2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Learning Terminal State of the Trajectory Planner: Application for Collision Scenarios of Autonomous Vehicles

Joonhee Lim[1], Kibeom Lee[2], Jangho Shin[3], and Dongsuk Kum[4]

Abstract— **Collision Avoidance/Mitigation System (CAMS) for autonomous vehicles is a crucial technology that ensures the safety and reliability of autonomous driving systems. Conventional collision avoidance approaches struggle in complex and various scenarios by avoiding collisions based on rules for specific collision scenarios. This has led to learning-based methods using neural networks for adaptive collision avoidance. However, the approaches directly outputting control inputs through neural networks have drawbacks in interpretability and stability. To address these limitations, we propose a trajectory planning method for CAMS that combines deep reinforcement learning (DRL) and quintic polynomial (QP) trajectory planning. The proposed method determines the terminal state and confidence of the trajectory using DRL and plans a QP trajectory based on them. By utilizing the terminal state and confidence of the trajectory rather than direct control inputs as the output of the neural network, it generates a more realistic and continuous path. Moreover, this approach considers collision avoidance and mitigation in an integrated manner through the reward function of RL. Our experimental results demonstrate that the proposed method not only improves interpretability and stability compared to existing learning-based methods but also upholds performance in complex and various collision scenarios.**

I. Introduction

Vehicle safety systems have been playing a significant role in enhancing road traffic safety [1]. Among these vehicle safety systems, the Collision Avoidance/Mitigation System (CAMS) in the active safety system is instrumental in minimizing the damage to drivers and passengers by overriding the action of human drivers or automated driving algorithms to avoid and mitigate collisions in imminent collision situations [2].

Conventional motion planning methods for collision avoidance operate based on rules for specific scenarios [3]–[5]. Such collision avoidance approaches have limitations in handling a wide range of collision scenarios due to their heavy reliance on predetermined rules and decision algorithms [6], [7]. Hence, recent research has been actively conducted on optimization-based and learning-based collision avoidance and mitigation methods that can handle a wide range of collision scenarios.

*This work was supported by a research grant from Hyundai Motor Company.

[1]Robotics Program, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea kingear3@kaist.ac.kr

[2]Department of Mechanical Engineering, Gachon University, Seongnam, Republic of Korea kibeom.lee@gachon.ac.kr

[3]Research & Development Division, Hyundai Motor Company, Hwaseong, Republic of Korea jangho_shin@hyundai.com

[4]The Cho Chun Shik Graduate School of Mobility, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea dskum@kaist.ac.kr

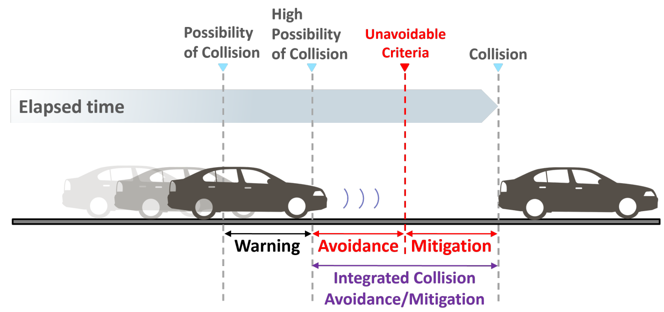


Fig. 1: Functionality of an active safety system over time. During the time between a high possibility of collision and collision, existing CAMS operates separately based on the criteria, and integrated CAMS operates comprehensively.

The optimization-based collision avoidance and mitigation method [2] has the advantage of comprehensively assessing the risks of surrounding vehicles in all directions through Predictive Occupancy Map (POM). By leveraging the POM, it can effectively handle complex collision scenarios involving multiple vehicles. Guardini et al. [8] address complex and risky collision situations by considering all binary collision risks of multiple vehicles in the scene through Probability of Collision with Injury Risk (PCIR). However, most of these optimization-based methods still have limitations when it comes to scenarios where the trajectories of surrounding vehicles are variable, as they are designed for scenarios where surrounding vehicles have constant speed, acceleration, and path.

On the other hand, by learning from diverse collision scenarios, deep reinforcement learning (DRL) based methods [9], [10] can consider not only complex collision scenarios but also various collision scenarios where surrounding vehicles have a wide range of speeds, accelerations, and paths. However, they only consider collision avoidance, excluding collision mitigation even when it cannot be avoided. In contrast, several works focus only on minimizing collision severity through optimization when collision avoidance is impossible [11], [12]. Liu et al. [13] also consider only collision avoidance based on the exclusive area of the relative velocity vector in complex collision situations. These separated approaches have a severe problem: they can make inappropriate decisions on avoidance and mitigation. For example, only collision mitigation is considered even when collision avoidance is possible, and only collision avoidance is considered when collision avoidance is impossible.

