2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

Safe Reinforcement Learning for Urban Driving using Invariably Safe Braking Sets

Hanna Krasowski*,1, Yinqiang Zhang*,2, and Matthias Althoff[1]

***Abstract*— Deep reinforcement learning (RL) has been widely applied to motion planning problems of autonomous vehicles in urban traffic. However, traditional deep RL algorithms cannot ensure safe trajectories throughout training and deployment. We propose a provably safe RL algorithm for urban autonomous driving to address this. We add a novel safety layer to the RL process to verify the safety of high-level actions before they are performed. Our safety layer is based on invariably safe braking sets to constrain actions for safe lane changing and safe intersection crossing. We introduce a generalized discrete high-level action space, which can represent all high-level intersection driving maneuvers and various desired accelerations. Finally, we conducted extensive experiments on the inD dataset containing urban driving scenarios. Our analysis demonstrates that the safe agent never causes a collision and that the safety layer's lane changing verification can even improve the goal-reaching performance compared to the unsafe baseline agent.**

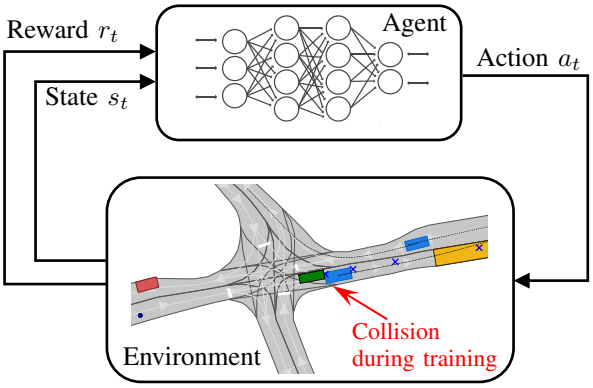


Fig. 1. Conventional RL process including an example collision situation.

I. Introduction

Motion planning in urban areas is challenging because of different road geometries and frequent interactions with traffic participants. A method suited explicitly for solving such tasks is reinforcement learning (RL) [1], [2]. With deep RL algorithms, vehicles can learn to control their motion for different tasks, such as lane-keeping and changing [3], [4], path tracking [5], [6], ramp merging [7], [8], navigating through intersections [9]–[12], and emergency braking [13], [14]. However, most deep RL approaches only focus on one simplified driving sub-tasks. Moreover, learning a driving policy with conventional deep RL is inherently unsafe. As shown in Fig. 1, the stochastic exploration process possibly guides the vehicle to unsafe states, where causing a collision cannot be avoided anymore. Furthermore, frequent visits to unsafe and meaningless states can decrease learning efficiency. To mitigate this problem, the exploration of RL agents can be directed or constrained. Lu et al. [15] designed risk networks that can guide a safe policy optimization. However, their approach cannot guarantee safety during driving. Another method is using control barrier functions [16], [17]; however, finding suitable control barrier functions for complex tasks, for example, urban autonomous driving, is not trivial.

To efficiently guarantee safety for an RL agent, we propose a safe RL algorithm for autonomous driving in urban scenarios, where safe actions are identified by a safety layer and the action selection of the RL agent is constrained such that only safe actions can be selected. Our work generalizes our pre-study on highway driving [18] so that it is also applicable in an urban setting. Notably, our contributions are as follows:

- By combining invariably safe braking sets and conflict zones, we introduce a safety layer that can verify the safety of junction crossing.
- We propose a generalized high-level action space to solve various driving tasks in urban scenarios.
- We conducted extensive numerical experiments and an ablation study to show the validity and efficiency of our implementation.

The remainder of this paper is structured as follows: Section II shows the related literature on RL algorithms for autonomous driving and the safety guarantees of RL algorithms. Section III describes the details of our proposed algorithms, particularly the safety layer. Section IV records the experimental settings, results of conducted experiments, and an ablation study. Finally, we conclude in Section V.

II. Related Work

RL research on motion planning for autonomous driving mainly differs in tasks and action specifications. Usually either high-level actions [19]–[21] or direct control inputs [5], [9], [17], [22] are learned, which differ depending on the regarded driving scenario. Furthermore, only some researchers tried to incorporate safety measures. We review current research on RL for autonomous driving and methods for extending RL to safe RL.

*The first two authors have contributed equally to this work.

[1]Hanna Krasowski and Matthias Althoff are with the Department of Informatics, Technical University of Munich, 85748 Garching, Germany, {hanna.krasowski, althoff}@tum.de

[2]Yinqiang Zhang is with the Department of Computer Science, University of Hong Kong, Hong Kong, China zyq507@connect.hku.hk

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922166

A. Reinforcement Learning for Autonomous Driving

To solve various driving tasks in urban scenarios, the action space should be appropriately designed. One RL approach is applying learned actions directly to the ego vehicle. With this end-to-end approach, the agent chooses an action value from a continuous action space, such as a speed set-point and steering angle [5], a velocity [22], an acceleration [9], or a yaw rate [4]. For these continuous action spaces, the agent often needs more training steps to learn the optimal policy because of the infinite number of action values that can be explored.

Other RL approaches use a discrete high-level action space, where actions typically represent different maneuvers. For instance, maneuvers [19], [23] are a commonly used action representation for lane keeping and changing tasks. A three-layer architecture for the lane-changing and left-turning tasks was recently proposed by Qiao et al. [20]. The top-level policy chooses a maneuver. An optimal trajectory is then created and tracked by a PID controller. Isele et al. [21] proposed three discrete action spaces for driving at intersections. They evaluated their approach on simulated traffic and found that the action space with a creep action (i.e., moving slowly) performs the best with occlusions near the intersection area. Li et al. [12] proposed a hierarchical framework with a high-level action space consisting of reference speeds and low-level controllers for intersection and round-about driving. Their evaluation shows that their approach can achieve high completion rates but causes more collisions than more conservative approaches. Many maneuvers such as lane following, lane changing, and intersection crossing have to be regarded in urban areas. A discrete action space can be used to efficiently learn in such a complex environment.

