Formally Constrained Reinforcement Learning for Traffic Signal Control at Intersections

Oumaima Barhoumi
Department of Electrical and Computer Engineering
Concordia University
Montreal, Quebec, Canada
o_barh@ece.concordia.ca

Mohamed H Zaki
Department of Civil and Environmental Engineering
Western University
London, Ontario, Canada
mzaki9@uwo.ca

Sofiène Tahar
Department of Electrical and Computer Engineering
Concordia University
Montreal, Quebec, Canada
tahar@ece.concordia.ca

2025 IEEE International systems Conference (SysCon) | 979-8-3315-0818-0/25/$31.00 ©2025 IEEE | DOI: 10.1109/SYSCON64521.2025.11014841

***Abstract*—Ensuring the safety of autonomous systems is paramount for their successful integration into real-world transportation networks. Autonomous Vehicles and Machine Learning-driven traffic management systems have the potential to enhance efficiency and mobility. However, their deployment presents significant safety challenges, particularly in managing interactions between autonomous systems, human-driven vehicles, and pedestrians. This paper addresses these challenges by focusing on Reinforcement Learning (RL)-based traffic signal control. It analyzes traffic safety using Time-To-Collision to identify potential traffic conflicts between vehicle interactions that could jeopardize safety. We propose a novel approach to mitigate these conflicts by guiding the learning process under a formally checked safety constraint. This approach leverages Satisfiability Modulo Theories as a formal method for rigorous verification to ensure that the RL agent's decisions remain within safe operational boundaries, even in dynamic and unpredictable traffic conditions. Additionally, our approach incorporates dynamic speed adjustment mechanisms to address scenarios in which safety constraints are violated. Through traffic simulations, we evaluate the effectiveness of our approach in achieving a balance between traffic signal optimization and traffic safety by ensuring the safe operation of RL under a safety constraint and demonstrate that this adaptive speed control strategy reduces both the frequency and severity of traffic conflicts.**

***Index Terms*—Autonomous vehicles, satisfiability modulo theories, reinforcement learning, traffic safety, time-to-collision**

I. Introduction

The widespread integration of Machine Learning (ML) technologies in autonomous systems, particularly in transportation, has led to a growing reliance on ML-driven solutions to automate tasks, optimize operations, and enhance decision-making. However, with increased performance comes heightened complexity, impacting not only the efficiency and scalability of autonomous systems but also their safety, reliability, and overall trustworthiness. Ensuring the safety of autonomous systems in dynamic, real-world environments has become a critical challenge, as even small errors or failures in decision-making can have significant consequences. Effectively managing this complexity is now a central focus in the development of safe and reliable autonomous systems. In transportation, ML applications are evident in two key instances: developing Autonomous Vehicles (AVs) and enhancing infrastructure efficiency. This prompts the following question:

RQ: What are the implications of the deployment of AVs on road safety?

Road safety has always been the concern of researchers and car manufacturers, and it continues to be an ongoing endeavor demanding persistent focus and dedication. For instance, deploying AVs on the road and in mixed traffic, where they interact with conventional human-driven vehicles, introduces a new level of complexity that might affect the efficiency of traffic management methods to optimize the use of transportation infrastructure. From this perspective, the infrastructure must be enhanced to accommodate the introduced complexity, which brings us to the second instance of ML applications that enhance traffic infrastructure.

Intersections, as part of traffic infrastructure, are introduced to facilitate the smooth traffic flow. Traffic Signal Control (TSC) has emerged as a crucial strategy for tackling congestion, queuing, pedestrian safety, and coordinating traffic signals. [1]. Historically, the predominant approach to TSC has been fixed time control [2], where the timing of signal phases is predetermined and remains constant throughout the day. Although fixed-time control offers simplicity and predictability in signal operation, it may not be the most efficient solution for accommodating the dynamic behavior of AVs, necessitating substantial infrastructure upgrades. In this context, Reinforcement Learning (RL) is defined as a branch of ML where an agent learns to make sequential decisions by interacting with an environment to maximize cumulative rewards [3]. The application of RL in traffic signal control offers a dynamic and adaptive approach by learning optimal signal timing policies based on the feedback received from a traffic environment. RL-based TSC can potentially optimize traffic flow, reduce congestion, and improve overall transportation efficiency in urban areas. However, safety considerations need to be addressed for widespread adoption.

Studies by the National Highway Traffic Safety Administration (NHTSA) [4] state that conflicts arising in intersections are more likely to lead to crashes. For instance, according to Transport Canada's Road Safety report for 2011-2020 [5], 27% of fatalities occurred at intersections. Motivated by the recent concerns with AVs safety and the need to achieve their seamless and secure integration into modern transportation networks, one promising approach to enhancing confidence in transportation systems is to combine simulation-based tools with rigorous modeling and analysis techniques such as formal methods [6]. By modeling system behaviors, constraints, and requirements using formal languages and logical formulas, formal methods enable exhaustive analysis and verification of critical properties such as safety, correctness, and compliance with regulations.

As a popular formal methods tool, Satisfiability Modulo Theories (SMT) [7] is a type of constraint solving that extends traditional Boolean satisfiability (SAT) solving to handle constraints involving rich mathematical theories beyond propositional logic. In SMT solving, constraints are expressed using a combination of first-order logic formulas and specialized theories such as arithmetic and arrays. The key idea behind SMT solving is to decide the satisfiability of a logical formula with respect to a given theory. This involves determining whether values are assigned to variables that satisfy the formula while respecting the constraints imposed by the underlying theory. SMT solvers use efficient algorithms and decision procedures tailored to specific theories to solve these constraint satisfaction problems.