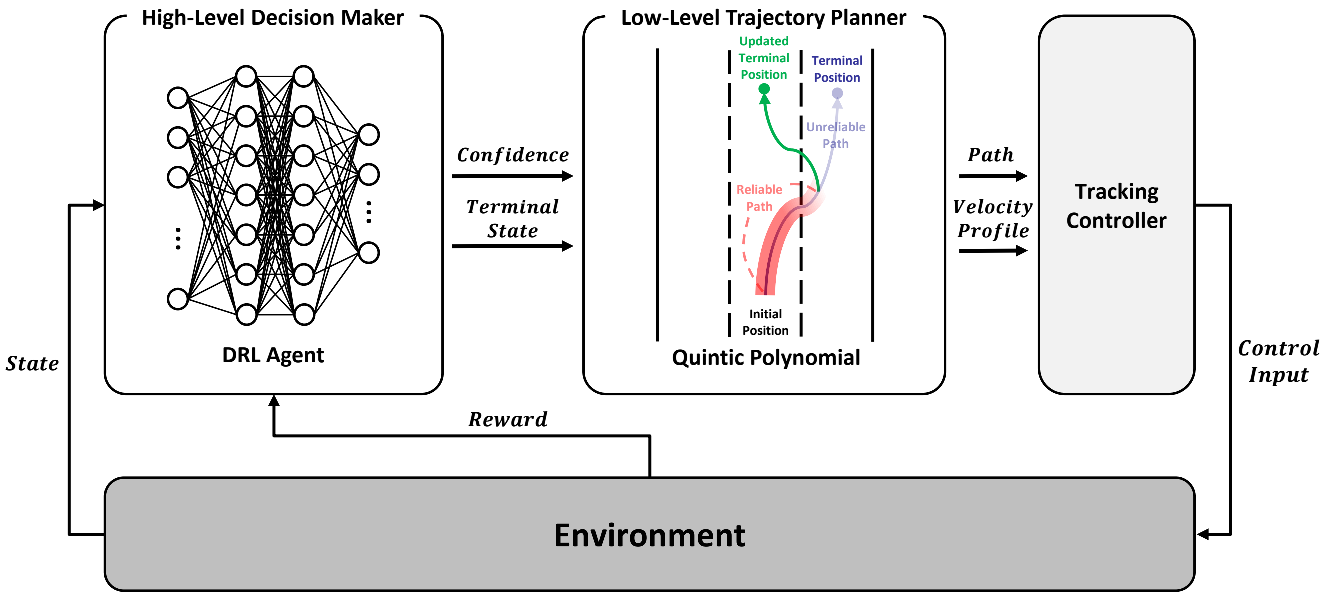


Fig. 2: Overall framework of the proposed method.

In consideration of both collision avoidance and mitigation, the methods using the time-to-collision (TTC) to determine the unavoidable criteria were devised [14], [15]. Nonetheless, the deterministic separation between collision avoidance and mitigation, as highlighted in red in Fig. 1, implies that while the problem is alleviated, it is not fully resolved.

There were efforts to address collision avoidance and mitigation in an integrated manner, as depicted in purple in Fig. 1, through the reward function of RL [16], [17]. However, they use the neural network's output as direct control inputs, such as steering, making it difficult or even impossible to interpret the system's behavior [18]. Since the paths are outputted directly as a black box through the neural network, they lack realism and continuity, which may compromise safety and reliability. Lastly, Wang et al. [19] comprehensively address collision avoidance and mitigation by optimizing the Potential Crash Severity Index (PCSI) and Artificial Potential Field (APF). However, it still retains the scenario adaptation limitations of the optimization-based approaches.

To address the limitations, we propose a trajectory planning approach for CAMS that combines DRL and quintic polynomial (QP) trajectory planning. The proposed method handles complex imminent collision scenarios of surrounding vehicles with various positions, speeds, and trajectories through DRL. It generates a more realistic and continuous path by utilizing the terminal state and confidence of the QP trajectory rather than direct control inputs as the output of DRL. Finally, it considers collision avoidance and mitigation in an integrated manner through the reward function of DRL. Experiments for validation are conducted in the CARLA simulator [20], and our experimental results demonstrate that the proposed method achieves collision avoidance performance comparable to existing learning-based methods while alleviating their interpretability and stability issues. The main contributions of this paper can be summarized below:

- By utilizing DRL, handle complex and various collision scenarios.
- By combining DRL with QP trajectory planning, alleviate the problems of interpretability and stability.
- By addressing collision avoidance and mitigation in an integrated manner via the reward function of DRL, find the optimal criteria for collision avoidance and mitigation.