B. Provably Safe Reinforcement Learning

To guarantee safety, the agent's exploration must be limited to the safe state space. For that, two approaches are most relevant: advising the agent after the action selection with a possibly adapted safe action or constraining actions to safe actions before the agent can choose one [24]. For the first approach, usually, a penalty is given in case a correction is necessary [16], [17], [25], [26]. Saunders et al. [26] proposed a trained human-like supervisor in their RL algorithm to intervene in the agent's behavior when it tends to go into unsafe or risky states. Cheng et al. [16] presented an end-to-end safe RL algorithm, where control barrier functions restrict exploration and deployment. Similarly, Wang [17] proposed control barrier functions to achieve end-to-end safe RL for autonomous highway driving.

The second approach removes unsafe and meaningless actions in advance [9], [18], [27]–[29]. Only actions that entirely satisfy safety specifications are accessible to the agent. For instance, the methods in [9], [28] ensure safe intersection navigation by verifying safety with linear temporal logic and differential dynamic logic. Additionally, Q-masking removes meaningless and unsafe actions for Q-learning [23], [29]. For example, Mirchevska et al. [29] used the safe braking distance to decide, which actions are unsafe and need to be masked out. Krasowski et al. [18] built on this work and present a safety layer for highway driving, which generates safe action masks for the proximal policy optimization (PPO) algorithm [30]. They use set-based predictions [31] for the other traffic participants to identify safe actions. In this work, we build on the masking approach, which allows us to ensure safety by identifying safe actions in advance of their execution.

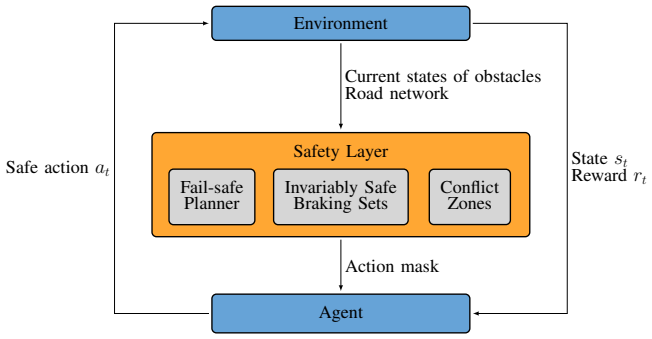


Fig. 2. Implementation overview of RL framework.

III. Safe Reinforcement Learning in Urban Environments

RL problems can be formulated as Markov decision processes, which is illustrated in Fig. 1. Our safe RL framework is extended by a safety layer (see Fig. 2). This safety layer generates an action mask that indicates the safe actions and removes unsafe and meaningless actions, e.g., actions that would lead off-road or actions that violate safe distances to other traffic participants. As a result, the agent can explore only safe actions. We use PPO with action masking as our learning approach and refer the interested reader to [18], [32] for implementation and theoretical details. The observation space, action space, and reward function employed in this work are first introduced in the following parts. The safety layer, which incorporates the concept of conflict zones [33] and invariably safe sets [34], is then thoroughly explained.

A. Observation Representation

Our 40-dimensional continuous state space consists of 26 observations from CommonRoad-RL [35] and 14 new intersection-related observations (see Table I). To define the intersection-related observations, we need to specify the intersection area, which is the area that is mutually accessible to vehicles arriving at the intersection from different entries. Furthermore, intersection-entering vehicles are those for which the following hold:

- the position is at most the longitudinal distance $s_{\text{intersection}}$ away from the intersection,
- the vehicle is not a lane-based surrounding vehicle [35], i.e., not surrounding the ego vehicle on the ego vehicle's lane or on adjacent lanes in the same driving direction,
- the vehicle is driving toward the intersection.

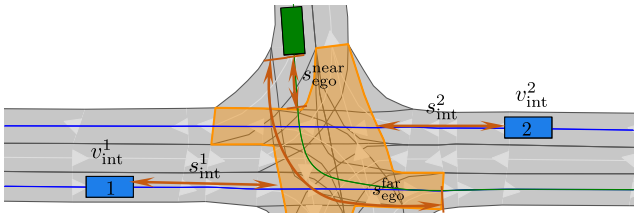


Fig. 3. Visualization of intersection-related observations: Relative distances $s_{\text{int}}^1, s_{\text{int}}^2$ between the vehicle $1, 2$ and the orange intersection, absolute velocities v_{int}^1 and v_{int}^2 of vehicle $1, 2$, and ego vehicle distances to intersection $s_{\text{ego}}^{\text{near}}$ and $s_{\text{ego}}^{\text{far}}$. The driving directions are indicated by light gray arrows.

The intersection-related observations are illustrated in Fig. 3. The first intersection-related observations are the absolute velocities v_{int}^i and relative distances s_{int}^i to the intersection for the intersection-entering vehicles $i \in 1, ..., n_{\text{intersection}}$. The observations are sorted based on the vehicles' distances to the intersection so that only the $n_{\text{intersection}}$ closest vehicles are considered. If fewer vehicles are detected, we set the relative distance and velocity to the predefined values $s_{\text{intersection}}$ (here $50\,\text{m}$) and $0\,\text{m}\,\text{s}^{-1}$, respectively. The remaining intersection-related observations are the longitudinal distances along the reference lane between the ego vehicle position and the intersection (cf. $s_{\text{ego}}^{\text{near}}$ and $s_{\text{ego}}^{\text{far}}$ in Fig. 3).