The lining of vehicles waiting to proceed through a signalized intersection leads to the formation of a queue [8]. Queuing is a common traffic event characterized by stationary or crawling vehicles, where the primary type of multi-vehicle crash is the rear-end collision. To ensure traffic safety during queuing, this paper involves studying traffic safety, specifically during the implementation of RL-based TSC at intersections. This entails analyzing how applying RL algorithms to control traffic signals impacts safety within a queue at an intersection. As a key aspect, the study focuses on analyzing traffic conflicts that occur during queuing at intersections, with a specific emphasis on using Time-To-Collision (TTC) as a primary indicator of their occurrence. In this context, we introduce a formal constraint-based approach to guide the RL process. We define the safety constraint as TTC greater than 3 seconds [9]. To ensure the consistency of this safety requirement, we propose using the Z3 SMT solver [10] for satisfiability checking [11]. Upon detecting violations of the constraint, we conduct a thorough analysis of both the severity and frequency of traffic conflicts by exploring different thresholds of TTC. Subsequently, we implement a speed adaptation process to dynamically adjust vehicle speeds in response to the severity of conflicts and the original speeds of the vehicles involved. This adjustment process operates concurrently with the ongoing RL process. This study aims to achieve a trade-off between traffic safety and RL-based TSC optimization by reducing the severity and the number of conflicts and optimizing the waiting time per lane.

The rest of the paper is structured as follows. Section II presents prior work on RL in traffic signal control. In Section III, we highlight the importance of traffic conflicts and TTC in traffic safety analysis and provide an overview of SMT. In Section IV, we provide an overview of the proposed methodology, followed by a case study in Section V to assess the safety constraint in a one-way traffic intersection scenario. Finally, Section VI summarizes the study's findings and discusses future research directions.

II. Related Work

Traditionally, traffic signal control operates based on fixed timing plans, often not adapting to real-time traffic conditions. However, machine learning introduces a data-driven approach to dynamically adjust signal timings based on observed traffic patterns. The advantages of employing machine learning for optimizing signal timings are recognized across various aspects. For instance, the main focus of the study in [12] is on reducing CO_2 emissions and fuel consumption by employing a Deep RL approach for traffic signal control. In [13], a case study in New York City was conducted using federated RL with the sole objective of reducing traffic congestion in urban areas. To enable more effective, scalable, and stable learning in complex adaptive traffic signal control environments, the authors of [14] employ an attention RL–based strategy. This approach aims to diminish computational complexity, stabilize the training process, and ultimately lead to reduced congestion levels and faster congestion recovery. The work of [15] aims to improve urban traffic under partially observable and noisy traffic information by employing Ontology-based Intelligent Traffic Signal Control.

While these works make valuable contributions to enhancing traffic efficiency by optimizing queue length, delay, and travel time, it is crucial to acknowledge that traffic safety, particularly during queuing, remains a significant challenge that requires careful consideration. In this context, some studies have addressed safety concerns using different methods. For instance, the authors of [16] used a safety performance function integrated into a RL algorithm, estimating conflicts by analyzing shockwaves and queue length. While their approach focuses on a macroscopic scale, treating vehicles as a singular moving entity and approximating changes in relative density and flow rates over time, it fails to consider the individual spatial and temporal positioning of each vehicle within the queue at a microscopic level. This finer-grained perspective could offer a more precise method for tackling safety concerns. Therefore, in this paper, we aim to address traffic safety issues both during queuing at intersections and throughout the RL process. Through analyzing vehicle-by-vehicle dynamics using TTC and speed analysis, we define a TTC-based safety constraint and check its satisfiability using formal methods.

The employment of formal methods in machine learning, particularly in reinforcement learning (RL) has been gaining traction in the last few years. For instance, in [17], the authors generate adversarial agents to exhibit flaws in the agent's policy by presenting moving adversaries. Subsequently, they employ reward shaping within the Q-Learning algorithm [18] to improve the learned defense policy. To assess the proficiency of their approach, the authors use the PRISM model checker [19] to verify the agent's reachability properties in four different scenarios. A similar approach using the PRISM model checker as well was presented in [20], where the cooperative multi-agent reinforcement learning (CMARL) agents are formally checked in an adversarial CMARL setting to guarantee that CMARL agents still comply with given safety requirements. In [21], the authors propose the verification of policies synthesized by RL to improve their safety. Their approach entails a formal encoding of the policy using probabilistic computational tree logic (PCTL) [22]. Based on this encoding, they calculate the probability of reaching unwanted states or of executing unwanted transitions to assess whether safety requirements hold.

Given their fully automated capabilities, SMT solvers were widely integrated by various works. In [23], the authors propose a formal approach to ensuring the safety of RL-based agents. To this end, they integrate RL, simulation and formal analysis to train a real safety-critical RL-agents to learn safe actions. By formulating safety properties as propositional logic formulas, they check whether the action in the given environment meets/violates these safety properties using SMT solvers. In [24], the authors propose an RL framework to execute autonomous driving tasks. Using Deep Q-Learning Networks (DQN) [25] and Deep Deterministic Policy Gradient (DDPG) [26], they train multiple RL agents on a generic set of driving maneuvers to learn specific maneuvers. These agents are then triggered once the maneuver learned and about to be executed is deemed to be safe. This is achieved by a structured program where safety specifications are defined as Linear Temporal Logic (LTL) formulas [27] and embedded as assertions. The verification process is carried by Nagini [28] with the Z3 SMT solver [10] as the back-end solver.

