IET Radar, Sonar & Navigation

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Robust Multi-Agent Reinforcement Learning Against Adversarial Attacks for Cooperative Self-Driving Vehicles

Chuyao Wang | Ziwei Wang | Nabil Aouf

Department of Engineering, School of Science and Technology, City St George's, University of London, London, UK

Correspondence: Chuyao Wang (chuyao.wang@city.ac.uk)

Received: 11 April 2025 | **Revised:** 28 April 2025 | **Accepted:** 7 May 2025

Handling Editor: Hugh Griffiths

Funding: The authors received no specific funding for this work.

Keywords: decision making | multi-robot systems | neural nets

ABSTRACT

Multi-agent deep reinforcement learning (MARL) for self-driving vehicles aims to address the complex challenge of coordinating multiple autonomous agents in shared road environments. MARL creates a more stable system and improves vehicle performance in typical traffic scenarios compared to single-agent DRL systems. However, despite its sophisticated cooperative training, MARL remains vulnerable to unforeseen adversarial attacks. Perturbed observation states can lead one or more vehicles to make critical errors in decision-making, triggering chain reactions that often result in severe collisions and accidents. To ensure the safety and reliability of multi-agent autonomous driving systems, this paper proposes a robust constrained cooperative multi-agent reinforcement learning (R-CCMARL) algorithm for self-driving vehicles, enabling robust driving policy to handle strong and unpredictable adversarial attacks. Unlike most existing works, our R-CCMARL framework employs a universal policy for each agent, achieving a more practical, nontask-oriented driving agent for real-world applications. In this way, it enables us to integrate shared observations with Mean-Field theory to model interactions within the MARL system. A risk formulation and a risk estimation network are developed to minimise the defined long-term risks. To further enhance robustness, this risk estimator is then used to construct a constrained optimisation objective function with a regulariser to maximise long-term rewards in worst-case scenarios. Experiments conducted in the CARLA simulator in intersection scenarios demonstrate that our method remains robust against adversarial state perturbations while maintaining high performance, both with and without attacks.

1 | Introduction

In recent years, deep reinforcement learning has demonstrated promising decision-making capabilities for autonomous vehicles in various environments [1–3]. As it brings the prospect of greater convenience, mobility efficiency, and safety to the automotive industry, an increasing number of self-driving vehicles will be deployed on roads in the near future. Research has shown that, compared to single-agent settings, cooperative systems yield higher efficiency [4–6]. When combined with deep reinforcement learning (DRL), MARL enables better coordination, more efficient learning, and improved overall performance in complex cooperative tasks [7, 8]. This is particularly relevant for applications such as autonomous driving, where tasks require multiple vehicles to work cooperatively and maintain awareness of the overall system rather than acting independently [9]. Although MARL has been extensively studied, challenges still persist in designing multi-agent systems. Existing works [10, 11] primarily train the decision-making agents with their local observations. Even though these agents are

trained within a multi-agent framework, they often lack a comprehensive awareness of the overall system and the individual contributions to it. To solve this, Sunehag et al. [8] attempt to use global rewards to inform local agents about the system performance. Additionally, QMIX [12] employs a mixing network to enhance the accuracy of global Q-value estimation by incorporating global awareness of the state. However, these approaches still neglect the individual contributions of each agent to the global performance. These interactions in MARL that each agent learns can be subtle, which means the famous centralised-training-decentralised-execution approaches may resemble multiple single agents working together, rather than achieving the full potential of multi-agent systems. Furthermore, in many multi-agent systems, each agent is designed to complete a specific task based on its own learnt policy network. In Ref. [13], agents are divided into two opposing groups for battle scenarios, as well as into predators and prey for pursuit tasks. In Ref. [14], the experimental setup employs the Google Football environment, where players are assigned different positions and roles to evaluate performance under various configurations. These designs mean that an agent trained for one task often struggles to adapt to others. For example, prey agents are not capable of pursuing predators, and vice versa. Similarly, a football agent trained to play as a forward may struggle when assigned the role of a goalkeeper. Although such a task-specific approach may be necessary for certain robotic applications, it is less appropriate for autonomous driving. In this domain, it is essential that every vehicle is capable of handling a wide range of situations and tasks it may encounter, ensuring a consistent and adaptable driving policy across all vehicles. For instance, in an intersection negotiation scenario, an autonomous vehicle trained specifically to make left turns would be incapable of executing right turns or other manoeuvres, limiting its practical deployment.

Besides the aspects above that MARL methods could benefit from further development, they face another challenges as the vulnerability to adversarial attacks [15]. These adversarial attackers affect and mislead the decision-making process of the learning-based models from designed cyberattack or environment uncertainties, especially harmful to MARL systems. Given the high safety requirements of autonomous vehicles, it is important to develop a robust DRL agent or MARL system to prevent catastrophic failures due to perturbations in realistic environments. This robustness is crucial not only for defending against adversarial attacks but also for handling unavoidable sensor errors and natural equipment inaccuracies, which can impact the agents in a similar manner.

Recognising the importance of model robustness, researches have investigated the single-agent DRL. Several works [16, 17] adopt adversarial training from supervised learning scheme to improve the robustness of such learning-based guidance schemes. Specifically, the agent is occasionally attacked and the adversarial trajectories are generated during the data collection. However, these data augmentation style training with adversarial samples only brings limited improvement. Moreover, comparing to robust single-agent algorithms, enabling robustness for multi-agent systems faces more challenges, as not only the driving agents are influenced by the adversarial attacks, but their behaviours after the attacks could affect other agents in the same scenario. In this work, the cooperative multi-agent autonomous driving is benefited from the introduced communication under normal conditions; however, the shared information also can be perturbed, further compromising the agents that rely on this information. The robustness of a multi-agent system refers to the tolerance of the system to various uncertainties. Besides the adversarial attacks, the highly dynamic driving environment itself introduces uncertainties. One approach to achieve this is by designing models that quantify risk during the task, which can help monitor the level of cautiousness in the system. In DRL, to enhance the system's tolerance to uncertainty, risk is often incorporated into the reward function [18]. However, due to the differing natures of risk and reward, effectively estimating and balancing them together presents a significant challenge.

This paper introduces a novel algorithm to address these key aspects of MARL: developing a more efficient MARL system and increasing its robustness. Specifically, the goal is to enable robust, cooperative, multi-agent reinforcement learning-based self-driving, for handling intersection-passing scenarios in the presence of adversarial observation attacks. Our key contributions are summarised as follows:

i. We establish a cooperative and communicated MARL framework with a universal policy network. The communication and the universal policy network allow us to take into account the interactions among agents based on Mean-Field theory in the training, thereby enhancing the MARL performance. Additionally, the universal policy indicates that each agent is not limited to a specific task. It ensures that one policy can manage all tasks after training.

ii. We define a risk assessment formulation to model both the system and individual risk levels at current state. Similar to long-term rewards and the value network, the risks is minimised with a dedicated risk network at the same time. Moreover, the gap between system risks and individual risks is formulated as a credit assigning method, allowing the policy to be updated by accounting for contributions of each agent to the MARL system.

iii. We propose a robust constrained objective function to obtain a robust policy for intersection negotiation under bounded optimal observation perturbations and a safety criteria.

Experiments are carried out to evaluate the performance of our proposed algorithm for intersection-passing tasks in the realistic unreal-engine powered simulator CARLA [19]. The results show the improvement of performance and robustness of our proposed adversarial defence method for MARL system.

2 | Related Work

2.1 | MARL for Autonomous Driving

Multi-agent deep reinforcement learning (MARL) aims to maximise team rewards, and several effective approaches have been developed. MADDPG [7] extends DDPG to multi-agent

settings, where each agent has its own actor and critic networks, using global information to learn coordinated policies. MAAC (Multi-Agent Actor-Critic) [20] improves coordination by using attention mechanisms to prioritise relevant information from other agents, making it more efficient in complex environments. G2ANet [21] further enhances communication by incorporating graph attention networks to capture agent interactions dynamically. On the value-based side, QMIX uses a mixing network to combine individual Q-values into a global Q-value, ensuring a more accurate global Q-value estimation. QPD [22] decomposes the global Q-function into individual components for better scalability, while QPLEX [23] uses duplex duelling networks to model the interplay between agents. Mean-filed actor-critic method (MFAC) [24] applies the mean-field theory to MARL and thus successfully improves the scalability of MARL with a large number of agents.