B. Action Representation

Driving in urban traffic requires various maneuvers. We used a two-level framework to represent this. The policy chooses the maneuver on the higher level, and the sampling-based motion planner from [36] concretizes the maneuver to a drivable trajectory on the lower level. The high-level action space consists of three action types. The first action type a_{lane} indicates maneuvers restricted to the current and adjacent lanes, i.e., change to left ($a_{\text{lane}} = 0$), or right lane ($a_{\text{lane}} = 2$), and keep driving in the current lane ($a_{\text{lane}} = 1$). The second action type a_{dir} can take three values and describes the driving directions at the next intersection, for example, turn left, right, or go straight. Note that if there is just one possible direction for driving, only $a_{\text{dir}} = 0$ is used; if there are two possible driving directions, $a_{\text{dir}} = 0$ corresponds to the left-most action and $a_{\text{dir}} = 1$ to the other. The third action type a_{acc} represents the desired longitudinal accelerations. Seven values can be selected: $\mathcal{A}_{acc} = \{0\,\text{m}\,\text{s}^{-2}, \pm 1.0\,\text{m}\,\text{s}^{-2}, \pm 2.0\,\text{m}\,\text{s}^{-2}, \pm 4.0\,\text{m}\,\text{s}^{-2}\}$. All possible combinations of the three action types ($a_{\text{lane}} \times a_{\text{dir}} \times a_{\text{acc}}$) lead to the action set $\mathcal{A}_{\text{regular}}$, which represent the regular maneuvers possible at an at most four-legged intersection. These 63 regular actions plus the fail-safe action lead to a 64-dimensional discrete action space $\mathcal{A}$. More complex intersections can be represented by extending the possible values for a_{lane} and a_{dir}.

TABLE I

40-DIMENSIONAL CONTINUOUS STATE SPACE

Dim.	Description
1-6	distance between ego vehicle and six lane-based surrounding traffic participants
7-12	velocity between ego vehicle and six lane-based surrounding traffic participants
13-14	velocity and acceleration of the ego vehicle
15-16	longitudinal distance and motion advance to goal area
17-18	lateral distance and motion advance to goal area
19-23	lateral distances from dynamically extrapolated ego vehicle positions to goal
24	remaining time steps to reach the goal area
25	orientation of goal area
26	remaining time steps in scenario
27-28	longitudinal distances to intersection area ($s_{\text{ego}}^{\text{near}}$, $s_{\text{ego}}^{\text{far}}$)
29-34	distance between intersection and six traffic participants ($s_{\text{int}}^{\text{i}}$ for $i = 1, ..., 6$)
35-40	velocity between ego vehicle and six lane-based surrounding traffic participants ($v_{\text{int}}^{\text{i}}$ for $i = 1, ..., 6$)

Note that the upper 26 observations are implemented as in [35] and the remaining 14 observations are derived in Sec. III-A.

C. Reward Function

We use sparse and dense components for the reward function. The sparse components are:

$$\begin{aligned} r_{\text{reach\_goal}} &= 50 \cdot \mathbf{1}_{\text{reach\_goal}}, \\ r_{\text{time\_out}} &= -10 \cdot \mathbf{1}_{\text{time\_out}}, \\ r_{\text{collision}} &= -50 \cdot \mathbf{1}_{\text{collision}}, \\ r_{\text{mask}} &= -10 \cdot \mathbf{1}_{\text{mask}}, \end{aligned}$$

where $\mathbf{1}_{\square}$ denotes binary variables that evaluate to 1 if the corresponding condition $\square$ is satisfied. Particularly, the reward r_{mask} is given when no regular action is verified as safe and the fail-safe planner is activated. The dense reward component guides the agent toward the goal at each time step:

$$r_{\text{goal\_guiding}} = \frac{-40 \cdot \Delta d_{\text{lat}}^t + 20 \cdot \Delta d_{\text{lon}}^t}{d_{\text{lon}}^{\text{total}}}, \tag{1}$$

where Δd_{lat}^t and Δd_{lon}^t are the position differences toward the goal in the longitudinal and lateral directions within a curvilinear coordinate system (introduced in Sec. III-D) at time step t compared to the previous time step. To reduce the influence of different distances between the initial state and the goal, we divide by the longitudinal distance $d_{\text{lon}}^{\text{total}}$ from the initial position to the goal. The final reward function is:

$$\begin{aligned} r = &r_{\text{reach\_goal}} + r_{\text{time\_out}} \\ &+ r_{\text{collision}} + r_{\text{mask}} + r_{\text{goal\_guiding}}. \end{aligned} \tag{2}$$

D. Preliminaries and Assumptions for the Safety Layer

The road network consists of lanelets [37], which are atomic, interconnected, and drivable road segments. We condensate the road network into a set of lanes $\mathcal{L}$. A lane is defined as a set of longitudinally adjacent lanelets from a lanelet that has no predecessor to a lanelet that has no successor [38]. We assign unique identifiers to the lanes and use $\mathcal{L}_k$ to specify the occupancy of the lane with the identifier k. In addition, $\mathcal{K}$ is the set of identifiers of all lanes in the scenario.