The remainder of this paper is organized as follows: In Section II, Detailed explanation of the proposed algorithm is provided. Section III presents the simulation environment and experimental results. Finally, Section IV offers a summary and conclusion of our work.

II. Methodology

A. Overview

The proposed method consists of two main components: the high-level decision maker, composed of a DRL agent, and the low-level trajectory planner, which utilizes QP trajectory planning, as depicted in Fig. 2.

The high-level decision maker determines the terminal state and confidence for a QP trajectory, considering collision avoidance and mitigation in an integrated manner. On the other hand, the low-level trajectory planner utilizes the terminal state from the high-level decision maker to plan a QP trajectory (Blue in Fig. 2), and the confidence to clip the trajectory only up to a reliable point (Red in Fig. 2). It then re-plans the trajectory (Green in Fig. 2) for an updated terminal state. A tracking controller converts the planned trajectories into vehicle control inputs. This process is iteratively performed in various imminent collision scenarios, allowing the DRL agent to learn appropriate high-level decisions about collision avoidance and mitigation.

B. High-Level Decision Maker

The high-level decision maker of the proposed algorithm is composed of a DRL agent. We use the Soft Actor-Critic [21] for the DRL algorithm, which is suitable for continuous action spaces like autonomous driving. We define the problem of determining high-level decisions for collision avoidance and mitigation as a Markov Decision Process (MDP) problem. MDP consists of tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is set of actions, $\mathcal{P} = P(S_{t+1} = s'|S_t = s, A_t = a)$ is the transition probability that the next state will be s' when the current state is s and the current action is a, $\mathcal{R} = E[R_{t+1}|S_t = s, A_t = a]$ is reward function and γ is a discount factor, $\gamma \in [0, 1]$.

The SAC algorithm is a model-free RL algorithm, which means that the agent does not have prior knowledge of the transition probabilities and solves the problem without explicit information about them. Additionally, it incorporates the concepts of maximum entropy and RL to promote diverse action exploration and enhance its robustness. In the problem formulation, the state space, action space, and reward function are structured as follows.

1) State Space: As inputs to the DRL agent, the following information is used:

- Global X, Y speed and yaw angle of the ego vehicle
- Relative X, Y position/speed and yaw angle of surrounding vehicles
- Lane information of ego and surrounding vehicles

The number of surrounding vehicles can be adjusted according to the scenario requirements. In our case, we assume three vehicles are positioned around the ego vehicle.

2) Action Space: The outputs of the DRL agent are as follows:

- R: Longitudinal distance to the terminal state of QP trajectory
- L: Lateral distance to the terminal state of QP trajectory
- v_T: Velocity at the terminal state of QP trajectory
- T: Time to reach the terminal state of QP trajectory
- p: Confidence of QP trajectory

L is selected from lateral distances for lane keep and left/right lane change to generate lane-based collision avoidance and mitigation paths. These decisions regarding lateral movement are made within the road space, excluding sidewalks where vehicles should not travel. The range of the action is set to plan short trajectories between 1 and 2 seconds, suitable for imminent collision avoidance and mitigation. How these actions of the DRL agent are combined with the low-level trajectory planner will be further explained in detail within the low-level trajectory planner's context.

3) Reward Function: The reward function is designed as:

$$R_{\text{total}} = w_{\text{collision}} R_{\text{collision}} + w_{\text{risk}} R_{\text{risk}} \quad (1)$$

$$R_{\text{collision}} = \begin{cases} -\text{Collision Impact} & \text{if collision occurs} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$R_{\text{risk}} = -\max_{i \in \{1,2,\ldots,n\}} \text{Risk}_{veh_i} \ \text{(every time step)} \quad (3)$$

$$\text{Risk}_{veh_i} = \begin{cases} \text{clip}(\text{TTC}_i^{-1}, 0, 4) & \text{if TTC} > 0 \text{ seconds} \\ 5 & \text{if TTC} = 0 \text{ seconds} \end{cases} \quad (4)$$

where i denotes the indices of n surrounding vehicles.

By imposing a penalty proportional to the collision impact, the algorithm learns to avoid actions leading to collisions and generate the trajectories to minimize collision impact if a collision is unavoidable. This penalty allows for a comprehensive consideration of collision avoidance and mitigation strategies.