These works primarily use formal methods reactively, applying them in the final stages to validate RL policies. Although effective, this approach is limited as it verifies policy correctness only for scenario-specific properties rather than proactively guiding optimal policy learning. With safety now becoming a fundamental requirement in system design, a proactive approach is needed to integrate formal methods throughout the development and testing of AVs. In this paper, we embed formal verification early in the development process to systematically address safety risks, enhancing the reliability and trustworthiness of these advanced systems.

III. PRELIMINARIES

A. Time-To-Collision for Traffic Safety Assessment

Traffic conflicts, defined as "an observable situation in which two or more road users approach each other in space and time to the extent that there is a risk of collision if their movements remain unchanged" [29], play a crucial role in proactive road safety management systems. One key metric for analyzing such conflicts is Time to Collision (TTC), which measures the time remaining before two road users would collide if they maintain their current speeds and trajectories. By quantifying the imminence of potential collisions, TTC provides valuable insights into the severity of traffic conflicts and is widely used to assess safety levels and predict severe situations. According to a study conducted by Daimler-Benz in 1992 [30], it was found that providing drivers with a warning half a second before a rear-end collision could potentially prevent 60 percent of such incidents. Moreover, by extending the alert to a full second before the collision, up to 90 percent of collisions could potentially be avoided. For instance, TTC as a safety indicator in traffic flow has been widely studied and has been shown to be effective in identifying potential collisions.

TTC was first introduced by Hayward in 1971 as a temporal-proximity measure that predicts the time it would take for two vehicles to collide if no preventative measures are taken [31]. In a subsequent study by Hayward in 1972 [32], it was shown that TTC has an impact on the speed of the vehicle and can be used to prevent collisions by alerting drivers to potential hazards. Based on a study conducted in 1994 [33], TTC has been identified as the primary indicator used in designing collision avoidance systems. This highlights the importance of TTC in ensuring the safety of drivers and passengers on the road and underscores its significance as a critical safety indicator in traffic flow. In the study by Hirst et al. [9], a TTC of 4 seconds was initially identified as indicating a conflict situation for a vehicle. However, further analysis revealed that TTC values between 4 to 5 seconds sometimes resulted in false positives, suggesting potential collisions when a typical braking maneuver would resolve the situation safely. As a result, it was collectively decided that setting a TTC threshold at 3 seconds is more accurate.

B. Satisfiability Modulo Theories

Satisfiability Modulo Theories (SMT) [7], as an automated theorem-proving technique, combines classical propositional logic (SAT) with mathematical theories such as arithmetic, arrays, and bit-vectors. In SMT, problems are expressed as logical formulas involving variables and constraints. The goal is to find an assignment of values to the variables that satisfy the formula, considering the constraints imposed by the underlying theories. These formulas can include Boolean operators and operations from various mathematical theories, such as Boolean logic, linear integer arithmetic, or real-number arithmetic. SMT solvers leverage efficient algorithms to determine whether a given formula is satisfiable and, if so, to compute valid assignments for the variables. Z3 [10], developed by Microsoft Research, is one of the most widely used SMT solvers. Z3 takes as input logical formulas involving variables and constraints from different mathematical theories and finds satisfying value assignments to the variables that meet the given constraints. By applying a combination of techniques, including satisfiability solving algorithms for the Boolean parts of the formula and specialized decision procedures for different theories, Z3 efficiently searches for a solution. Its broad applicability makes it a powerful tool in formal verification, program analysis, and constraint solving across various domains. In this paper, we employ satisfiability checking to determine whether a given set of logical constraints is satisfiable, meaning that there exists an assignment of values to the variables that satisfies all the constraints. Satisfiability

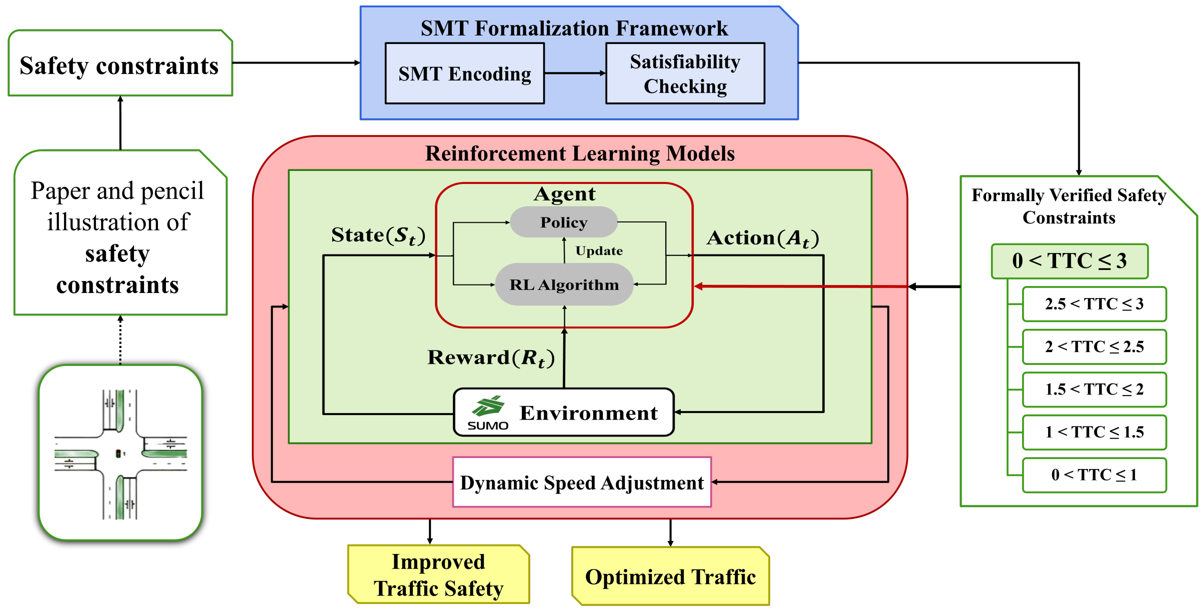

Fig. 1. Proposed Methodology for Formally-Constrained RL-based Traffic Signal Optimization

checking asks whether an interpretation (or assignment of truth values) exists to the variables of a formula such that the formula evaluates to true.