Following the major breakthroughs of MARL in recent years, recent advancements in multi-agent deep reinforcement learning (MARL) have expanded the scope of autonomous driving systems by addressing complex interactions among multiple vehicles and agents. Unlike single-agent DRL, which focuses on optimising the performance of an individual vehicle, MARL considers the collaborative and competitive dynamics in a shared driving environment [25, 26]. This approach is particularly useful for scenarios involving multiple autonomous vehicles that must navigate and negotiate their movements in real-time. MARL techniques can significantly enhance the coordination and cooperation among vehicles, improving overall traffic flow, safety, and efficiency. These methods allow vehicles to learn not only from their own experiences but also from the interactions with other agents, leading to more robust decision-making and adaptive behaviours [2, 27, 28]. For instance, D. Li et al. [29] introduce a cooperative control framework where autonomous vehicles use MARL to synchronise their movements and optimise lane merging and intersection crossing, resulting in smoother traffic management and reduced congestion. Furthermore, MARL can address the challenge of nonstationary environments where the behaviour of other agents is dynamic and uncertain. By leveraging multi-agent techniques, vehicles can develop strategies to anticipate and respond to the actions of neighbouring vehicles more effectively [30]. H. Lin et al. [31] propose an enhanced state representation for MARL-based platoon-following models by integrating inter-vehicle dynamics and global traffic information, achieving improved stability over standard MARL baselines. Z. Huang et al. [32] present a method that integrates MARL with communication protocols, allowing vehicles to share information about their intentions and local environment, thereby improving coordination and reducing the likelihood of accidents.

However, MARL systems face notable challenges, particularly regarding their vulnerability to adversarial attacks. The collaborative nature of MARL can make it more susceptible to disruptions caused by malicious agents or unexpected behaviours from other vehicles. Unlike single-agent systems, where robustness can be achieved through isolated adjustments, maintaining robustness in a multi-agent setting is more complex due to the interdependence among agents.

2.2 | Adversarial Attacks on DRL

Though deep learning models recently achieve significant improvement, research shows that these well-trained models are still very vulnerable to adversarial attacks. Adversarial attacks on camera sensor often lead to visually similar images to the normal images from a human perspective, yet they can deceive deep learning models into generating inaccurate predictions. Generally there are two types of adversarial attack methods, white-box attacks and black-box attacks, depending on if the attacker has full access to the models' parameters or not.

In Ref. [33], a Bayes optimisation-based approach was proposed to generate the painting of black lines on the road to counterfeit lane lines and make the vehicle deviate from the original orientation. Experiments were conducted in CARLA simulator, and results showed that end-to-end driving models were attacked and deviated to the orientation chosen by attackers. He et al. [34] combined Bayesian optimisation and Jensen-Shannon (JS) divergence to measure average variation distance of the policies attacked by the observation perturbations for optimal black-box attacks. Behzadan and Munir [35] studied black-box attacks on DQNs with discrete actions via transferability of adversarial examples. Pattanaik et al. [36] further enhanced adversarial attacks to DRL with multi-step gradient descent and better engineered loss function. They required a critic or Q function to perform attacks. Typically, the critic network learnt during agent training. For white-box approaches, S. Huang et al. [37] evaluated the robustness of deep reinforcement learning policies through an FGSM based attack on Atari games with discrete actions. Kos and Song [38] proposed to use the value function to guide adversarial perturbation search. Y. C. Lin et al. [39] considered a more complicated case where the adversary is allowed to attack only a subset of time steps, and used a generative model to generate attack plans luring the agent to a designated target state. In our work, we introduce a FGSM based method to generate optimal adversary examples by maximising the pre-defined collision risk. Results show that with a small strength parameter ε and minimal visual difference, our method can efficiently misguide the well-trained agent.

2.3 | Mitigation Against Adversarial Attacks

Defence methods against adversarial attacks have been explored recently. Zhang, Chen, Xiao et al. [40] proposed a novel Markov decision process (SA-MDP) that considers state-adversarial perturbations and provides a theoretical foundation for robust single-agent reinforcement learning. They developed the principle of policy regularisation that can possibly be applied to many DRL algorithms. Based on SA-MDP, Zhang, Chen, Boning et al. [41] proposed an alternative training framework with learnt adversaries and developed a robust Markov game to address environmental uncertainty by introducing uncertainty into the reward function. Oikarinen et al. [42] proposed the robust ADversarIAl loss (RADIALRL) method, which can improve the robustness of DRL within the ℓ_p norm boundary against attacks with lower computational complexity. Kumar

et al. [43] proposed certified robustness by adding smoothing noise to the state. However, these methods are designed for single-agent RL systems and overlook the specific challenges of MARL, making them difficult to apply effectively. MARL systems are often more vulnerable to adversarial attacks, even when only a single agent is targeted [44]. To counter state-based attacks, Zhou et al. [13] proposed robust policies by minimising the cross-entropy loss between the actions of agents in non-perturbed and perturbed states. Compared to previous methods, our work focuses on mitigating perturbations in both local agent states and global shared states, addressing a more challenging problem than prior approaches.

3 | Methodology

3.1 | Framework Overview

The system overview is shown in Figure 1. Generally, the cooperative MARL system includes N agents with an information sharing module and an adversarial attacker. During inference, the environment generates a tuple of local camera observations $(s_t^1, s_t^2, ..., s_t^N)$. The adversarial attacks are then applied to the clean states, resulting in a new perturbed tuple $(s_{A,t}^1, s_{A,t}^2, ..., s_{A,t}^N)$. Through the information sharing scheme for the connected multi-agent system, the observations will be processed from the perturbed local observation $s_{A,t}^N$ and the neighbour observations $s_{A,t}^{-N}$ to high-level features $Hs_{A,t}^N$ and $Hs_{A,t}^{-N}$ for each individual agent. All agents share a nontask-oriented, universal policy network with the exact same parameters. Based on the local observation and the shared observation (which could be perturbed or not), each agent outputs the action and executes it in the environment. As previously discussed, developing a robust multi-agent system poses significant challenges, particularly due to the uncertainty regarding the number of agents that may be compromised by adversarial attacks. Additionally, while information sharing here enhances decision-making in nonperturbed contexts by incorporating diverse perspectives from neighbouring agents, it can have a backward impact when the shared observations themselves are exposed to attacks, leading to more erroneous decision-making. In this section, we address the existing problems and propose a robust cooperative deep adversarial reinforcement learning approach for autonomous driving agents against strong observation perturbations.

3.2 | Mean-Field Communicated Multi-Agent Structure

A multi-agent deep reinforcement learning task can be considered as a continuous decision-making problem, which follows the Stochastic Games (SG) [45]. SG is defined as a tuple $(\mathcal{S}, \mathcal{A}^1, ..., \mathcal{A}^N, \mathcal{R}^1, ..., \mathcal{R}^N, P, \gamma)$. $\mathcal{N}$ is the number of the agents, $\mathcal{A}^j$ is the action space of agent j, $\mathcal{R}^j$ is the step reward of agent j. The agents interact with the environment of successive joint state $(\mathcal{S}: s^1 \times \cdots \times s^n)$ with a joint action, getting the step rewards $(r^1 \in \mathcal{R}^1, ..., r^N \in \mathcal{R}^N)$ and the transition probability $p \in \mathcal{P}$ to the next joint state under the current joint state and the current joint action. γ indicates the discount factor. Considering the state perturbation, we introduce the worst case perturbed joint state and the perturbed action responded to it $(\mathcal{S}_A, \mathcal{A}_A^j)$ to the original SG tuple. The goal of the robust MARL is to find a series of optimal policies that return the maximum accumulative discounted team returns under the worst adversarial attacked states:

$$Q_{\pi^*}^i = \max_{\pi} \min_{\mathcal{S}_A} \left[\mathbb{E}\left(\sum_t^T \gamma^t r_t^i(s_t^i/s_{A,t}^i, a_t^i/a_{A,t}^i) \right) \right] \tag{1}$$

The attacked state s_A is a shifted state, which models the worst case perturbation due to attacks on the sensor leading to a_A suboptimal than a^*. A well-trained guidance policy network may be able to cope with a weak and quick perturbation, turning back to the optimal actions and the desired trajectory after the state observations get back to normal. However under strong and continuous adversarial attacks in the context of multi-agent, the guidance policy networks can easily fail.