Even without actual collisions, we enhance safety by utilizing the inverse of TTC as a risk. The constant turn rate & velocity (CTRV) model [22] is employed to compute the TTC. This model is well-suited for predicting imminent collision scenarios due to its computational efficiency and validity for short time horizons [23]. The TTC-based risk for each surrounding vehicle is clipped as in Eq. (4) to prevent the risk from diverging infinitely [2]. Then, the negative of the maximum value among the risks of the surrounding vehicles is used as R_{risk}.

C. Low-Level Trajectory Planner

In the low-level trajectory planner, QP trajectories are planned based on the terminal states determined by the high-level decision maker. Then, based on the confidence outputted through the neural network, the trajectory is clipped only up to the highly reliable parts, as illustrated in Fig. 2.

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (5)$$

$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 \quad (6)$$

The QP trajectory planning refers to determining the six coefficients of the 5th-order polynomial [24] to define the desired trajectory over time for both the x and y-axis. The polynomial-based motion planning methods are well-suited for lane-based path planning of autonomous vehicles. They can satisfy constraints such as the required position, yaw angle, and velocity at interpolating points [25].

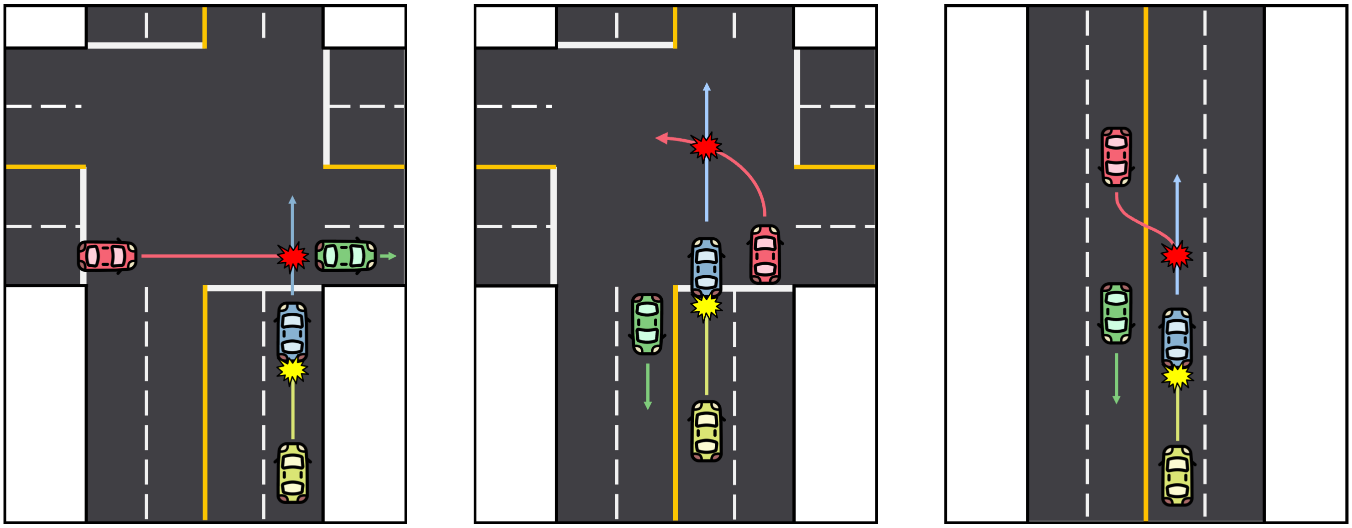
$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & T & T^2 & T^3 & T^4 & T^5 \\ 0 & 1 & 2T & 3T^2 & 4T^3 & 5T^4 \\ 0 & 0 & 2 & 6T & 12T^2 & 20T^3 \end{bmatrix}^{-1} \begin{bmatrix} x(0) \\ \dot{x}(0) \\ \ddot{x}(0) \\ x(T) \\ \dot{x}(T) \\ \ddot{x}(T) \end{bmatrix} \quad (7)$$

with $a_0, a_1, \ldots, a_5 \in \mathbb{R}$.

$$\begin{cases} x(T) &= x_{\text{ego}} + R \\ \dot{x}(T) &= v_T \\ \ddot{x}(T) &= 0 \end{cases} \qquad \begin{cases} y(T) &= y_{\text{ego}} + L \\ \dot{y}(T) &= 0 \\ \ddot{y}(T) &= 0 \end{cases} \quad (8)$$

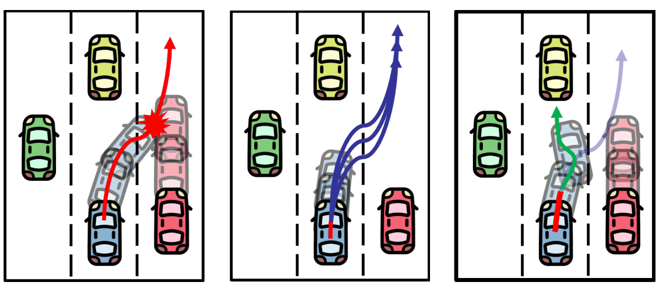
where x_{ego}, y_{ego} represent the current positions of the ego vehicle along the x and y-axis, respectively. The values of R, L, v_T and T are determined through DRL.