IV. PROPOSED METHODOLOGY

Traffic Signal Control (TSC) has emerged as a crucial strategy for tackling congestion, queuing, pedestrian safety, and coordinating traffic signals. [1]. The application of RL in traffic signal control offers an adaptive approach by learning optimal signal timing policies based on the feedback received from a traffic environment. In this study, we introduce a formally constrained RL process that aims to reach a compromise between optimizing traffic signal timings and ensuring traffic safety. This approach integrates formal constraints into the RL framework to guide decision-making, prioritizing safety and performance optimization. To achieve this, we exploit an existing RL implementation for a single intersection as described in [34] and leverage the integration of the traffic simulator SUMO with RL, i.e., SUMO-RL environment [34], to optimize traffic signal timings. Figure 1 illustrates the proposed methodology for the formally constrained RL model for traffic signal optimization, where traffic signal control is achieved through a Q-learning algorithm [18]. We start by identifying the safety conditions for the intersection by considering the vehicle dynamics as well as the registered values of TTC for each vehicle. By doing so, we establish the safety constraint for this traffic scenario. Using Z3, we validate the satisfiability of the established constraint, i.e., TTC $>$ 3 seconds. As shown by Figure 1, we consider the following sub-intervals: TTC $\in$]0,1],]1,1.5],]1.5,2],]2,2.5], and]2.5,3], to account for different levels of conflicts severity. Subsequently, the formal constraint is explicitly incorporated as a prerequisite for the learning process. The learning process proceeds only if vehicles entering the intersection adhere to the safety constraint. To perform a traffic safety analysis, we focus on a lane within the intersection where vehicles enter the queue, and monitor the vehicle dynamics at this stage. Consequently, we identify several conflicts with TTC values that fall between 0 and 3 seconds (excluding 0s). In order to reduce traffic conflicts occurrences as well as their severity, we initiate a speed adaptation mechanism to adjust vehicle speeds in order to increase TTC, which allows vehicles to better accommodate to the traffic flow ahead. The speed adjustments are determined by both the TTC value and the specific interval to which it belongs. By doing so, we achieve two objectives: first, ensuring safe vehicle interactions during intersection queueing and throughout the RL process, and second, we optimize traffic flow by reducing congestion and waiting time at intersections.

V. CASE STUDY

A. *Reinforcement Learning-based Traffic Signal Control*

To gain a comprehensive understanding of the research presented in this paper, we provide a brief description of the RL mechanism employed. We conduct our analysis over a signalized single intersection, which serves as the agent under training. As depicted by Figure 1, the traffic simulator SUMO is utilized as the learning environment. The RL technique implemented to optimize the signal timings is the Q-learning algorithm, where the agent observes the current state of the environment (signal phase) and selects an action (e.g., adjusting signal timings for different phases). The Q-learning algorithm updates its Q-values by incorporating observed rewards from the environment and the Q-value of the next state to represent the expected cumulative reward for specific actions in given states. In this context, the reward function computes the change in cumulative vehicle delay in relation to the previous time step, as given by Equation 1.

$$r_t = D_{a_t} - D_{a_{t+1}} \tag{1}$$

where D_{a_t} and $D_{a_{t+1}}$ refer to the total delay (sum of the waiting times of all approaching vehicles) at times t and $t+1$, respectively. By iteratively updating its Q-values, the agent converges to an optimal policy that minimizes traffic congestion and maximizes throughput at the intersection. This learning process is guided by the following key hyperparameters: the learning rate ($\alpha = 0.1$) determines how much new information influences the updates, the discount factor ($\gamma = 0.99$) prioritizes the long-term rewards, and epsilon ($\epsilon = 0.05$) balances the exploration and exploitation. For an in-depth explanation of the RL environment, please refer to [34].

B. Mathematical Formulation of Time-To-Collision

The mathematical expression for TTC, as provided in Equation 2, is defined for a generic number of vehicles.

$$TTC = \frac{x_i - x_{i+1} - L_i}{v_{i+1} - v_i}, \qquad v_{i+1} > v_i \tag{2}$$

where vehicles *i* and *i+1* are the leading and following vehicles, respectively, x_i, x_{i+1}, v_i and v_{i+1} are the positions and velocities of vehicles *i* and *i+1*, respectively, and *L* is the length of vehicle *i*. This TTC formulation provides a quantitative measure that considers both the distance between the vehicles and their relative velocities, thereby offering insight into the temporal proximity of a potential collision based on these factors.