One problem for multi-agent reinforcement learning is the difficulty to model the interactions among the agents. Mean-field actor-critic reinforcement learning (MFAC) uses the mean-field theory to transform the interactions of multiple agents into the interactions between two agents, which enables possible large-scale multi-agent reinforcement learning. In MFAC, the long-term expected Q value for agent i at state s with the joint action a, $Q^i(s, a)$ is decomposed to the sum of Qs when interacting with each agents:

$$Q^i(s, a) = \frac{1}{N-1} \sum_{k \subseteq N-1} Q^i(s, a^i, a^k) \tag{2}$$

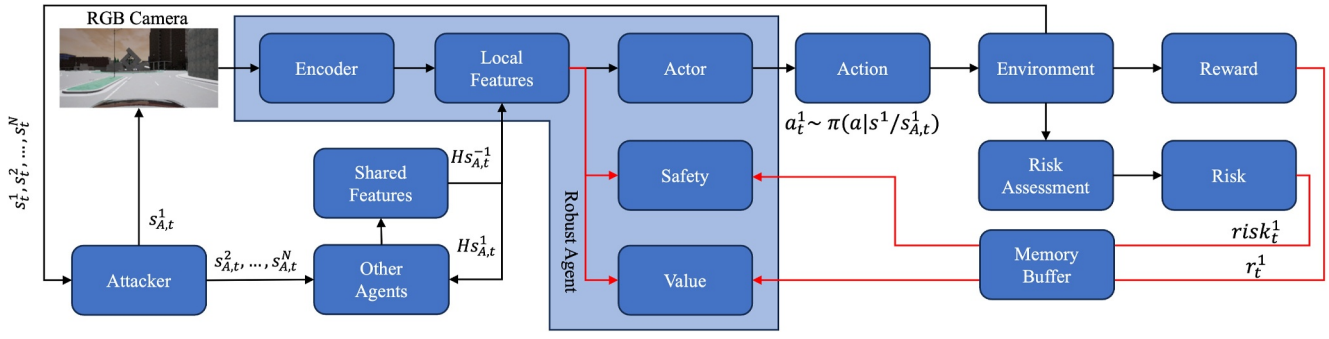

FIGURE 1 | The framework overview. One robust agent is displayed as the primary example to demonstrate the information flow and main opponents of the system. Red arrows indicate training only while black arrows indicate training and inference.

where the a^k counts as the local interaction between agent i with the neighbour agent k. In our case, we use a universal policy network for every agent in the MARL, to enable a universal driving agent that works in every task in the intersection scenario. Therefore, the decomposition becomes:

$$Q^i(s,a) = \frac{1}{N-1}\sum_{k \subseteq N-1} Q^i\left(s^i, a^i, s^k\right) \tag{3}$$

As we use a universal policy, the actions a^k generated by different agents only depend on their observations s^k. The action of one neighbour can be replaced by the state of the neighbour. It is proved by MFRL [24].

$$Q^i(s,a) \approx Q^i\left(s^i, a^i, \bar{s}^k\right) \tag{4}$$

where s^{-k} is the mean state or fusion state of all neighbour agents. The local agent i makes decision based on the local observation and the information shared by other agents, which is processed through two FC layers and an attention module.

3.3 | Gradient-Based Attacker

In this section, the inner minimisation part of Equation (1) is solved by modelling the optimal adversarial attacks on observations based on a white-box technique. We require an adversarial attack generation method that prioritises ease of implementation and computational efficiency. The Fast Gradient Sign Method (FGSM), introduced by Goodfellow et al. [46], fulfils these criteria. Additionally, FGSM offers flexibility, allowing for the application of techniques like Basic Iterative Method (BIM) [47] to introduce iterations and enhance the adversarial attack instead of increasing the perturbation parameter ε.

FGSM works by using the gradients of the neural network to create an adversarial example. For deep reinforcement learning, the method uses the gradient $\bigtriangledown_s L(\pi, s, a)$ of the loss with respect to the input observation to create a new observation that maximises this loss. The π refers to the policy network parameters, while the a is the output action. The new masked observation is obtained by the following equation:

$$s_A = s + \varepsilon \times \text{sign}\left(\bigtriangledown_s L(\pi, s, a)\right) \tag{5}$$

This new observation s_A is called the adversarial state. The ε parameter controls the strength of the attack, and it varies from 0 to 1. The closer the value to 1, the stronger effect on the targeted policy network, but this also makes s_A more visually distinguishable. On the contrary, if the ε is small, its impact is weaker, and it becomes harder for human eyes to detect differences from the normal state. As discussed in the beginning of the section, to solve the inner minimisation, the generated perturbations should lead the policy to worst case situations. In our cooperative MARL system, we find an unintended-action-leading loss function instead of directly minimising the long-term rewards. Within the realm of autonomous driving, certain throttle and steering actions in certain scenarios may pose potential danger. Based on this idea, we introduce the gradient-based perturbations into the control system by maximising the throttle and reversing the steering, thereby deliberately maximising the risk of collisions. After introducing our augmented FGSM, the agent will be encouraged to accelerate and make wrong steering to make the collision.

The problem of using FGSM is the effective sensitivity to ε. In practice, we need a larger ε to make the perturbation effective on the deep guidance model decision. However during training, this will make perturbations predictable and tolerated by the model [17]. We employ the BIM to intentionally escalate the attack effeteness associated with autonomous driving. The BIM iterates the process of a single FGSM, which means a previous generated adversarial state will be the input of the next adversary generation. By iterating FGSM, the influence of the attacks can be increased without affecting the sensory image as much as by increasing the ε by a comparable amount. Finally, with the risk encouragement and BIM, the optimal adversary is obtained as follows:

$$\begin{aligned} s_A^i &= s_n^i + \varepsilon \times \text{sign}\left(\bigtriangledown_{s_n^i}\left(L_n^i\right)\right) \\ s_n^i &= s_{n-1}^i + \varepsilon \times \text{sign}\left(\bigtriangledown_{s_{n-1}^i}\left(L_{n-1}^i\right)\right) \\ &\cdots \end{aligned} \tag{6}$$

where

$$L_n^i = -2 \times \left(steer^i | s_n^i\right) + 1 - \left(throttle^i | s_n^i\right) \tag{7}$$

s_A^i indicates the optimal adversary and n is the iteration number. $steer^i$ and $throttle^i$ are the outputs of the actor network with observation s_n^i. Note that the attacker applies perturbations only on the targeted agent's observation, which is more realistic for the attacker to aim on the camera of that agent. The shared information could be perturbed through the cameras of neighbour agents, interfering with the decision-making on the receiving end. These attacks will lead the model to the worst actions in most of the cases.

3.4 | Risk Assessment Formulation

To enhance the robustness of the MARL algorithm, a safety criteria is also introduced. We quantify the risks in the multi-agent framework as the collision probability of any two agents in the whole system. Reinforcement autonomous driving agents need to not only maximise the rewards and finish the tasks, but also be aware and reduce the risks during the navigation, which is particularly crucial for multi-agent self-driving. To simplify the risk analysis and disregarding the varying dimensions and shapes of individual vehicles, we represent each vehicle as a circle with a diameter equal to its diagonal. As shown in Figure 2, considering two agents running towards each other with current speeds v_1, v_2 and rotation to the world coordinate θ_1, θ_2, at world locations $(x_1, y_1), (x_2, y_2)$. Assume if the two agents maintain their current speeds and rotations, they will meet at the intersection of their trajectories. If we slide the two circles representing the two vehicles along their respective trajectories (the dotted lines), the first point of collision is identified when agent 1 is at $P1$ and Agent 2 is at $P2$, and the last point

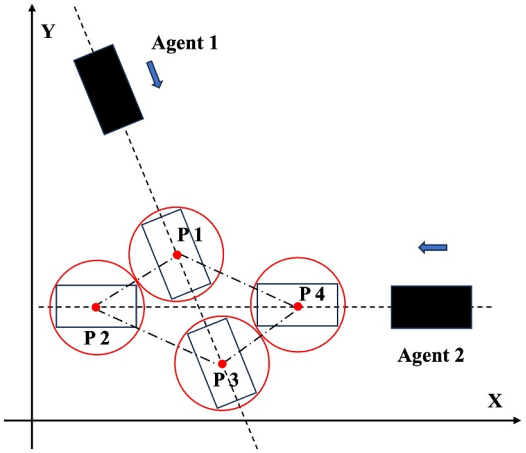

FIGURE 2 | The illustration of the area that could lead to collision between two agents.

of contact when agent 1 is at $P3$ and Agent 2 is at $P4$. This implies that between the moments when Agent 2 passes $P4$ and leaves $P2$, while agent 1 remains between $P1$ and $P3$, there is a highly chance of collision between the two agents. Given the orientations, diagonals, and locations of the two agents, we can determine the coordinates of the four points $P1, P2, P3$ and $P4$. Based on the agents' current speeds, the times associated with these positions are as follows: t_0 represents current time, t_1 is the time when agent 1 reaches P_1, t_3 is the time when agent 1 leaves P_3, t_4 is the time when Agent 2 reaches P_4, t_2 is the time when Agent 2 leaves P_2. The risk probability is then defined as follows:

$$P_{risk} = \frac{[t_1, t_3] \cap [t_4, t_2]}{[t_1, t_3] \cup [t_4, t_2]} \tag{8}$$

$P_{risk} \subseteq [0, 1]$ indicates the ratio of the overlap in the time intervals when both agents are within the collision zone to the total time interval between the moment the first agent entering the zone and the last agent leaving the zone. To calculate the overall risk in a MARL system, where P_{risk} describes the risk between two agents:

$$P_{tot} = 1 - \prod_{i=1}^{N-1} \left(1 - P_{risk,i}\right) \tag{9}$$

where N is the number of agents and $P_{tot} \subseteq [0, 1]$.