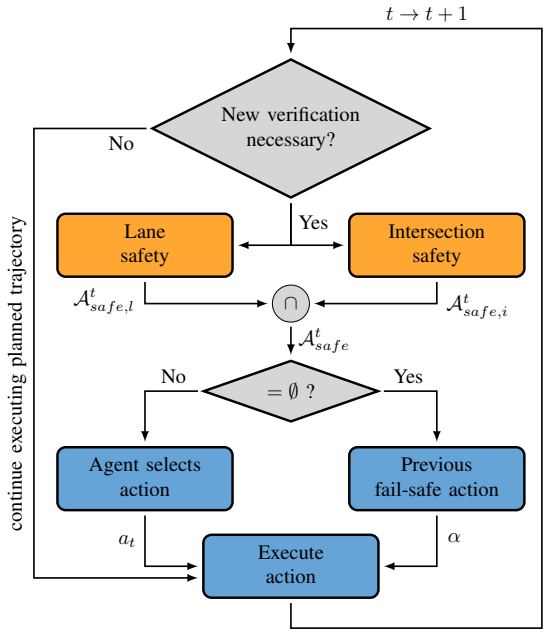


Fig. 4. Safety verification flowchart. Orange blocks belong to the safety layer and blue ones to the RL agent and environment.

We use a curvilinear coordinate system along the lanes such that the ego vehicle's state at each time step t is $x_t = (s, d, v)$, where s is the longitudinal position along the lane, d is the lateral position, and v is the velocity. The function $\text{proj}_s(x_t)$ returns the longitudinal position for a state x_t. The function plan creates the set of ego vehicle states $\{(x_{ego,0}, 0), ..., (x_{ego,t_p}, t_p), ..., (x_{ego,t_f}, t_f)\}$ for all time steps until the final time t_f. The function uses the sampling-based planner from [36] to generate the set of states for action $a = (a_{\text{lane}}, a_{\text{dir}}, a_{\text{acc}})$ until the planning horizon t_p and then attaches the fail-safe maneuver indicated by α until the final time t_f. The two types of fail-safe maneuvers considered in this work are braking with maximum deceleration $-a_{max}$ until standstill (i.e., $\alpha = 1$), or accelerating with maximum acceleration a_{max} until the ego vehicle fully left the intersection and then braking until standstill with maximum deceleration (i.e., $\alpha = 0$). To adequately address the computational demands of RL, we only consider on these two fail-safe maneuvers. Our verification is based on the assumptions that the absolute acceleration of all vehicles is less or equal to the maximum acceleration a_{max}. If traffic participants cause accidents by not respecting traffic rules, these collisions are considered to be not the fault of the ego vehicle.

E. Safety Layer

The safety layer (see Fig. 4) identifies the safe discrete action space $\mathcal{A}^t_{\text{safe}}$ when (a) the time step $(t_p - \Delta t)/\Delta t$ since the last verification cycle is reached, (b) the accessible road network for the ego vehicle changed since the last time step, or (c) a lane change finished. Note that we start the verification at least one time step before the planning horizon t_p is reached to simulate that for real-world experiments the verification calculations must be finished before the planning horizon is reached. If a verification is necessary, the trajectories for all regular actions $\mathcal{A}_{\text{regular}}$ are generated, and we check if safety can be verified in the two relevant safety dimensions: *lane safety* verification for distances to the leading vehicle and lane-changing maneuvers results in the safe action set $\mathcal{A}^t_{\text{safe,l}}$ and *intersection safety* verification for crossing intersections results in the safe action set $\mathcal{A}^t_{\text{safe,i}}$. Thus, the set of safe actions at time step t is:

$$\mathcal{A}^t_{\text{safe}} = \mathcal{A}^t_{\text{safe,l}} \cap \mathcal{A}^t_{\text{safe,i}} \ . \tag{3}$$

Subsequently, the RL agent selects an action from $\mathcal{A}^t_{\text{safe}}$ and the previously calculated fail-safe action. When no action from $\mathcal{A}_{\text{regular}}$ can be verified as safe, the fail-safe trajectory attached to the previously chosen action is executed.

a) Identifying meaningful actions: To minimize the verification effort, first, we identify if the action is meaningful with the predicate $\texttt{meaningful}(a_t, a_{t-1}, x_t)$ where a_t is the action to verify, a_{t-1} is the action of the previous time step, and x_t is the ego vehicle's state. This predicate evaluates to true if and only if for $a_t, a_{t-1} \in \mathcal{A}_{\text{regular}}$:

- the lane to change to exists for a_{lane}, and
- the driving direction of a_{dir} is permitted, and
- no lane change is currently conducted.

However, if the action verification determines that it is unsafe to proceed with the lane change, it will be aborted and the fail-safe plan will be executed instead.

b) Verifying safe actions: Only for meaningful maneuvers, trajectories for the desired accelerations are generated. To verify the safety of a trajectory, we use a subset of the invariably safe set $\mathcal{S}^t$ [34, Proposition 1] – the invariably safe braking set $\mathcal{S}^t_1$ [34, Algorithm 1, line 10]:

$$\begin{aligned}\mathcal{S}^t_1 \leftarrow \{(s,d,v)^T \in \mathcal{X} \,|\, \forall s_j \in \mathcal{O}_j(t) : s \leq s_j - \Delta^t_{\text{safe}}(v, b_j) \\ \wedge\, v \leq v_{\max} \wedge s \in \mathcal{C}_{b_i,b_j}\},\end{aligned} \tag{4}$$

where $\mathcal{X}$ is the state space, $\mathcal{O}_j(t)$ is the predicted occupancy for an obstacle b_j, s_j is its longitudinal position, $\Delta^t_{\text{safe}}(v, b_j)$ is its safe distance to the ego vehicle, $v_{\max}$ is the speed limit, and $\mathcal{C}_{b_i,b_j}$ is the part of the road network (e.g., a lane) regarded for the invariably safe braking set calculation of obstacles b_i and b_j. In a nutshell, driving in the invariably safe braking set $\mathcal{S}^t_1$ guarantees safety for a vehicle in a lane based on its current position and velocity, obstacle dynamics, and safe distance constraints. We define $\mathcal{S}^t_{\mathcal{L}_k}$ as the invariably safe braking set $\mathcal{S}^t_1$ of a lane k at time step t (Eq. (4) with $\mathcal{C}_{b_i,b_j} = \mathcal{L}_k$).