C. Traffic Conflicts Analysis

In order to illustrate the occurrence of traffic conflicts, we monitor TTC values during the RL process to optimize signal timings at a single intersection. During this process, by analyzing TTC values, we identify traffic conflicts and extract the improved waiting time. Using SUMO, we first run the simulation to evaluate the improved waiting time per lane and compare it with the number of traffic conflicts that occur during the RL process. In this context, Figure 2 depicts the improved waiting time per lane versus the rising number of conflicts. While the RL process was effective in reducing the waiting time per lane compared to the original values, it overlooked safety at the individual vehicle level, leading to an increase in traffic conflicts. The recorded TTC values serve as indicators of the severity level of these conflicts. For instance, lower TTC values indicate that a collision is imminent, representing a high-severity conflict, as the time remaining before a potential collision is very short. Conversely, higher TTC values suggest that a collision is less likely to occur, representing a low-severity conflict, as there is more time available to avoid a collision, making the situation less critical. Considering the diverse range of TTC values, we deduce that traffic conflicts during the learning process had varying severity levels, spanning from very severe to less severe. Hence, in our study, we examine various thresholds

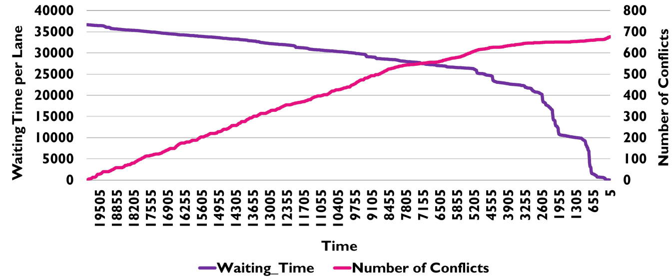

Fig. 2. Waiting Time per Lane versus Traffic Conflicts Occurrence

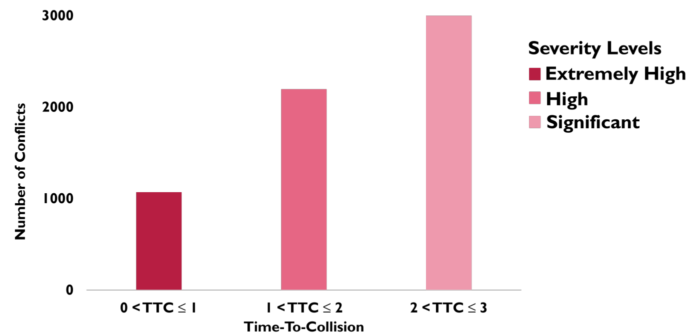

Fig. 3. Number of Conflicts and Severity Levels

of TTC to account for different levels of conflict severity. A closer examination of the TTC values for each vehicle with a TTC of less than 3 seconds reveals varying levels of conflict severity. To account for these differences, we focus on sub-intervals of TTC across three distinct intervals, i.e., $0 <$ TTC ≤ 1, $1 <$ TTC ≤ 2, and $2 <$ TTC ≤ 3. In Figure 3, the TTC distribution, extracted from the SUMO simulation, as well as the registered number of conflicts in each one are illustrated. We also consider three categories, namely *extremely high* ($0 <$ TTC ≤ 1), *high* ($1 <$ TTC ≤ 2) and *significant* ($2 <$ TTC ≤ 3).

D. Encoding of Safety Constraint using Z3

The safety constraint is encoded in Z3 following the steps in Algorithm 1. We begin by declaring variables of various types, such as integers and real numbers. Here, *N* represents the total number of vehicles within the flow. Subsequently, *x* and *v* represent the positions and velocities of the vehicles, respectively. In this context, the leading and following vehicles are identified by *i* and *j*, respectively. Their positions and velocities are denoted by *x_i*, *v_i*, *x_j*, and *v_j*, respectively. To ensure the validity and relevance of TTC calculation, the conditions defined by *cond* are essential. These conditions specify that the velocity of the following vehicle (*v_j*) must be greater than the velocity of the leading vehicle (*v_i*) and the position of the first vehicle (*x_i*) is greater than the position of the second vehicle (*x_j*).

In our encoding of TTC, we abstract the length of the vehicle *L* given in Equation 2. In the mathematical formulation of TTC, x_i represents the position of the leading vehicle at its front bumper. Therefore, we subtract the length L from x_i to accurately represent the distance between the vehicles. However, due to our integration with SUMO, the TTC values extracted from the simulation are determined from the rear-end position of the leading vehicle to the front-end position of the following vehicle. The safety constraint, i.e., *safety_constraint*, is defined to ensure a conflict-free traffic flow by establishing a threshold of 3 seconds. Consequently, we add the conditions, i.e., *cond*, along with the safety constraint, i.e., *safety_constraint*, to the solver that will attempt to find a solution that satisfies both constraints simultaneously, if such a solution exists. Once satisfied, we assign it as a constraint to the learning process, where every TTC value extracted from the simulation will then be checked against this constraint.

Algorithm 1 TTC-based Safety Constraint

```
N = Int('N') # Declare variables
N > 0
solver = Solver() # Create a solver instance
for i = 0 to N − 1 do # Define vehicles' dynamics
    x_i = Real('x'.format(i)) # position of leading vehicle i
    v_i = Real('v'.format(i)) # velocity of leading vehicle i
    x_j = Real('x'.format(i+1)) # position of following vehicle j
    v_j = Real('v'.format(i+1)) # velocity of following vehicle j
    cond = And(v_i >= 0, v_j >= 0, v_j > v_i, x_i > x_j)
    TTC = (x_i - x_j)/(v_j - v_i)
    safety_constraint = TTC > 3 # Safety constraint
    solver.add(cond) # Add constraints to the solver
    solver.add(safety_constraint) # Constraint must be satisfied
end for
if solver.check() == sat then
    # Constraint is satisfiable
end if
```