To compute the coefficients of the QP trajectory for the x-axis, it is necessary to determine T and the boundary conditions at T seconds, as indicated in Eq. (7-8). These terminal states of the trajectory play a crucial role in determining which lane to move in for collision avoidance and mitigation,



(a) (b) (c)

Fig. 3: Target imminent collision scenarios. (a) Fast approach of the vehicle ignoring traffic lights at the intersection. (b) Abrupt lane intrusion of the vehicle ignoring traffic lights at the intersection. (c) Sudden lane intrusion of the vehicle from the opposing lane on the straight road.



(a) Full Tracking (b) Init Tracking (c) Confidence based

Fig. 4: Planned trajectory based on the tracking level.

as well as generating the path's shape. The equation for the y-axis follows the same form as Eq. (7).

By outputting the terminal state R, L, v_T, T of the QP instead of control inputs like steering, it generates a realistic and continuous path while handling complex and various collision scenarios. In addition, yaw constraint is used for the planned paths to prevent unrealistic path generation.

Coupling the path and velocity planning has the advantage of finding a better solution in which path and speed interact strongly, even in emergency collision situations. However, if this coupled trajectory is tracked without re-planning until the terminal state, sudden movements of surrounding vehicles cannot be considered, as illustrated in Fig. 4a. The approach of tracking only the initial segment of the trajectory and then re-planning the trajectory, as depicted in Fig. 4b, also has a limitation. This is due to the nature of the polynomial-based method, where the initial portion of the planned path features very gradual changes in yaw angle and speed. Consequently, reaching the target point takes considerable time, making it challenging to avoid collision. Thus, we further output confidence p through the neural network to determine how much the planned trajectory should be trusted and followed. This improves the algorithm by clipping and tracking the trajectory only to the reliable point and re-planning the trajectory, as shown in Fig. 4c.

TABLE I: DRL Hyperparameters.

Parameter	Value
optimizer	Adam [26]
learning rate	0.0003
discount factor	0.99
batch size	64
buffer size	10^6
number of hidden layers	2
number of hidden units per layer	256

III. Experiments and Results

The proposed trajectory planning method is trained and validated in the CARLA simulator [20]. For the DRL agent to learn to make appropriate decisions about collision avoidance and mitigation, learning is conducted at collision imminence levels divided from 1 to 3, ranging from collision situations when collision avoidance is possible to situations when collision avoidance is impossible. As the collision imminence level increases, the distance between the risky and the ego vehicle becomes closer, and the speed at which the risky vehicle approaches the ego vehicle increases. The hyperparameters of the DRL agent used for training are shown in Table I.

Based on the Fatality Analysis Reporting System (FARS) and National Automotive Sampling System General Estimates System (NASS-GES), about 40 percent of crashes that occurred in the United States occurred at intersections [27]. Therefore, two collision scenarios that occur at intersections and one collision scenario that occurs on a general straight road are used as target scenarios, as shown in Fig. 3. In

TABLE II: Collision rate & Average collision impact in various collision imminence levels.

Evaluation Metric	Collision Imminence Level	Algorithm			
		TTC-based AEB	End-to-End RL	Ours (Only Avoidance)	Ours (Integrated)
Collision Rate [%]	1	39.2%	3.5%	3.4%	**3.3%**
	2	51.1%	9.5%	14.1%	10.2%
	3	75.9%	28.2%	25.3%	27.2%
	All	54.7%	13.3%	13.1%	**12.9%**
Avg. Collision Impact [N·s]	1	7989 N·s	5678 N·s	5412 N·s	**4773 N·s**
	2	8683 N·s	6721 N·s	7019 N·s	**6450 N·s**
	3	7845 N·s	7901 N·s	7650 N·s	**6644 N·s**
	All	8056 N·s	7338 N·s	7260 N·s	**5793 N·s**