The verification of *lane safety* is depicted in Algorithm 1. For a given action a and ego vehicle state $x_{ego,0}$, the function $\texttt{get\_current\_lane}(a, x_{ego,0})$ returns an identifier e, which indicates the current lane of the ego vehicle and its driving direction (cf. line 4). Then, the invariably safe braking set for all vehicles in this lane in front of the ego vehicle is calculated in line 5. If $a_{lane} = 1$, then the action is a lane-keeping action. For these lane-keeping actions, we only verify the safety of the planned trajectories with respect

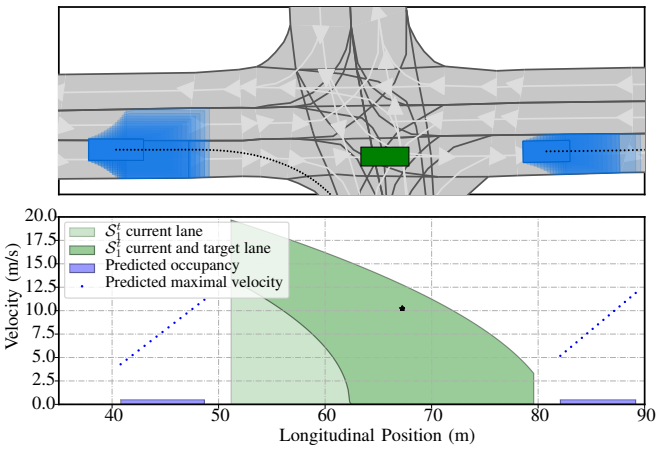


Fig. 5. Safety verification in a lane. Top: scenario with green ego vehicle and blue occupancy predictions for other vehicles; bottom: velocity and position for the invariably safe region in green, predicted maximal velocities and occupancies for traffic participants in blue, and ego state marked with a star.

to the leading vehicles (cf. line 7). For other actions, the function $\texttt{get\_target\_lane}(a, x_{ego,0})$ returns the identifier for the target lane of the lane change and we verify the safety with respect to all vehicles on the target lane and the leading vehicles on the current lane (cf. line 9-10). The study [39] describes a similar online verification for fail-safe planning of autonomous vehicles. In contrast to their work, we only considered two fail-safe maneuvers and limit the invariably safe set to the invariably safe braking set. This increases the efficiency of the implementation, which is necessary because of the learning setting of our work. Fig. 5 visualizes the invariably safe sets for an example lane safety situation.

For *intersection safety*, we verify the agent's actions at intersections such that the ego vehicle does not access an intersection in case another traffic participant could occupy it.

Algorithm 1 laneSafety()

Input: $\mathcal{S}^t_{\mathcal{L}_k}\ \forall k \in \mathcal{K}$, $x_{ego,0}$, a_{t-1}
Output: Safe lane actions $\mathcal{A}^t_{\text{safe,l}}$

1: $\mathcal{A}^t_{\text{safe,l}} := \emptyset$
2: **for all** $a \in \mathcal{A}_{\text{regular}} \wedge \texttt{meaningful}(a, a_{t-1}, x_{ego,0})$ **do**
3: **for all** $\alpha \in \{0, 1\}$ **do**
4: $e := \texttt{get\_current\_lane}(a, x_{ego,0})$
5: $\mathcal{S}^t_{\mathcal{L}_{e,lead}} := \{(s, d, v)^T \in \mathcal{S}^t_{\mathcal{L}_e} | s \geq \text{proj}_s(x_{ego,0})\}$
6: **if** $a_{lane} = 1$ **then**
7: $\mathcal{A}^t_{\text{safe,l}} \leftarrow \{(a, \alpha) | \text{plan}(a, \alpha) \subset \mathcal{S}^t_{\mathcal{L}_{e,lead}}\}$
8: **else**
9: $c := \texttt{get\_target\_lane}(a, x_{ego,0})$
10: $\mathcal{A}^t_{\text{safe,l}} \leftarrow \{(a, \alpha) | \text{plan}(a, \alpha) \subset \mathcal{S}^t_{\mathcal{L}_{e,lead}} \wedge \text{plan}(a, \alpha) \subset \mathcal{S}^t_{\mathcal{L}_c}\}$
11: **end if**
12: **end for**
13: **end for**
14: **return** $\mathcal{A}^t_{\text{safe,l}}$

Algorithm 2 intersectionSafety()

Input: $\mathcal{A}_{\text{regular}}$, $s_{i,start}$, $s_{i,end}$, $\mathcal{O}$, X, $\mathcal{L}$, $x_{ego,0}$, a_{t-1}
Output: Safe intersection actions $\mathcal{A}^t_{\text{safe,i}}$

1: $\mathcal{A}^t_{\text{safe,i}} := \emptyset$
2: **for all** $a \in \mathcal{A}_{\text{regular}} \wedge \texttt{meaningful}(a, a_{t-1}, x_{ego,0})$ **do**
3: **for all** $\alpha \in \{0, 1\}$ **do**
4: $\mathcal{CO} := \emptyset$
5: $e := \texttt{get\_current\_lane}(a, x_{ego,0})$
6: **for all** $o \in \mathcal{O}$ **do**
7: $\mathcal{C}_o := \texttt{get\_accessible\_lanes}(x_o) \cap \mathcal{L}_e$
8: $t_{cz} := \texttt{get\_t\_conflict}(x_o, \mathcal{C}_o)$
9: $\mathcal{CO} \leftarrow \{(s, t) | t \geq t_{cz} \wedge \min_s(\mathcal{C}_o) \leq s \leq \max_s(\mathcal{C}_o)\}$
10: **end for**
11: $\mathcal{A}^t_{\text{safe,i}} \leftarrow \{(a, \alpha) | \text{plan}(a, \alpha) \cap \mathcal{CO} = \emptyset\}$
12: **end for**
13: **end for**
14: **return** $\mathcal{A}^t_{\text{safe,i}}$