After the careful design of the risk evaluation, it is then embedded to our robust MARL.

3.5 | Robust Constrained Cooperative MARL

In this section, we introduce a robust multi-agent constrained optimisation with risk assessment integrated to solve the outer maximisation in Equation (1). We extend Proximal Policy Optimisation (PPO) to the cooperative multi-agent context, where each agent learns a policy while coordinating with other agents, leveraging a centralised training with decentralised execution structure:

$$\mathcal{L}^i_{clip}(\pi) = \hat{E}_t\left[\sum_{i=1}^{N} \min\left(p^i_t(\pi), clip\left(p^i_t(\pi), 1-\varepsilon, 1+\varepsilon\right)\right)\hat{A}^i_t\right] \tag{10}$$

where π is the policy parameter of the deep guidance agents and N is the number of Agent. In our case, π is the universal policy parameters. $\hat{E}_t$ denotes the estimated expectation over t of the collected trajectories. p_t is the ratio of the probability under the new and old policy, while ε stands for the constant clip term that limits the policy update by constraining the ratio within a specified range, preventing excessively large updates. $\hat{A}^i_t$ is the estimated advantage value for agent i, obtained by the difference between the observed reward for taking an action a^i at local state s^i with shared state s^{-k} and the expected value predicted by the critic network at the same local and shared states:

$$\hat{A}^i_t = Q^i_t(s^i, a^i, s^{-k}) - V^i_t(s^i, s^{-k}) \tag{11}$$

3.5.1 | Long-Term Risk Minimisation

Equation (10) aims to maximise the long-term expected returns. Similar to the expected returns, we can also design a risk objective to minimise the long-term expected risks based on the risk assessment, for the agent safety in the MARL. At every step, the risk of the MARL system is evaluated, and at the end of the episode, the overall risks are computed as the discounted cumulative risks. In order to fit in the same maximisation form as the returns, we use $(1 - risk) \in [0, 1]$ as the safety term, and the safety-to-go at t is defined as follows:

$$S^i_t = \sum_{j=0}^{T-t} (\gamma\lambda)^j\left(\left(1 - P_{tot}(s^i_t, s^{-k}_t)\right) + \gamma S_\theta(s^i_{t+1}, s^{-k}_{t+1})\right) \tag{12}$$

where λ is the Generalised Advantage Estimation (GAE) [48] parameter used to control the bias-variance trade-off. Maximising the S_t is equivalent to minimising the risks. Similar to the critic network, the risk network S_θ will be iteratively updated to achieve accurate long-term expected safety prediction by the mean square error (MSE):

$$\mathcal{L}^i_{risk} = \frac{1}{T}\sum_{t=0}^{T}\left(S^i_\theta(s^i_t, s^{-k}_t) - S^i_t(s^i_t, s^{-k}_t)\right)^2 \tag{13}$$

For policy optimisation, in addition to the advantage value obtained from Equation (11), a risk advantage function plus the single agent contribution is considered:

$$\hat{A}^i_{fi,t} = \hat{A}^i_t + \left(S^i_t - S^i_\theta\right) + Credict^i \tag{14}$$

We define $Credict^i$ as the weighted risk advantage value, which reflects the importance of agent i within the MARL system or the contribution of agent i to the system's overall performance. Credit assigning is crucial in cooperative MARL training, as it ensures individual actions are rewarded or penalised appropriately for the overall performance. Instead of only assigning the team reward uniformly to every agent, we introduce a risk-value-based method enabling nonlinearity in credit assigning. The credit is

defined as the difference between the total risk of the MARL system when agent Beta distribution is included and excluded:

$$Credict^i = \prod_{j=1,j\neq i}^{N-1}\left(1-P_{risk,j}\right) - \prod_{j=1}^{N-1}\left(1-P_{risk,j}\right) \tag{15}$$

By replacing $\hat{A}_t^i$ with $\hat{A}_{fi,t}^i$ in Equation (10), the long-term expected risk is minimised efficiently in the MARL context.

3.5.2 | Adversarial Regulariser

The proposed stochastic universal policy network outputs probability distributions of the predicted actions. We employ a multivariate Beta distribution as the policy output, since a bounded action space is required for steering and throttle within the interval **[0,1]**. The actions are randomly sampled from this distribution to increase exploration in training phase, and the mean value is chosen for inference phase. A robust MARL system should behave similarly or close enough under perturbed observations and normal observations, which means the predicted action distributions with minimal divergence. We extend the theorem in Ref. [49] to our cooperative MARL scheme with universal policy. Given a policy π and its value function $V_\pi(s^i, s^{-k})$ and considering the worst perturbation situation where all agents are attacked with optimal perturbation s_A^i, for all $s^i, s^{-k} \in \mathcal{S}$ and the corresponding $s_A^i, s_A^{-k} \in \mathcal{S}_A$, we obtain the following:

$$\max V\left(\left(s^i, s^{-k}\right), \left(s_A^i, s_A^{-k}\right)\right) \leq \max D\left(\left(s^i, s^{-k}\right), \left(s_A^i, s_A^{-k}\right)\right) \tag{16}$$

where

$$V\left(\left(s^i, s^{-k}\right), \left(s_A^i, s_A^{-k}\right)\right) = V_\pi\left(s^i, s^{-k}\right) - V_\pi\left(s_A^i, s_A^{-k}\right) \tag{17}$$

$$D\left(\left(s^i, s^{-k}\right), \left(s_A^i, s_A^{-k}\right)\right) = D_{TV}\left(\pi\left(s^i, s^{-k}\right), \pi\left(s_A^i, s_A^{-k}\right)\right) \tag{18}$$

$D_{TV}\left(\pi\left(s^i, s^{-k}\right), \pi\left(s_A^i, s_A^{-k}\right)\right)$ is the total variation distance between the predicted action distributions when all the agents are attacked and attack-free. This is the largest possible difference between the probabilities that the two probability distributions can assign to the same action. According to Pinsker's inequality [50], D_{TV} can be linked to another distance Kullback–Leibler (KL) divergence [51]:

$$D_{TV}\left(\pi\left(s^i, s^{-k}\right), \pi\left(s_A^i, s_A^{-k}\right)\right) \leq \sqrt{\frac{1}{2} D_{KL}\left(\pi\left(s^i, s^{-k}\right), \pi\left(s_A^i, s_A^{-k}\right)\right)} \tag{19}$$

In practice, computing KL divergence is more efficient compared to the summing operation in D_{TV} and easier to realise in continuous space. The differentiable aspect of KL divergence also benefit the gradient based optimisation in the DRL algorithms. Therefore, we further amend Equation (16) with the KL divergence to:

$$\max V_\pi\left(\left(s^i, s^{-k}\right), \left(s_A^i, s_A^{-k}\right)\right) \leq \max \sqrt{\frac{1}{2} D_{KL}\left(\pi\left(s^i, s^{-k}\right), \pi\left(s_A^i, s_A^{-k}\right)\right)} \tag{20}$$

Which means the minimisation on KL divergence of the action probabilities under nonperturbed and attacked states guarantees the minimisation on the total variation distance D_{TV}, therefore the performance gap could be minimised too. The regulariser will be added to the final objective function to update the policy network as follows:

$$\mathcal{L}_{reg,t}^i = \sqrt{\frac{1}{2} D_{KL}\left(\pi\left(s_t^i, s_t^{-k}\right), \pi\left(s_{A,t}^i, s_{A,t}^{-k}\right)\right)} \tag{21}$$

3.5.3 | Constrained Objective Function

Consider a well-trained cooperative MARL system. For every state $\mathcal{S}$ in all possible trajectories, there exists a subset actions $\mathcal{A}_{robust}^i \in \mathcal{A}$ under bounded worst-case state perturbations that leads to $\mathcal{S}_{safe}^i \in \mathcal{S}$ avoiding high-risk states for agent i. Therefore, it is possible to guarantee a robust policy by constraining the policy to output safe trajectories and the worst case perturbations. Intuitively, there are two constraints in the objective function to design: one for a bounded adversarial perturbations and one for safe states.