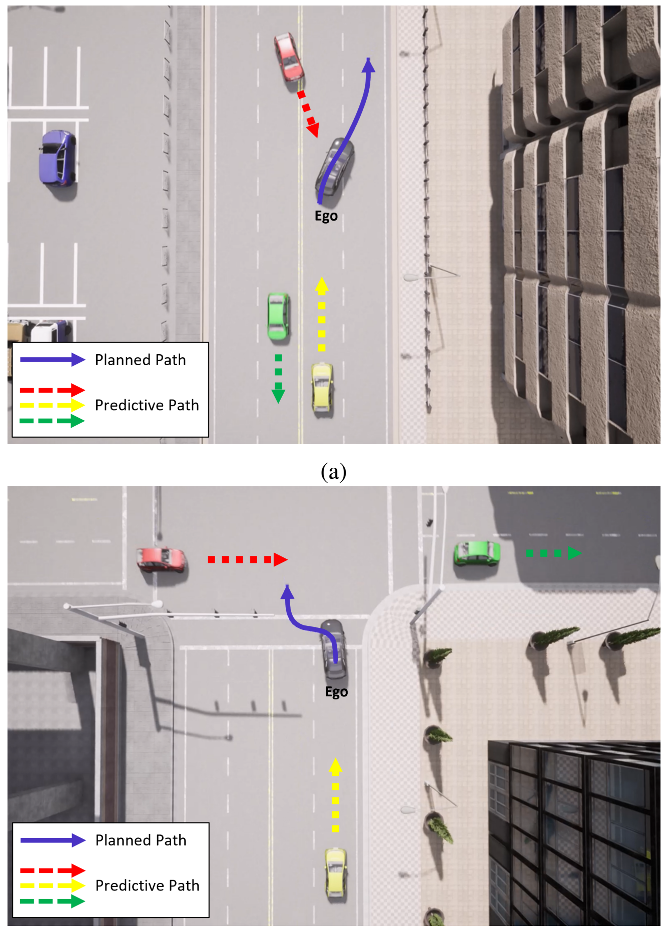
scenario (a), the red risky vehicle ignores the stop signal at the intersection and quickly approaches the blue ego vehicle. In scenario (b), the risky vehicle ignores the straight-ahead signal at the intersection and makes a sudden left turn toward the ego vehicle. Lastly, in scenario (c), the risky vehicle suddenly drives in reverse on the straight road toward the ego vehicle through various paths. In all scenarios, the yellow vehicle behind the ego vehicle causes a secondary collision when the ego vehicle engages emergency brakes to avoid a risky vehicle, further complicating the collision situation.

A. *Performance Evaluation*

To evaluate the collision avoidance and mitigation performance, we conducted 1,000 simulation tests and compared the results with a TTC-based Autonomous Emergency Braking (AEB) system, an End-to-End Reinforcement Learning (RL) algorithm, and our algorithm that only considers collision avoidance. The comparison among the algorithms was based on collision rate and the average collision impact. The TTC-based AEB system is a system that performs longitudinal speed control for collision avoidance based on TTC. The End-to-End RL algorithm is trained with the same state space, reward function as ours, and action space consisting of throttle, steering, and brake. We assume that the perception of the End-to-End RL algorithm in the experiment is accurate. Lastly, in our algorithm that only considers collision avoidance, the reward function is modified to impose a constant penalty for collisions rather than a penalty proportional to the collision impact.

Table II demonstrates that the proposed method achieves a significantly lower collision rate than the TTC-based AEB system across all collision imminence levels. It also exhibits a comparable collision rate to the End-to-End RL algorithm, and the algorithm focuses solely on collision avoidance. Moreover, our approach shows the best collision avoidance performance among the compared algorithms, both at collision imminence level 1 and when including all collision imminence levels. Although unexpected and potential collisions must be considered in the tested scenarios, learning-based methods, including ours, show generally high collision avoidance rates.

Regarding collision impact related to collision mitigation performance, the proposed method that considers collision avoidance and mitigation in an integrated manner shows the best mitigation performance at all collision imminence levels. The proposed method achieves comparable collision avoidance performance to an algorithm that only considers collision avoidance while showing a lower average collision impact. It demonstrates that our approach performs collision avoidance when collision avoidance is possible while also performing collision mitigation when it is impossible. The End-to-End RL algorithm was also trained using the same reward function as the proposed method but shows relatively lower collision mitigation performance.



(a)

(b)

Fig. 5: Examples of planned paths in simulation. (a) Planned path in straight road collision scenario. (b) Planned path in intersection collision scenario.