In contrast to other research on intersection safety [40], [41], our approach can deal with arbitrary real-world drivers and does not assume a cooperative setting. Algorithm 2 specifies the verification process. For all actions a and fail-safe actions α, we first identify the current lane e (cf. line 5). Then, we calculate the conflict zones $\mathcal{C}_o$ by intersecting the accessible lanes of each surrounding vehicle o with the lane $\mathcal{L}_e$, which corresponds to the regarded action (cf. line 7). For that, we use the obstacle set $\mathcal{O}$ containing identifiers for all obstacles within a circle around the ego vehicle's center with radius r_{int}. The current state of an obstacle x_o is obtained from the matrix $X = [x_1, ..., x_{\mathcal{O}}] \in \mathbb{R}^{N \times \mathcal{O}}$ where N is the number of state dimensions. The function $\texttt{get\_t\_conflict}(x_o, \mathcal{C}_o)$ returns the last time step t_{cz} before the surrounding obstacle o could reach its conflict zone with the ego vehicle $\mathcal{C}_o$ (cf. line 8). The reaching time is when the surrounding obstacle's occupancy (predicted using the SPOT [31] tool) intersects with the conflict zone $\mathcal{C}_o$. With the conflict zones $\mathcal{C}_o$ and the time step t_{cz}, we generate a collision object $\mathcal{CO}$ that describes the potential occupation of the conflict zones $\mathcal{C}_o$ for all surrounding obstacles o (cf. line 9). Finally, an action is safe if its corresponding trajectory, which includes the fail-safe trajectory, does not intersect with the collision object $\mathcal{CO}$ (cf. line 11).

IV. Experiments

We evaluated our implementation with recorded urban traffic data. First, we exhaustively specify the experimental setup. Then, we present the results, followed by an ablation study, and discuss our findings.

A. Experimental Setup

The inD dataset [42] contains recorded traffic data from four urban locations in Aachen, Germany. Particularly, two locations are at four-legged intersections (abbreviated by AAH_1 for Bendplatz and AAH_2 for Frankenburg) and

TABLE II
EXPERIMENTAL PARAMETERS

Parameter	Value	Parameter	Value
a_{max}	$8\,\mathrm{m\,s^{-2}}$	t_p	$0.4\,\mathrm{s}$
Δt	$0.04\,\mathrm{s}$	r_{int}	$50\,\mathrm{m}$
$s_{\text{intersection}}$	$50\,\mathrm{m}$	$n_{\text{intersection}}$	6

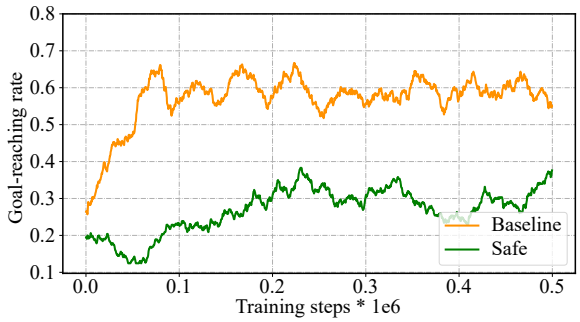
another at a T-junction (AAH_3 for Heckstraße). In this study, we excluded the data from the more complex T-junction at Neuköllner Strasse because without considering traffic signs and lights, the agent cannot reach the goal when the safety layer is activated and it mostly stops at an intersection. An open-source data converter[1] was used to convert the raw data was converted into CommonRoad scenarios [43]. Pedestrians and bicyclists were excluded for this study. Furthermore, we exactly detected the positions and velocities of the vehicles from the scenario data and no occlusions occured. To generate planning problems for the RL agent, one vehicle was removed from each scenario and its initial and finial state enlarged by the spatial dimensions of the vehicle were used as initial state and goal region for the planning problem. If the initial state of a generated scenario is not invariably safe, we did not use this scenario for learning. Additionally, we excluded scenarios where other vehicles appear in the scenario within the first planning cycle of the ego vehicle and close to the ego vehicle. Since these vehicles were absent for the first safety verification, they can lead to collisions due to the scenario data. Overall, we generated approximately 5000 traffic scenarios for the learning. Particularly, we used 1966 scenarios for AAH_1, 1904 for AAH_2, and 959 for AAH_3. The time step size for the scenarios is $0.04\,\mathrm{s}$ while the agent can decide on a new action every 10 time steps (i.e., $0.4\,\mathrm{s}$) in case a lane change did not finish before or the meaningful actions changed (see Fig. 4). All experimental parameters are specified in Table II.

We trained a safe and baseline agent without safety verification on each of the three inD locations and evaluated them on the test set. The safe agent uses the full safety layer for action verification. The baseline agent only uses the safety layer to eliminate meaningless actions (see Sec. III-E.a), thereby increasing the learning efficiency. For each experiment, we split the dataset into 70% training and 30% test sets. The implementation was based on the CommonRoad-RL[2] environment [35] and the stable baselines algorithm toolbox[3]. The PPO parameters and policy network architecture were identified by hyperparameter tuning. We trained every agent 500 000 training steps, which took approximately 24 hours for the safe agents with one thread on a machine with an AMD EPYC 7742 2.2 GHz processor and 1024 GB of DDR4 3200 MHz memory.