Control barrier function (CBF) is widely deployed in autonomous driving applications for ensuring safety by providing mathematical guarantees that the vehicle will avoid unsafe situations, such as collisions, while respecting constraints such as speed limits and obstacle avoidance. CBF enables real-time adaptability and seamless integration with existing control systems, ensuring safe and reliable navigation in dynamic environments. We define the CBF of our cooperative MARL as the risk value network $h\left(s^i, s^{-k}\right) = S_\theta\left(s^i, s^{-k}\right)$. The calculation of the proposed CBF is based on the nonperturbed states only, as we need the actual states of the agents for the actual risk estimations. The safety set of states is then defined as follows:

$$\mathcal{C} = \left\{s^i \in \mathcal{S} : h\left(s^i, s^{-k}\right) \geq \varepsilon\right\} \tag{22}$$

ϵ is the safety criteria, and the system dynamics is defined as follows:

$$\dot{s}^i = f\left(s^i\right) \tag{23}$$

As a model-free method, the agent learns a policy directly through the environment interactions without knowing or modelling the environment's dynamics. This presents a significant challenge when it comes to predicting how an action will transition the system from one state to the next. Based on Ref. [52], the continuous states of a well-trained model are Lipschitz-continuous. Therefore, the system dynamics could be approximated as the state difference over the time step:

$$\dot{s}^i = \frac{s_{t+\Delta t}^i - s_t^i}{\Delta t} \tag{24}$$

To ensure the states stay in the safety set, we need to make sure the Lie derivative condition [53]:

$$\begin{aligned}&\frac{d}{dt}h\left(s^i,s^{-k}\right)+\omega\left(h\left(s^i,s^{-k}\right)-\epsilon\right)\geq 0\\&\Rightarrow\nabla_{s^i}h\cdot\frac{s^i_{t+\Delta t}-s^i_t}{\Delta t}+\omega\left(h\left(s^i,s^{-k}\right)-\epsilon\right)\geq 0\end{aligned}\tag{25}$$

We use a linear $\omega>0$ to make sure $h\left(s^i,s^{-k}\right)$ can be updated to satisfy the criteria. For the bounded worst-case perturbations, let g be the attacker, we have the L_2 norm of $\left(g\left(s^i,\pi\right)-s^i\right)$ bounded by μ.

Then the objective function of our proposed robust cooperative MARL is developed. Combining maximising the long-term expected rewards, minimising the regulariser, subjecting to the constraints, we tend to solve the following optimisation problem:

$$\begin{aligned}&\min_{\pi}\left\{\frac{1}{T}\frac{1}{N}\sum_{t=0}^{T}\sum_{i=0}^{N}\left(\mathcal{L}^i_{fi}(\pi)+\mathcal{L}^i_{reg,t}\right)\right\}\\&subject\quad to\qquad \mu-L_2\left(f\left(s^i_t,\pi\right),s^i_t\right)\geq 0\\&\nabla_{s^i}h\cdot\frac{s^i_{t+\Delta t}-s^i_t}{\Delta t}+\omega\left(h\left(s^i_t,s^{-k}_t\right)-\epsilon\right)\geq 0\end{aligned}\tag{26}$$

where t and γ denote the time step and discount factor respectively. And $\mathcal{L}^i_{fi}(\pi)$ is the improved objective function for the multi-agent system:

$$\mathcal{L}^i_{fi}(\pi)=\min\left(p^i_t(\pi),clip\left(p^i_t(\pi),1-\varepsilon,1+\varepsilon\right)\right)\hat{A}^i_{fi,t}\tag{27}$$

Based on the Lagrange multiplier technique, we have the generalised Lagrange function of the objective:

$$\mathcal{L}(\pi,\alpha,\beta)=\frac{1}{T}\frac{1}{N}\sum_{t=0}^{T}\sum_{i=0}^{N}\left(\mathcal{L}^i_{fi}(\pi)-\mathcal{L}^i_{reg,t}-\alpha C_1-\beta C_2\right)\tag{28}$$

where $\alpha\geq 0,\beta\geq 0$ are the Lagrange multipliers, and C_1,C_2 are the first and the second constraints respectively. Now we define a function $\theta_P(\alpha,\beta)$ with respect to α,β:

$$\theta_P(\alpha,\beta)=\max_{\alpha\geq 0,\beta\geq 0}\mathcal{L}(\pi,\alpha,\beta)\tag{29}$$

If π satisfies the constrained conditions in Equation (26), we obtain the following:

$$\theta_P(\alpha,\beta)=\frac{1}{T}\frac{1}{N}\sum_{t=0}^{T}\sum_{i=0}^{N}\left(\mathcal{L}^i_{fi}(\pi)+\mathcal{L}^i_{reg,t}\right)\tag{30}$$

otherwise $\theta_P(\alpha,\beta)=\infty$. Equation (30) denotes that minimising the new defined function $\min_{\pi}\theta_P(\alpha,\beta)$ is equivalent to the primal problem in Equation (26). The primal optimisation problem is able to be transferred as follows:

$$\min_{\pi}\theta_P(\pi)=\min_{\pi}\max_{\alpha\geq 0,\beta\geq 0}\mathcal{L}(\pi,\alpha,\beta)\tag{31}$$

Furthermore, based on the Lagrange duality, the dual problem of Equation (31) will always have smaller value than the primal problem, which leads to smaller loss and better performance. With the generalised Lagrange function, $\forall\pi,\alpha,\beta$:

$$\begin{aligned}&\min_{\pi}\mathcal{L}(\pi,\alpha,\beta)\leq\mathcal{L}(\pi,\alpha,\beta)\leq\max_{\alpha\geq 0,\beta\geq 0}\mathcal{L}(\pi,\alpha,\beta)\\&\Rightarrow\min_{\pi}\mathcal{L}(\pi,\alpha,\beta)\leq\max_{\alpha\geq 0,\beta\geq 0}\mathcal{L}(\pi,\alpha,\beta)\\&\Rightarrow\max_{\alpha\geq 0,\beta\geq 0}\min_{\pi}\mathcal{L}(\pi,\alpha,\beta)\leq\min_{\pi}\max_{\alpha\geq 0,\beta\geq 0}\mathcal{L}(\pi,\alpha,\beta)\end{aligned}\tag{32}$$

Therefore, it is easier to optimise on the dual problem than the primal problem. This optimisation is detailed by the two following steps iteratively:

$$\begin{aligned}&1.\quad\min_{\pi}\mathcal{L}(\pi,\alpha,\beta)\\&2.\quad\max_{\alpha\geq 0,\beta\geq 0}\mathcal{L}(\pi,\alpha,\beta)\end{aligned}\tag{33}$$

First, we freeze the Lagrange multipliers α,β and update the policy π based on step 1, which is represented as minimising the $\mathcal{L}(\pi,\alpha,\beta)$:

$$\min_{\pi}\left\{\frac{1}{T}\frac{1}{N}\sum_{t=0}^{T}\sum_{i=0}^{N}\left(\mathcal{L}^i_{fi}(\pi)+\mathcal{L}^i_{reg,t}-\alpha C_1-\beta C_2\right)\right\}\tag{34}$$

Then similarly, the updated policy π are fixed to update the Lagrange multiplier α and β based on step 2:

$$\begin{aligned}\max_{\alpha\geq 0,\beta\geq 0}\Bigg\{\frac{1}{T}\frac{1}{N}\sum_{t=0}^{T}\sum_{i=0}^{N}\Bigg(&-\alpha\left(\mu-L_2\left(f\left(s^i_t,\pi\right),s^i_t\right)\right)\\&-\beta\left(\nabla_{s^i}h\cdot\frac{s^i_{t+\Delta t}-s^i_t}{\Delta t}+\omega\left(h\left(s^i_t,s^{-k}_t\right)-\epsilon\right)\right)\Bigg)\Bigg\}\end{aligned}\tag{35}$$

Finally, the original constrained minimisation problem becomes a maxmin problem without constraints. We use gradient descent to find the optimal robust universal policy π and the optimal Lagrange multipliers α,β repeatedly through Equations (34) and (35) in each batch update. The objective function is a jointly optimisation problem, where the policy network balances the performance under normal observation and the adversarial defence capability. In the experiments, we find the policy will learn to satisfy the constraints in Equation (26) and resulting two constrained functions to decrease during the gradient descent process, and α,β in step 2 will gradually approach zero, to an optimal or near optimal solution. Consequently, step 1 will find the minimum after α and β becomes stable. The method R-CCMARL is detailed in Algorithm 1.

ALGORITHM 1 | R-CCMARL.

1: Initialise the universal policy network parameter θ_0, the state-value critic network parameter ϕ_0, the state-risk network parameter ψ_0, the replay buffer $\mathcal{D}$;
2: Initialise the Lagrange multiplier α;
3: **for** $j=0,1,2,\ldots$ **do**
4: **for** $t=0,1,2,\ldots$ **do**
5: For each agent, share its nonperturbed observation and receive nonperturbed observations from all the neighbours.
6: Sample action $a^i_t\sim\pi\left(\theta_j\right)\left(a^i_t|s^i_t,s^{-k}_t\right)$ based on the policy $\pi\left(\theta_j\right)$ and state $\left(s^i_t,s^{-k}_t\right)$
7: Generate optimal adversary $s^i_{A,t}$ through Equation (6).
8: Obtain transitions $\left\{s^i_{t+1},r^i_t,P^i_{tot,t}\right\}$ by executing a^i_t.
9: Store the transitions $\left\{s^i_t,a^i_t,r^i_t,P^i_{tot,t},s^i_{t+1},s^i_{A,t}\right\}$ in $\mathcal{D}_j$.