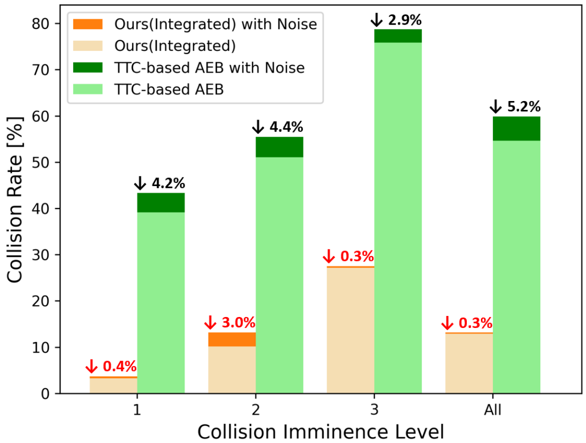


Fig. 6: Collision rate changes for noise in various collision imminence levels.

B. Interpretability and Stability

The interpretability of the system plays a vital role in autonomous driving [28]. For instance, in an autonomous driving system, a planned path through the planning module provides engineers and drivers with a clear understanding of the system's behavior, such as where the vehicle is heading. This planned path can also interact with the prediction module to assess whether following it would lead to potential collisions. In a non-interpretable system, it becomes challenging to pinpoint which module encounters issues when errors occur within the system, ultimately compromising the system's stability.

As depicted in Fig. 5, the proposed algorithm plans a realistic and continuous path for collision avoidance and mitigation in imminent collision scenarios. Through the planned path, the ego vehicle's goal point can be confirmed, enabling various interactions with other modules. It provides interpretability and stability for autonomous driving systems. In contrast, the learning-based method that directly outputs control input through a neural network lacks realism and continuity because there is no path, and control input is also output as a black box through a neural network, which reduces the interpretability and stability of the autonomous driving system.

C. Robustness to Noise

We conducted an additional experiment under noisy conditions to assess the practical utility of our system in the real world. In this experiment, our algorithm is trained using a state that included noise in the position and velocity information of surrounding vehicles. The noise used in the experiment is Gaussian noise with a mean of 0 and a variance of 1.

As shown in Fig. 6, the collision avoidance rate of our method at each collision imminence level is degraded by 0.4%, 3.0%, 0.3%, and 0.3% in noisy environments. In contrast, the TTC-based AEB system exhibits more considerable deterioration, with rates dropping by 4.2%, 4.4%, 2.9%, and 5.2%. It is shown that the proposed approach is more robust to noise, indicating it to be more suitable in real-world conditions. On the other hand, the TTC-based AEB system shows significant degradation in performance under noisy conditions due to factors such as model errors.

TABLE III: Ablation study for confidence p.

Tracking Level	Collision Rate	Avg. Collision Impact
Full Tracking	13.5%	6264 N·s
Init Tracking	40.7%	11188 N·s
Confidence based	**12.9%**	**5793 N·s**

D. Ablation Study

An ablation study was conducted to investigate the effect of confidence p on performance. Table III shows that the confidence-based method performs better regarding collision rate and average collision impact than the Full and Init Tracking methods. The Full Tracking method exhibits a swift ability to reach the target point, but its performance lags behind the confidence-based method due to its inability to consider the sudden interruption of surrounding vehicles while reaching the terminal point. Secondly, the Init Tracking method faces challenges in collision avoidance because there is little change in speed and yaw when only the initial part of the trajectory is tracked. Consequently, it shows the lowest performance among the compared methods. Lastly, the confidence-based method follows only the reliable part by utilizing the confidence p of the trajectory and generates a new trajectory to address sudden changes in surrounding vehicles. In addition, it moves quickly to the terminal point, showing the best performance in collision avoidance and mitigation.

IV. Conclusions

This paper presented a trajectory planning method for integrated CAMS that combines DRL and QP trajectory planning. The proposed approach deals with complex and various collision scenarios via the DRL and leverages the terminal state and confidence of the QP trajectory rather than the direct control input as the output of a neural network to create a more realistic and continuous path. The experimental results indicate that the proposed method achieves a collision avoidance rate comparable to the End-to-End RL algorithm and the algorithm that considers only collision avoidance while showing a lower average collision impact. It uses the terminal state and confidence of the trajectory as the output of a neural network, thereby improving interpretability and stability while maintaining collision avoidance performance compared to existing learning-based methods. Moreover, by evaluating the performance of the proposed algorithm in noisy environments, it is confirmed that the algorithm maintains its performance without significant degradation in noisy conditions. The proposed algorithm is expected to contribute to reducing traffic fatality by ensuring the safety and reliability of autonomous vehicles.

References

[1] A. Jarašūniene and G. Jakubauskas, "Improvement of road safety using passive and active intelligent vehicle safety systems," *Transport*, vol. 22, no. 4, pp. 284–289, 2007. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/16484142.2007.9638143

[2] K. Lee and D. Kum, "Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map," *IEEE Access*, vol. 7, pp. 52 846–52 857, 2019.