[1]commonroad.in.tum.de/dataset-converters

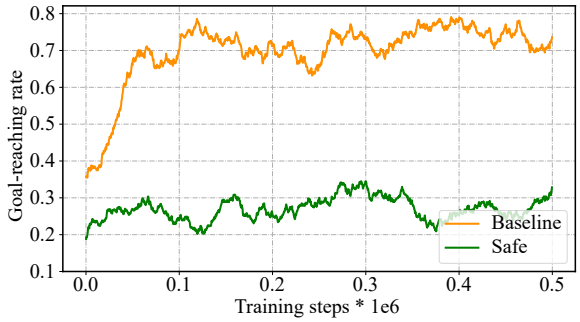
[2]We plan to release the exact implementation of this study with the next CommonRoad-RL release (commonroad.in.tum.de/commonroad-rl).

[3]https://github.com/hill-a/stable-baselines



(a) Goal-reaching rate of agents at location AAH_1.

(b) Goal-reaching rate of agents at location AAH_2.



(c) Goal-reaching rate of agents at location AAH_3.

Fig. 6. Goal-reaching rates for agents during training.

B. Results

The goal-reaching performance of the safe and baseline agent shows a approximately constant gap between 20 % and 40 % (see Fig. 6). The safe agent reached the goal for the AAH_1 location most often and the baseline agent for the AAH_3 location. The baseline agents still collided for 1.4 % to 7.2 % of the scenarios, whereas the safe agents did not cause any collision. Evaluation on the test set is similar to the training results indicating that the agents are not overfitted to the training set. The detailed training and testing results are shown in Table III.

C. Ablation Study

We conducted an ablation study to identify the benefits of the safety layer and its components. Therefore, we trained two additional safe agents: one restricts the actions only with the *intersection safety* (named safe int. agent) and

TABLE III

EVALUATION OF TRAINED AGENTS ON TRAINING AND TEST SETS FOR GOAL-REACHING RATE (COLLISION RATE).

Agent	AAH_1	AAH_2	AAH_3
Training Dataset			
Safe	30.4% (**0.0**%)	27.2% (**0.0**%)	29.1% (**0.0**%)
Safe lane	**73.0**% (3.1%)	**55.8**% (5.4%)	**73.9**% (1.5%)
Safe int.	42.1% (4.4%)	33.9% (4.8%)	52.4% (2.4%)
Baseline	65.1% (3.9%)	46.5% (6.8%)	72.6% (2.1%)
Test Dataset			
Safe	29.9% (**0.0**%)	25.3% (**0.0**%)	28.8% (**0.0**%)
Safe lane	**75.8**% (1.9%)	**54.6**% (4.6%)	71.2% (2.4%)
Safe int.	43.3% (4.5%)	30.4% (4.9%)	51.7% (2.4%)
Baseline	65.9% (4.1%)	44.4% (7.2%)	**75.0**% (1.4%)

Note: The collision rate is revised by collisions not caused by the ego vehicle, for example, another vehicle driving into the ego vehicle from behind, thus, violating the safe distance to the ego vehicle.

the other restricts the actions with the *lane safety* (named safe lane agent). The detailed evaluation results for the trained agents are shown in Table III. For the safe lane agent, the collision rate reduces to less than 5.5% for the training and test scenarios. Interestingly, at the same time the goal-reaching rate increased compared to that of the unsafe baseline agent. Thus, the agent learns better when guided by less and safer actions. For the agent whose actions are only restricted by intersection safety, the collision rate slightly increases compared to the baseline. Furthermore, the goal-reaching performance decreases on the training and test datasets compared to that of the baseline agent. However, only if the two concepts are combined in the safe agent, no collision caused by the ego vehicle occurred.

D. Discussion

The goal-reaching rate for the safe agent is comparably low. This is primarily due to the conservative setting of the parameters, which is necessary to guarantee safety with the current assumptions. However, integrating urban traffic rules in the verification of the safe actions could decrease conservative behavior in crowded intersections. This is supported by preliminary experiments on the data of the more complex T-junction at Neuköllner Strasse in Aachen. Furthermore, we plan to use our more holistic verification approach [44] in the future to alleviate conservativeness. This study has not realized this due to the RL's required low computation times.

Additionally, the current fail-safe planner is optimized for driving comfort and, thus, has limited capabilities to execute quick and uncomfortable reactions to maintain safety. Therefore, an advanced fail-safe planner [45] could be integrated. This is particularly important when human drivers break traffic rules since the autonomous agent needs to respond as quickly as possible to minimize the chances of an accident. However, the challenge is to decide for the correct time to use the fail-safe planner [46]. Additionally, formalized traffic rules could help to efficiently detect when and if a fail-safe planner should be activated.

To make our approach applicable to the real world, other traffic participants, such as pedestrians and cyclists, need to be included in the calculation of the invariably safe sets. Further, all urban traffic rules must be integrated into the verification process. Additionally, the current Python implementation would need to be computationally more efficient and possibly needs refactoring to C++. These issues are subject to future research.

V. CONCLUSIONS

We present a provably safe RL approach for urban driving that can simultaneously handle lane-changing and intersection crossing. Our general high-level action space can be applied to various intersection types. We show the capabilities of our approach on real-world traffic data from three intersections in Germany. These experiments demonstrate that our safety layer is inherently safe and provides safety guarantees for the ego vehicle. The ablation study indicates that compared to the unsafe baseline, adding the lane safety verification improves the performance while reducing collisions. To boost the provably safe RL agent's goal-reaching rate in the future, more traffic rules, a more complex fail-safe planner, better informed set-based prediction, and online verification of arbitrary maneuvers should be investigated.