10: **end for**
11: Calculate the reward-to-go $\hat{R}_t^i$.
12: Calculate the safety-to-go $\hat{S}_t^i$.
13: Calculate estimated advantage $\hat{A}_{fi,t}^{\ i}$ through Equation (14).
14: Calculate the regulariser $\mathcal{L}_{reg,t}^i$.
15: Update the universal policy θ by Equation (34).
16: Update the Lagrange multiplier α and β by Equation (35).
17: Update the state-value critic network parameter ϕ.
18: Update the state-risk network parameter ψ by Equation (13).
19: **end for**

The proposed robust cooperative communicated MARL network structure is illustrated in Figure 3. At each time step, agent i takes a monocular RGB image as input, which is passed through an encoder to extract essential visual features. These features are then concatenated with five additional numeric features that represent other important state information, including throttle, velocity, steer, distance to the road centre, angle between the vehicle vector and the waypoint vector. This concatenation of data results in a high-level features i. Simultaneously, each neighbouring agent extracts high-level features using the same encoder with identical parameters. The high-level features of all neighbours are concatenated and passed through two fully-connected layers, producing features $-k$ of the same shape as high-level features i.

Next, the local features of agent i and the shared features of its neighbours are fused by two attention modules to create a more comprehensive representation of the environment. Specifically, the customised attention module 1 takes the fused observation $-k$ as input, and outputs the learnable parameter weighted attention map $\gamma \cdot A^{-k}$. This module removes the element-wise summation between $-k$ and A^{-k}. Next, local high-level features i is concatenated with $\gamma \cdot A^{-k}$ and passed to attention module 2, where the summation operation between input features and weighted attention map is retained. Finally, after processing through the two attention modules, the final enhanced feature map is generated, which is then passed to the three separate sub-networks. The attention structure ensures that the MARL model relies less on shared information for decision-making during the early stages of training, as the learnable parameters are initialised to zero. Nevertheless, as training progresses, the additional information can increasingly contribute to the decision-making process. These sub-networks are responsible for predicting three key outputs for agent i: the long-term expected returns, the long-term expected safety values, and the optimal actions. These outputs guide the agent in making safe and effective decisions within the multi-agent environment, ensuring robustness against state perturbations.

3.6 | Reward Function

To ensure that the MARL system accomplishes the intersection negotiation task while maintaining efficiency, a simple yet efficient reward function is designed as the product of three factors plus the terminal reward:

$$reward = V \times D \times A + r_{done} \tag{36}$$

where V is the velocity factor, defined as follows:

$$V = \begin{cases} \dfrac{v}{v_{\min}}, & \text{if } v < v_{\min} \\ 1, & \text{if } v_{\min} \le v \le v_{\max} \\ 1 - \dfrac{v - v_{\max}}{3}, & \text{if } v > v_{\max} \end{cases} \tag{37}$$

here v represents the current speed of the agent. The deviation factor D is defined as follows:

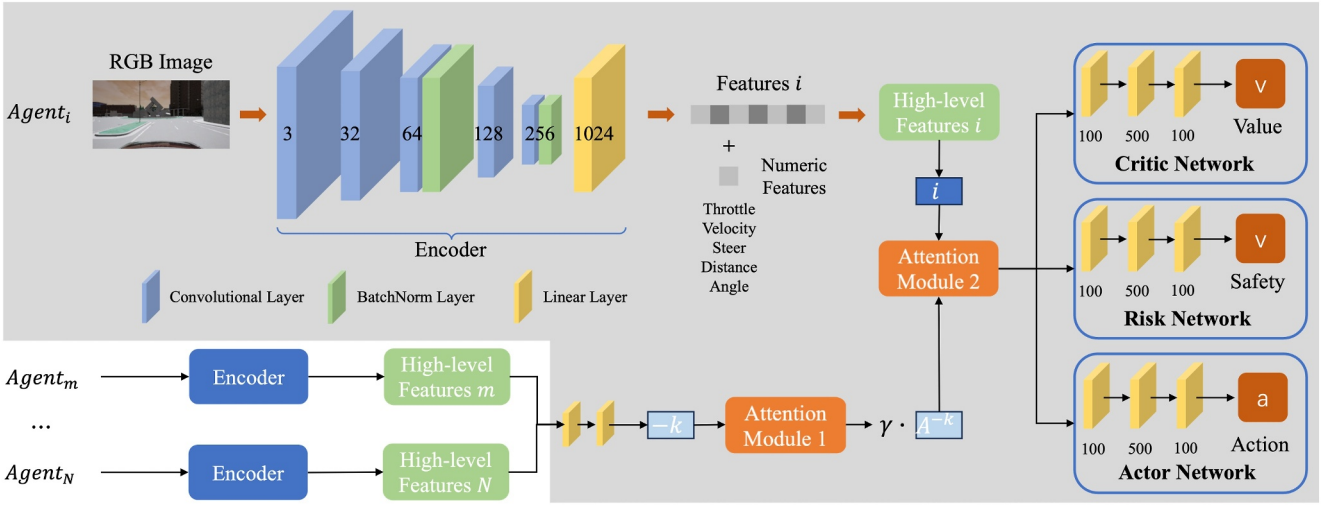

FIGURE 3 | Network structure of the proposed robust cooperative communicated MARL model. A universal driving model is applied, consisted of a feature extract encoder, an actor, critic, risk network and two fully-connected layers and two attention modules. The grey area indicates all the components in one agent. The three sub-task networks are fed with the enhanced features, which involves the local information the shared information from the neighbour agents.

$$D = 1 - \frac{\text{deviation}}{\text{deviation}_{\max}} \tag{38}$$

where deviation is the distance to the road centre. Finally, the angle factor A is defined as follows:

$$A = 1 - \frac{\text{angle}}{\text{angle}_{\max}} \tag{39}$$

where *angle* is the angle between the forward vectors of the vehicle and the road. For r_{done}, five terminal signals are considered: if $\text{deviation}_{\max}$ or $\text{angle}_{\max}$ is exceeded, the agent will receive a terminal reward of −50. A collision with another vehicle results in a reward of −100. A collision with anything other than a vehicle incurs a reward of −50. If the goal position is reached, the agent will receive a terminal reward of 20. This reward function is applied to all the models in the experiments.

4 | Experiments

The training and testing of the proposed framework are implemented in CARLA simulator. CARLA is an open-source platform for development, training, and validation of autonomous driving systems. It has a rich library of vehicle models and realistic urban road modelling, hence being near ideal to urban driving simulation. We set up our experiment scene in Town 2 within the CARLA map library.

4.1 | Implementation Details

In the experiment, we use three vehicles in the cooperative MARL system to complete the intersection-passing task in two types of settings:

i. T-shaped Intersection: As shown in Figure 4, the three vehicles are spawned in different directions at a T-shaped intersection, starting in the designated spawning areas (green areas). Agent 1 aims to go straight, while both Agent 2 and Agent 3 make left turns, creating a highly collision-prone situation. The experiments are conducted at two different intersections to introduce randomness.

ii. 4-way Intersection with Traffic: As shown in Figure 5, the vehicles are spawned in the same setting as the first scenario. In addition, 7 surrounding vehicles, controlled by autopilot, are positioned around the intersection and are ready to pass through. The cooperative MARL system must negotiate with the agent vehicles for intersection passage while avoiding collisions with the dynamic surrounding vehicles.

We select two state-of-the-art multi-agent methods and compare them with our proposed algorithm R-CCMARL and a variant of our method, referred as the risk-only model, in which the constraints for robust cooperative MARL are not implemented.

Each agent takes a 160 * 80 RGB image from a monocular camera and 5 numeric features as input: throttle, velocity, steer,

FIGURE 4 | Experimental scenario 1 with two different intersections.