E. Speed Adaptation Process

For the RL process to advance, it is crucial to satisfy the safety constraint, i.e., TTC $>$ 3 seconds. In situations where this constraint is not upheld, i.e., TTC $\leq$ 3 seconds, we propose a speed adaptation process that modifies the speeds of vehicles with a TTC less than 3 seconds. However, considering the severity level analysis outlined in Section V-C, it is imperative to account for the varying severity levels when implementing the speed adjustment. For example, conflicts classified as extremely highly severe require a different approach than those categorized as highly severe. As an initial step, we identify the stopping distance at which vehicles can safely halt while maintaining a sufficient gap to ensure a TTC of at least 3 seconds. In this context, we use a Python method *getStopSpeed(self, vehID, speed, gap)* [35] that returns the speed for stopping at the defined gap. To assign the stopping speed, we use the Python method *slowDown(self, vehID, speed, duration)* [35] that changes the speed smoothly to the given value over the specified amount of time in seconds. As a result, the stopping speed adjustment is achieved gradually over a specified duration and based on the severity of the conflicts.

F. Results and Discussion

a) Results: With the safety constraints in place, we run the SUMO simulation, incorporating the RL process along with the speed adjustments in the loop. In Figure 4, the number of conflicts computed for TTC $<$ 3 seconds, *before* and *after* the speed adaption process is employed, is depicted. The analysis considers the sub-intervals of TTC $<$ 3 seconds to account for different severity levels of the conflicts. As shown by Figure 4, the speed adjustment resulted in a reduction of the number of conflicts. For instance, for TTC $<$ 1 second, the registered decrease is by 28.5%, thereby minimizing the severity level of the conflicts. However, this decrease came at the expense of increasing the waiting time by 18.6%, as given in Figure 5, representing the waiting time *before* and *after* the speed update for TTC $\in$ [0,1] seconds. Similarly, for TTC $\in$ [1,1.5] and [1.5,2] seconds, the number of conflicts decreases by 42.9% and 33.6%, respectively, causing a remarkable decrease of the severity levels. Figure 6 represents the waiting time per lane before and after the speed adjustment for 1 $<$ TTC $<$ 2 seconds, showing an increase by 6.4% as a consequence of the speed adjustment. Figure 7 depicts the waiting time per lane *before* and *after* the speed adjustment for TTC $\in$ [2,3] seconds. In the case of TTC $\in$ [2,2.5] and [2.5,3] seconds, we register an increase in the number of conflicts, as shown by Figure 4, leading to an improvement in the waiting time per lane as reflected by Figure 7. The increase of traffic conflicts registered in this case is a consequence of reducing the severity of the conflicts reflected by TTC $\in$ [1,1.5] and [1.5,2] seconds. This reduction is translated by an increase in TTC values, which results in a revised value exceeding the previous one, thus no longer in the categories of TTC within [1,1.5] and [1.5,2] seconds. These results show that the speed adjustment effectively reduces both the frequency and severity of conflicts at an intersection queue during the learning phase. However, this enhancement comes with the trade-off of potentially compromising the efficiency of the Q-Learning algorithm in achieving an optimal outcome. This is shown by Figures 5 and 6, where the waiting time per lane has increased as a result of our methodology. Nevertheless, this increase remains acceptable, given our commitment to ensuring traffic safety throughout the process.

b) Discussion: To determine the required duration for assigning the stopping speed and evaluate its impact on traffic flow, we conducted a speed analysis of vehicles with TTC less than 3 seconds. Throughout this analysis, it became apparent that the adjusted time-to-collision (TTC) value for vehicles

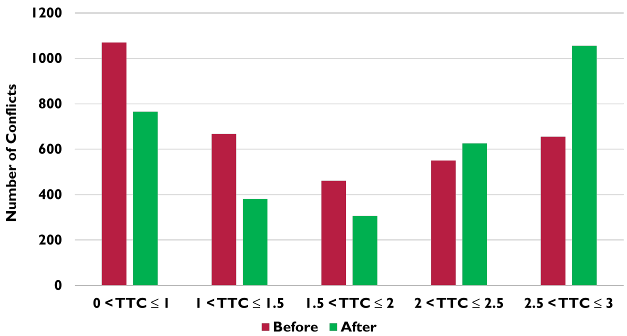

Fig. 4. Number of Conflicts Before and After the Speed Adjustment for a Profiling of TTC $\leq$ 3 seconds

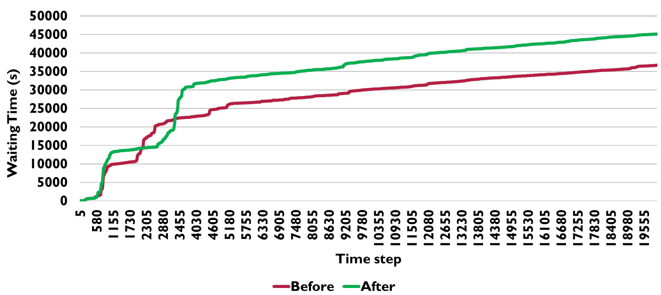

Fig. 5. Waiting Time per Lane Before and After the Speed Adjustment for $0 < \text{TTC} \leq 1$ seconds