ACKNOWLEDGMENT

The authors gratefully acknowledge the partial financial support of this work by the research training group ConVeY funded by the German Research Foundation under grant GRK 2428.

REFERENCES

[1] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2022.

[2] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.

[3] Z. Wang, Z. Yan, and K. Nakano, "Comfort-oriented haptic guidance steering via deep reinforcement learning for individualized lane keeping assist," in *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 2019, pp. 4283–4289.

[4] P. Wang, C.-Y. Chan, and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2018, pp. 1379–1384.

[5] 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 IEEE International Conference on Robotics and Automation*, 2019, pp. 8248–8254.

[6] I.-M. Chen and C.-Y. Chan, "Deep reinforcement learning based path tracking controller for autonomous vehicle," *Proc. of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 2-3, pp. 541–551, 2021.

[7] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2019, pp. 3441–3447.

[8] S. Triest, A. Villaflor, and J. M. Dolan, "Learning highway ramp merging via reinforcement learning with temporally-extended actions," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2020, pp. 1595–1600.

[9] D. Isele, A. Nakhaei, and K. Fujimura, "Safe reinforcement learning on autonomous vehicles," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 1–6.

[10] M. Shikunov and A. I. Panov, "Hierarchical reinforcement learning approach for the road intersection task," in *Proc. of Biologically Inspired Cognitive Architectures Meeting*, 2019, pp. 495–506.

[11] Y. Guan, Y. Ren, H. Ma, S. E. Li, Q. Sun, Y. Dai, and B. Cheng, "Learn collision-free self-driving skills at urban intersections with model-based reinforcement learning," in *Proc. of IEEE International Intelligent Transportation Systems Conference*, 2021, pp. 3462–3469.

[12] J. Li, L. Sun, J. Chen, M. Tomizuka, and W. Zhan, "A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning," in *Proc. of IEEE International Conference on Robotics and Automation*, 2021, pp. 2660–2666.

[13] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous braking system via deep reinforcement learning," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2017, pp. 1–6.

[14] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5876–5888, 2020.

[15] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng, "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2020, pp. 1–7.

[16] 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 AAAI Conference on Artificial Intelligence*, 2019, pp. 3387–3395.

[17] X. Wang, "Ensuring safety of learning-based motion planners using control barrier functions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4773–4780, 2022.

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

[19] C. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 294–305, 2020.

[20] Z. Qiao, J. Schneider, and J. M. Dolan, "Behavior planning at urban intersections through hierarchical reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 2667–2673.

[21] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *Proc. of IEEE International Conference on Robotics and Automation*, 2018, pp. 2034–2039.

[22] D. Kamran, C. F. Lopez, M. Lauer, and C. Stiller, "Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2020, pp. 1205–1212.

[23] T. Tram, I. Batkovic, M. Ali, and J. Sjöberg, "Learning when to drive in intersections by combining reinforcement learning and model predictive control," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2019, pp. 3263–3268.

[24] H. Krasowski, J. Thumm, M. Müller, X. Wang, and M. Althoff, "Provably safe reinforcement learning: A theoretical and experimental comparison," *arXiv preprint arXiv:2205.06750*, 2022.

[25] Z. Li, U. Kalabić, and T. Chu, "Safe reinforcement learning: Learning with supervision using a constraint-admissible set," in *Proc. of Annual American Control Conference*, 2018, pp. 6390–6395.

[26] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without error: Towards safe reinforcement learning via human intervention," in *Proc. of International Conference on Autonomous Agents and MultiAgent Systems*, 2018, p. 2067–2069.

[27] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Safe reinforcement learning with scene decomposition for navigating complex urban environments," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2019, pp. 1469–1476.

[28] N. Fulton and A. Platzer, "Safe reinforcement learning via formal methods: Toward safe control through proof and learning," in *Proc. of AAAI Conference on Artificial Intelligence*, 2018, pp. 6485–6492.

[29] 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 IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 2156–2162.

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

[31] M. Koschi and M. Althoff, "SPOT: A tool for set-based prediction of traffic participants," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2017, pp. 1686–1693.

[32] C.-Y. Tang, C.-H. Liu, W.-K. Chen, and S. D. You, "Implementing action mask in proximal policy optimization (PPO) algorithm," *ICT Express*, vol. 6, no. 3, pp. 200–203, 2020.

[33] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *Proc. of IEEE Conference on Decision and Control*, 2015, pp. 4708–4713.

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

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

[36] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *Proc. of IEEE International Conference on Robotics and Automation*, 2010, pp. 987–993.

[37] P. Bender, J. Ziegler, and C. Stiller, "Lanelets: Efficient map representation for autonomous driving," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2014, pp. 420–425.

[38] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, and M. Althoff, "Formalization of interstate traffic rules in temporal logic," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2020, pp. 752–759.

[39] C. Pek and M. Althoff, "Fail-safe motion planning for online verification of autonomous vehicles using convex optimization," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 798–814, 2021.

[40] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, p. 804–818, 2011.

[41] G. R. de Campos, F. D. Rossa, and A. Colombo, "Safety verification methods for human-driven vehicles at traffic intersections: Optimal driver-adaptive supervisory control," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 1, pp. 72–84, 2018.

[42] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD dataset: A drone dataset of naturalistic road user trajectories at German intersections," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2020, pp. 1929–1934.

[43] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.

[44] 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.

[45] S. Magdici and M. Althoff, "Fail-safe motion planning of autonomous vehicles," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 452–458.

[46] 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.