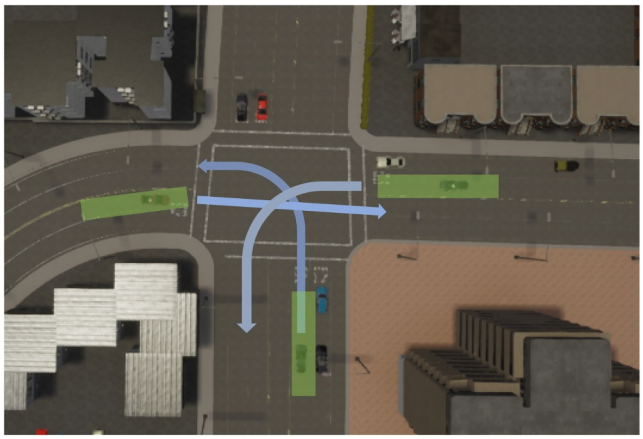

FIGURE 5 | Experimental scenario 2 with surrounding traffic.

the distance between the vehicle and the road centres, and the angle between the vehicle forward vector and the tangent to the road as shown in the network structure in Figure 3. The policy network, the value network and the safety network share the parameters in feature extractor, where the high-level information is shared to benefit all tasks during the training. For the constraint threshold μ and ϵ, we selected suitable values manually from candidate values through experimentation. These thresholds serve to restrict the magnitude of adversarial attacks and the safety criteria, and are sensitive to the scenario while significantly affecting the stability and performance of the training process. The main hyper-parameters used are shown in Table 1. To achieve the nontask-oriented policy network in multi-agent self-driving, all the agents share the same model structure and hyper-parameters. For adversarial attack generations, we use $\varepsilon = 0.1$ and iterations of 20. In the robust MARL evaluation section, we compare the performance of our proposed method with the variant risk-only-model and existing

methods. All models are set to run at 10 fps in CARLA on a single RTX 4090 graphic card.

4.2 | Results

In this section, we conduct experiments to evaluate the robust cooperative MARL guidance policy. Two sets of tests are performed. In Scenario 1, the proposed R-CCMARL algorithm, the variant risk-only model, and the well-known MARL algorithm MAPPO [14] are trained and evaluated under the same hyper-parameter settings and environmental conditions. In Scenario 2, R-CCMARL and ERNIE [52] are trained and evaluated under the same hyper-parameter settings and environmental conditions as well. Note that, to adapt ERNIE to our setting, we replace its network with ours, while still following its robust training formulation.

4.2.1 | Robustness MARL Evaluation

To evaluate the performance of our proposed robust multi-agent guidance algorithm under observation perturbations, we choose a difficult attack configuration, $\varepsilon = 0.1, iteration = 20$ and multiple attack strategies. As mentioned before, one of the challenges in robust MARL system against adversarial attacks is the number of agents being attacked is unknown. Therefore, four attack strategies are conducted: attack on each single agent and attack on all agents, as well as no attack to make sure the system works well in normal situation at the same time. For each category all models are evaluated for 50 episodes.

TABLE 1 | Hyper-parameters for the experiments.

Hyper-parameter	Value
Discount factor γ	0.99
RL network learning rate	$1e^{-4} \sim 1e^{-6}$
Lagrange multiplier learning rate	$1e^{-2} \sim 1e^{-4}$
Initial Lagrange multiplier α	0.01
Initial Lagrange multiplier β	0.01
Memory size	5000
Clipping ratio	0.2
Constraint threshold μ	0.001
Constraint threshold ϵ	0.1

We evaluate MAPPO, ERNIE, the only-risk model where the constraint for robust cooperative MARL is not implemented, and the full proposed algorithm R-CCMARL in five evaluation metrics. The evaluation metrics are defined as follows:

1. Average Team Reward: the average accumulative reward of the MARL system per episode.
2. Average Individual Reward: the average accumulative reward of each agent per episode.
3. Average Individual Success Rate: current success times to reach the goal position over current episodes.
4. Average Team Risk: the average accumulative safety of the MARL system per episode.
5. Average Individual Risk: the average accumulative safety of each agent per episode.

Note that despite the concerns about bias in evaluation due to incorporating extra, nonattacked numeric data as inputs to enhance stability and performance, our results show substantial impacts on the MAPPO model, resulting in low success rate.

Scenario 1: Our proposed method R-CCMARL effectively handles strong adversarial attacks under the same conditions, indicating that the additional inputs do not compromise evaluation outcomes. In Table 2, R-CCMARL leads all the categories in success rate, long-term return, and long-term risk under every attack strategy. In terms of success rate, MAPPO delivers ok performance when Agent 1 is attacked. Specifically, under this setting, agent 1 reaches 80% of success, while Agent 2 and Agent 3 remain 50% and 60%. However, in other situations,

TABLE 2 | Detailed individual performance comparison in various attack conditions (Scene 1).

		Attack on agent 1			Attack on agent 2			Attack on agent 3			Attack on all		
Method	Metrics	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3
MAPPO	Success rate	0.80	0.50	0.60	0.80	0.00	0.40	0.60	0.10	0.10	0.40	0.00	0.00
	Return	26.30	42.63	37.73	25.68	34.78	37.46	28.38	47.70	37.01	26.06	29.41	34.59
	Risk	−10.26	−7.16	−1.12	−6.37	−4.10	−1.16	−8.26	−4.57	−5.75	−9.89	−4.94	−2.31
Risk-only	Success rate	0.88	0.70	0.82	0.84	0.24	0.78	0.82	0.60	0.74	0.78	0.08	0.70
	Return	34.65	51.26	49.75	34.37	37.11	51.10	36.09	59.05	43.83	32.77	34.21	40.62
	Risk	−6.10	−4.14	−0.34	−3.53	−2.61	−0.37	−6.64	−3.94	−4.54	−8.37	−3.94	−1.79
R-CCMARL	Success rate	0.98	0.92	0.94	0.98	0.82	0.90	0.96	0.88	0.88	0.90	0.78	0.84
	Return	36.37	63.84	52.13	34.98	56.81	50.05	40.53	69.74	56.07	37.59	45.21	52.77
	Risk	−5.61	−3.85	−0.41	−3.51	−1.70	−0.42	−5.77	−3.38	−4.39	−6.97	−4.05	−1.45

where Agent 2 is attacked, Agent 3 is attacked and all agents are attacked, MAPPO shows weak robustness. Agent 2 is not able to finish the intersection passing task, leaving only 10% of success rate at most. Agent 3 is a bit better than Agent 2 but gets 40% at most. Lacking adversarial training and risk awareness, MAPPO shows no defence to strong adversarial perturbations. The risk-only method, though is not involved in adversarial training, shows certain improvement over MAPPO, the performance is far from satisfaction, especially the attack is implemented on Agent 2. In contrast, R-CCMARL benefiting from the multi-agent interaction modelling (mean-field information sharing), long-term risk minimisation, and the constraint adversarial optimisation with CBF, remain resilient to the strong perturbations, delivering robust performance even in the Attack on All condition. For robust MARL algorithms, it is also important to maintain performance in normal conditions while capable of mitigating the strong attacks. Table 3 shows the performance comparisons in the nonattacked environments. For success rate, the full proposed method achieves perfect scores for Agents 1 and 3, and nearly perfect for Agent 2, while MAPPO performs worst, particularly for Agent 2. In terms of return, R-CCMARL leads with higher returns, especially for Agents 2 and 3, indicating better performance compared to the other models. Additionally, R-CCMARL consistently demonstrates lower risk values, particularly for Agents 2 and 3, suggesting that it not only yields high returns but does so more safely than MAPPO and Risk-Only, making it the most robust method overall.

TABLE 3 | Detailed individual performance comparison in normal (no attack) observation condition (Scene 1).

Method	Metrics	Agent 1	Agent 2	Agent 3
MAPPO	Success rate	1.00	0.66	0.78
	Return	27.71	57.89	47.34
	Risk	−3.75	−5.66	−1.79
Risk-only	Success rate	1.00	0.92	0.98
	Return	38.48	61.29	57.30
	Risk	−1.98	−4.33	−1.12
R-CCMARL	Success rate	1.00	0.94	1.00
	Return	36.34	95.02	70.53
	Risk	−1.91	−1.85	−0.43

Figure 6 illustrates the performance of these methods under the five attack situations in a more comprehensive way. The thin line is the mean value of each category and the standard deviation is visualised around the mean value to show stability. Overall as it can be seen, in terms of average accumulative rewards, the risk-only model outperform MAPPO in each attack situations, indicating the effeteness of the risk assessment function and the additional long-term risk minimisation. The full algorithm R-CCMARL further boost performance over MAPPO and risk-only model. Specifically, in normal condition showed in Figure 6a, MAPPO is at 136.15 per episode. In contrast, R-CCMARL achieves around 203.04 accumulative reward an episode, which represents 49.13% gain from MAPPO. Comparing to the risk-only model, we achieve 28.5% improvement. Figure 6a states that our robust cooperative MARL improves the guidance performance even without considering adversarial conditions. Figure 6b–d show the performance when different individual agent is under attacks. It is evident that in a MARL system, attacking different agents has varying effects,