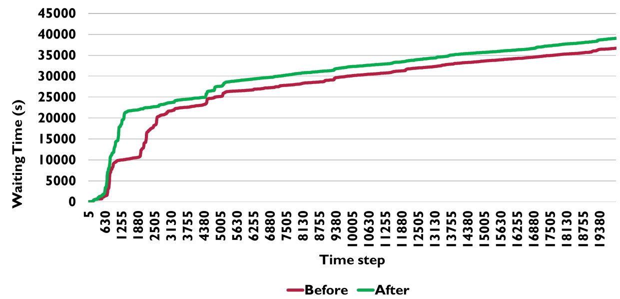

Fig. 6. Waiting Time per Lane Before and After the Speed Adjustment for $1 < \text{TTC} \leq 2$ seconds

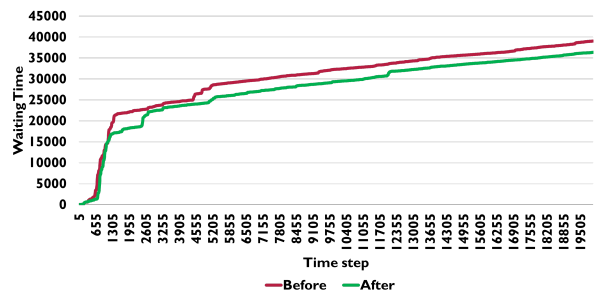

Fig. 7. Waiting Time per Lane Before and After the Speed Adjustment for $2 < \text{TTC} \leq 3$ seconds

initially with TTC less than 1 second deteriorates whenever the change duration is a one-time step or more. On the contrary, an instantaneous change, where the duration is equal to 0-time steps, enhanced the TTC and consequently reduced the severity level. We computed the delay for each vehicle to determine the duration for vehicles with TTC less than 2 and 3 seconds. This delay is the difference in time between when the vehicle entered the intersection and when it exited. This duration enables vehicles to begin braking gradually, thereby increasing the gap with following or leading vehicles until they come to a complete stop or accelerate if the light is green.

The speed adaptation process this study uses is a centralized computation that analyzes TTC values and dynamically adjust vehicle speeds. By leveraging TTC and traffic flow data, the system determines optimal speed adjustments in real time. This practical solution works with existing sensor technologies, eliminating the need for additional infrastructure. In contrast, Vehicle-to-Vehicle (V2V)-based systems [36] require specialized hardware and widespread adoption. They also face communication delays from network latency, packet loss, or interference, which can compromise real-time safety decisions.

VI. Conclusion

To bolster trust in autonomous systems, ensuring safety during their deployment is a critical prerequisite. This study focuses on traffic safety at an intersection where Reinforcement Learning (RL) is used to optimize traffic signal timings. We propose a formally verified safety constraint that guides the RL process, ensuring that the RL agent consistently operates within safe temporal boundaries. To achieve this, we use metrics such as Time-to-Collision (TTC) and set the safety threshold at TTC greater than 3 seconds. This constraint ensures that vehicles have adequate time to react and maneuver safely within the intersection, mitigating the risk of collisions. To confirm the validity of this constraint, we employed the Z3 SMT Solver for satisfiability checking. Once the safety constraint is verified, the learning process continues with TTC values monitored against the constraint using SUMO. If the safety constraint is unmet, a safety analysis is conducted using interval profiling of TTC values below 3 seconds to assess the severity and frequency of intersection conflicts. Based on this analysis, a speed adaptation process is implemented to dynamically adjust vehicle speeds while considering conflict severity, allowing the RL algorithm to continue its operation. The findings of this study demonstrate the feasibility as well as the efficiency of the proposed approach in the case of traffic intersections. The integration of formal constraints to guide RL models showcased a significant improvement in decision-making, reducing both the frequency and severity of traffic conflicts, while improving the traffic flow.

To further study the efficiency of the proposed approach, in an immediate future work, we aim to validate it over a two-way intersection. Additionally, to ensure traffic safety at intersections, we plan to formalize an intersection scenario using Z3 and establish a formal traffic safety rule, defined in [37], as a constraint. This rule combines metrics such as Time-to-Collision, space headway, and shockwave speed for an accurate safety analysis. In the pursuit of achieving trustworthy AVs, we aim to embed formal safety constraints directly into the objective function of Reinforcement Learning algorithms to ensure that safety becomes an inherent aspect of ML algorithms.

REFERENCES

[1] Y. Zhao and Z. Tian, "An overview of the usage of adaptive signal control system in the united states of america," *Applied Mechanics and Materials*, vol. 178, pp. 2591–2598, 2012.

[2] A. Muralidharan, R. Pedarsani, and P. Varaiya, "Analysis of fixed-time control," *Transportation Research Part B: Methodological*, vol. 73, pp. 81–90, 2015.

[3] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, Learning, and Optimization*, vol. 12, no. 3, p. 729, 2012.

[4] S. E. Lee, E. Llaneras, S. Klauer, and J. Sudweeks, "Analyses of rear-end crashes and near-crashes in the 100-car naturalistic driving study to support rear-signaling countermeasure development," *DOT HS*, vol. 810, no. 846, pp. 1–125, 2007.

[5] "Road safety in Canada 2020." [Online]. Available: https://tc.canada.ca/sites/default/files/2023-01/Road_Safety_in_Canada_2020_EN.pdf

[6] O. Hasan and S. Tahar, "Formal verification methods," in *Encyclopedia of Information Science and Technology*. IGI Global, 2015, pp. 7162–7170.

[7] N. Bjørner, L. de Moura, L. Nachmanson, and C. M. Wintersteiger, "Programming Z3," *International School on Engineering Trustworthy Software Systems*, pp. 148–201, 2019.

[8] G. Stephanopoulos, P. G. Michalopoulos, and G. Stephanopoulos, "Modelling and analysis of traffic queue dynamics at signalized intersections," *Transportation Research Part A: General*, vol. 13, no. 5, pp. 295–307, 1979.

