning," *Applied Intelligence*, vol. 48, no. 4, pp. 886–908, 2018.

[23] N. Fulton and A. Platzer, "Verifiably safe off-model reinforcement learning," in *Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, 2019, pp. 413–430.

[24] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[25] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, p. 109597, 2021.

[26] A. Akametalu, S. Kaynama, J. Fisac, M. Zeilinger, J. Gillula, and C. Tomlin, "Reachability-based safe learning with Gaussian processes," in *Proc. of the IEEE Conf. on Decision and Control*, 2015, pp. 1424–1431.

[27] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2156–2162.

[28] D. Isele, A. Nakhaei, and K. Fujimura, "Safe reinforcement learning on autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 1–6.

[29] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.

[30] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *Proc. of IEEE Conf. on Decision and Control*, 2018, pp. 7130–7135.

[31] ——, "Nonlinear learning-based model predictive control supporting state and input dependent model uncertainty estimates," *International Journal of Robust and Nonlinear Control*, 2021.

[32] M. Nagumo, "Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen," in *Proc. of the Physico-Mathematical Society of Japan*, vol. 24, 1942, pp. 551–559.

[33] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. of the IEEE European Control Conference*, 2019, pp. 3420–3431.

[34] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control lyapunov functions and control barrier functions," *arXiv preprint arXiv:2004.07584*, 2020.

[35] J. Chen, B. Yuan, and M. Tomizuka, "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 2884–2890.

[36] G. Notomista, M. Wang, M. Schwager, and M. Egerstedt, "Enhancing game-theoretic autonomous car racing using control barrier functions," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 5393–5399.

[37] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," *arXiv preprint arXiv:2007.11718*, 2020.

[38] C. Pek and M. Althoff, "Efficient computation of invariably safe states for motion planning of self-driving vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 3523 – 3530.

[39] A. De Luca, G. Oriolo, and C. Samson, *Feedback control of a nonholonomic car-like robot*. Springer Berlin Heidelberg, 1998, pp. 171–253.

[40] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[41] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[42] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

[43] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. of the IEEE Conf. on Decision and Control*, 2019, pp. 474–479.

[44] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[45] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[46] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2118–2125.

[47] X. Wang, H. Krasowski, and M. Althoff, "CommonRoad-RL: A configurable reinforcement learning environment for motion planning of autonomous vehicles," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2021, pp. 466–472.

[48] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[49] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[50] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using SUMO," in *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, 2018, pp. 2575–2582.

[51] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 518–528, 2020.