[3] P. Seiler, B. Song, and J. K. Hedrick, "Development of a collision avoidance system," *SAE Transactions*, vol. 107, pp. 1334–1340, 1998. [Online]. Available: http://www.jstor.org/stable/44741070

[4] S. Moon, I. Moon, and K. Yi, "Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance," *Control Engineering Practice*, vol. 17, no. 4, pp. 442–455, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066108001548

[5] S. Noh and W.-Y. Han, "Collision avoidance in on-road environment for autonomous driving," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, 2014, pp. 884–889.

[6] Q. Liu, X. Li, S. Yuan, and Z. Li, "Decision-making technology for autonomous vehicles learning-based methods, applications and future outlook," 2021.

[7] X. Chen, Y. Zhai, C. Lu, J. Gong, and G. Wang, "A learning model for personalized adaptive cruise control," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 379–384.

[8] L. A. Serafim Guardini, A. Spalanzani, P. Martinet, C. Laugier, T. Genevois, and A.-L. Do, "Minimal injury risk motion planning using active mitigation and sampling model predictive control," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 1262–1267.

[9] M. Yoshimura, G. Fujimoto, A. Kaushik, B. K. Padi, M. Dennison, I. Sood, K. Sarkar, A. Muneer, A. More, M. Tsuchiya, S. Araki, A. Hebber, T. Tieleman, and Y. Yasui, "Autonomous emergency steering using deep reinforcement learning for advanced driver assistance system," in *2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2020, pp. 1115–1119.

[10] P. Lv, J. Han, J. Nie, Y. Zhang, J. Xu, C. Cai, and Z. Chen, "Cooperative decision-making of connected and autonomous vehicles in an emergency," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 2, pp. 1464–1477, 2023.

[11] B. Simon, F. Franke, P. Riegl, and A. Gaull, "Motion planning for collision mitigation via fem–based crash severity maps," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2187–2194.

[12] M. Parseh, F. Asplund, L. Svensson, W. Sinz, E. Tomasch, and M. Törngren, "A data-driven method towards minimizing collision severity for highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 723–735, 2021.

[13] Z. Liu, J. Chen, H. Xia, and F. Lan, "Quasi-critical collision-avoidance strategy for autonomous vehicles in complex traffic scenarios based on exclusive area of relative velocity vector algorithm," *Robotics and Autonomous Systems*, vol. 153, p. 104049, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022000197

[14] D. Li, J. Zhang, B. Xiao, B. Tang, and Z. Hu, "Vehicle Crash Mitigation Strategy in Unavoidable Collision Scenarios: Focusing on Motion Planning by Considering a Generalized Crash Severity Model," 11 2022. [Online]. Available: https://www.techrxiv.org/articles/preprint/Motion_Planning_of_Vehicles_for_Crash_Mitigation_in_Emergency_Driving_Scenarios/17055446

[15] Y. Qin, E. Hashemi, and A. Khajepour, "Integrated crash avoidance and mitigation algorithm for autonomous vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7246–7255, 2021.

[16] X. Hou, J. Zhang, C. He, C. Li, Y. Ji, and J. Han, "Crash mitigation controller for unavoidable t-bone collisions using reinforcement learning," *ISA Transactions*, vol. 130, pp. 629–654, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057822001422

[17] M. Kim, S. Lee, J. Lim, J. Choi, and S. G. Kang, "Unexpected collision avoidance driving strategy using deep reinforcement learning," *IEEE Access*, vol. 8, pp. 17 243–17 252, 2020.

[18] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1364–1384, apr 2022. [Online]. Available: https://doi.org/10.1109%2Ftnnls.2020.3043505

[19] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3313–3323, 2019.

[20] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017.

[21] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2019.

[22] R. Schubert, C. Adam, M. Obst, N. Mattern, V. Leonhardt, and G. Wanielik, "Empirical evaluation of vehicular models for ego motion estimation," in *2011 IEEE intelligent vehicles symposium (IV)*. IEEE, 2011, pp. 534–539.

[23] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, 07 2014.

[24] A. Takahashi, T. Hongo, Y. Ninomiya, and G. Sugimoto, "Local path planning and motion control for agv in positioning," in *Proceedings. IEEE/RSJ International Workshop on Intelligent Robots and Systems '. (IROS '89) 'The Autonomous Mobile Robots and Its Applications*, 1989, pp. 392–397.

[25] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[27] E.-H. Choi, "Crash factors in intersection-related crashes: An on-scene perspective," Tech. Rep., 2010.

[28] Éloi Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," 2022.