[9] S. Hirst and R. Graham, "The format and presentation of collision warnings," in *Ergonomics and safety of intelligent driver interfaces*. CRC Press, 2020, pp. 203–219.

[10] L. De Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Tools and Algorithms for the Construction and Analysis of Systems*, vol. 4963 of LNCS. Springer, 2008, pp. 337–340.

[11] E. Ábrahám and G. Kremer, "Satisfiability checking: Theory and applications," in *Software Engineering and Formal Methods Proceedings 14*, vol. 9763 of LNCS. Springer, 2016, pp. 9–23.

[12] H. Kamal, W. Yánez, S. Hassan, and D. Sobhy, "Digital-twin-based deep reinforcement learning approach for adaptive traffic signal control," *Internet of Things Journal*, vol. 11, no. 12, pp. 21 946–21 953, 2024.

[13] Y. Fu and X. Di, "Federated reinforcement learning for adaptive traffic signal control: A case study in new york city," in *Intelligent Transportation Systems*. IEEE, 2023, pp. 5738–5743.

[14] G. Han, X. Liu, H. Wang, C. Dong, and Y. Han, "An attention reinforcement learning–based strategy for large-scale adaptive traffic signal control system," *Journal of Transportation Engineering, Part A: Systems*, vol. 150, no. 3, pp. 04 024 001–1–04 024 001–12, 2024.

[15] J. Guo, S. Ghanadbashi, S. Wang, and F. Golpayegani, "Urban traffic signal control at the edge: An ontology-enhanced deep reinforcement learning approach," in *Intelligent Transportation Systems*. IEEE, 2023, pp. 6027–6033.

[16] T. Ghoul and T. Sayed, "Real-time signal-vehicle coupled control: An application of connected vehicle data to improve intersection safety," *Accident Analysis & Prevention*, vol. 162, pp. 106–389, 2021.

[17] P. S. N. Mindom, A. Nikanjam, F. Khomh, and J. Mullins, "On assessing the safety of reinforcement learning algorithms using formal methods," in *Software Quality, Reliability and Security*. IEEE, 2021, pp. 260–269.

[18] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[19] M. F. Fels *et al.*, "PRISM: An introduction," *Energy and Buildings*, vol. 9, no. 1-2, pp. 5–18, 1986.

[20] D. Gross, C. Schmidl, N. Jansen, and G. A. Pérez, "Model checking for adversarial multi-agent reinforcement learning with reactive defense methods," in *Automated Planning and Scheduling*, vol. 33, no. 1, 2023, pp. 162–170.

[21] S. Pathak, L. Pulina, and A. Tacchella, "Verification and repair of control policies for safe reinforcement learning," *Applied Intelligence*, vol. 48, pp. 886–908, 2018.

[22] F. Ciesinski and M. Größer, "On probabilistic computation tree logic," *Validation of Stochastic Systems: A Guide to Current Research*, pp. 147–188, 2004.

[23] A. Murugesan, M. Moghadamfalahi, and A. Chattopadhyay, "Formal methods assisted training of safe reinforcement learning agents," in *NASA Formal Methods*, vol. 11460 of LNCS. Springer, 2019, pp. 333–340.

[24] B. Gangopadhyay, H. Soora, and P. Dasgupta, "Hierarchical program-triggered reinforcement learning agents for automated driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 902–10 911, 2021.

[25] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep q-learning," in *Learning for Dynamics and Control*, vol. 120. Proceedings of Machine Learning Research, 2020, pp. 486–489.

[26] H. Tan, "Reinforcement learning with deep deterministic policy gradient," in *Artificial Intelligence, Big Data and Algorithms*. IEEE, 2021, pp. 82–85.

[27] K. Y. Rozier, "Linear temporal logic symbolic model checking," *Computer Science Review*, vol. 5, no. 2, pp. 163–203, 2011.

[28] M. Eilers and P. Müller, "Nagini: a static verifier for python," in *Computer Aided Verification*, vol. 10981 of LNCS. Springer, 2018, pp. 596–603.

[29] F. Amundsen and C. Hydén, "Proceedings of the first workshop on traffic conflicts: Oslo 77." Norwegian Council for Scientific and Industrial Research, 1977.

[30] D. R. Ankrum, "Ivhs-smart vehicles, smart roads," *Traffic Safety (Chicago)*, vol. 92, no. 3, 1992.

[31] J. Hayward, *Near misses as a measure of safety at urban intersections*. Pennsylvania Transportation and Traffic Safety Center, USA, 1971.

[32] ——, "Near miss determination through use of a scale of danger," *Highway Research Record*, vol. 384, pp. 24–34, 1972.

[33] R. Van der Horst and J. Hogema, "Time-to-collision and collision avoidance systems," *In International Cooperation on Theories and Concepts in Traffic Safety*, p. 109–121, 1993.

[34] L., Alegre, "SUMO-RL." https://github.com/LucasAlegre/sumo-rl, 2019.

[35] German Aerospace Center, "Vehicle Value Retrieval in Traci," https://sumo.dlr.de/docs/TraCI/Vehicle_Value_Retrieval.html, 2024.

[36] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, J. Wang *et al.*, "Vehicle-to-vehicle communications: readiness of V2V technology for application." United States. National Highway Traffic Safety Administration, Tech. Rep., 2014.

[37] O. Barhoumi, M. H. Zaki, and S. Tahar, "A formal approach to road safety assessment using traffic conflict techniques," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 606–619, 2024.