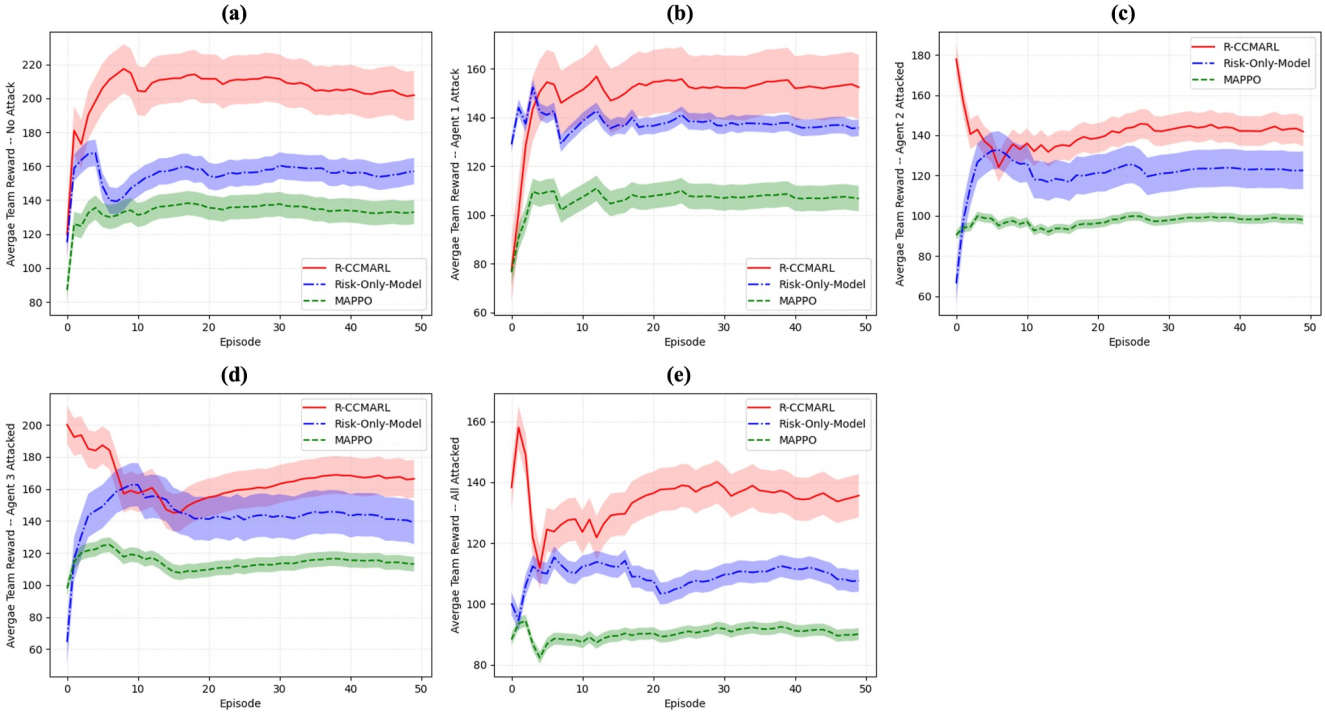

FIGURE 6 | Detailed comparisons between our proposed method, the variant and MAPPO in Scenario 1. We evaluate the multi-agent system by attacking different agents ((a) No agents attacked, (b) Agent 1 attacked, (c) Agent 2 attacked, (d) Agent 3 attacked and (e) All agents attacked) with the same strength $\varepsilon = 0.1$, $iteration = 20$. The curve line is the average team reward while transparent area indicates standard deviation.

revealing the different levels of importance each agent holds in the system. In Figure 6b, when agent 1 is attacked, the average team reward of the three methods drop by 25.2%, 13.2% and 23.1%, resulting in rewards of 152.2, 137.1 and 104.9. When Agent 2 is attacked, the performance further reduced. In Figure 6c MAPPO achieve 99.2 in the long-term reward, and the risk-only model achieves 122.4. R-CCMARL leads the board at 142.3 marking 43.4% and 16.3% improvement over MAPPO and risk-only model. In Figure 6d, Agent 3 is attacked. R-CCMARL obtain 168.6 of accumulative reward while the risk-only model and MAPPO are at 140.1 and 116.4, showing 20.3% and 44.8% of improvement respectively. Among the individual agent perturbations, attacks on Agent 2 have the most significant impact on the entire system, making Agent 2 the most vulnerable. We further increase the level of attacks by applying perturbations on every agent. In Figure 6e, R-CCMARL maintains the reward of 136.9. The risk-only model and MAPPO, both lacking constraint adversarial training, are not able to handle the situations, reaching only 106.2 and 91.7. These results demonstrate the effectiveness of our proposed constraint objective function in enhancing the robustness of the MARL system.

Scenario 2: As the complexity and difficulty of the environment increase, there is a decline in overall success rate. However, R-CCMARL consistently outperforms the ERNIE method and remains capable of handling most challenging cases. In Table 4, under normal conditions, R-CCMARL achieves high success rates of 80%, 76% and 80% for Agents 1, 2 and 3, respectively—comparable to or slightly exceeding those of ERNIE, especially for Agent 3. It also delivers higher long-term returns, with Agent 3 reaching 193.67 compared to 185.53 of ERNIE. Additionally, R-CCMARL demonstrates lower risk values across all agents, such as −0.17 for Agent 1 versus −7.31 under ERNIE. These results indicate that even in regular settings, R-CCMARL offers not only strong task completion but also improved stability and risk control. In Table 5, with adversarial attack introduced, R-CCMARL maintains its advantage. For example, when Agent 3 is attacked, R-CCMARL achieves success rates of 70%, 66% and 56% across the three agents, while ERNIE drops to 60%, 60% and 50%. The gap becomes more significant in the most challenging case when all agents are attacked, where R-CCMARL sustains 56%, 58% and 42% success rates, while ERNIE falls sharply to 30%, 18% and 24%. R-CCMARL also consistently achieves comparable or higher returns in various settings than ERNIE, especially in attack on all situation, where ERNIE starts to receive negative reward for agent 1 and Agent 2.

TABLE 4 | Detailed individual performance comparison in normal (no attack) observation condition (Scene 2).

Method	Metrics	Agent 1	Agent 2	Agent 3
ERNIE	Success rate	0.81	0.74	0.68
	Return	66.50	71.33	185.53
	Risk	−7.31	−0.70	−9.53
R-CCMARL	Success rate	0.80	0.76	0.80
	Return	87.32	81.64	193.67
	Risk	−0.17	−0.83	−11.83

In Figure 7, the rewards of the MARL systems are visualised as the team reward. In the normal condition, shown in Figure 7a, ERNIE achieves a team return of 323.36, while R-CCMARL demonstrates a 12% improvement, reaching 362.63. This performance advantage becomes more evident under adversarial settings. In Figure 7b, where Agent 1 is attacked, R-CCMARL attains 159.90 in team reward, which is 25% higher than 127.85 of ERNIE. In the more challenging scenario in Figure 7c, R-CCMARL achieves 147.01, outperforming 126.12 of ERNIE by approximately 17%. In Figure 7d, R-CCMARL reaches 163.74, showing a 32% improvement over 123.76 by ERNIE. Most significantly in Figure 7e where all agents are attacked, the performance of ERNIE drops sharply to 55.93 while R-CCMARL achieves 126.99, an impressive 127% increase. Note that in Figure 7a,e, the sharp initial drop in the team rewards simply indicates a failure case occurred at an early stage of the evaluation.

To further demonstrate the impact of adversarial attacks on the multi-agent systems, we evaluate the deviation in actions generated by the models under the strongest attack condition compared to the normal condition in scenario 2. The action deviation is assessed in terms of steering and throttle deviations. As shown in Figure 8, the blue, red, and green curves represent the action deviations of Agent 1, Agent 2 and Agent 3, respectively. The agents show different time steps in Figure 8, as they complete each episode at varying steps; for instance, in a successful scenario, Agent 1 finishes first, followed by Agent 2, and finally, Agent 3. This ordering is the result of the autonomous

TABLE 5 | Detailed individual performance comparison in various attack conditions (Scene 2).

		Attack on agent 1			Attack on agent 2			Attack on agent 3			Attack on all		
Method	Metrics	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3
ERNIE	Success rate	0.66	0.58	0.52	0.66	0.58	0.52	0.60	0.60	0.50	0.30	0.18	0.24
	Return	21.97	23.22	82.66	29.01	30.02	67.09	18.65	35.81	69.30	−1.18	−7.08	64.19
	Risk	−0.04	−0.28	−5.33	−0.03	−1.06	−5.28	−0.04	−0.27	−1.19	−0.01	−0.60	−14.70
R-CCMARL	Success rate	0.66	0.68	0.70	0.58	0.64	0.70	0.70	0.66	0.56	0.56	0.58	0.42
	Return	34.37	30.94	94.59	22.31	35.22	89.45	39.14	28.84	95.76	35.18	29.09	62.74
	Risk	−5.61	−3.85	−0.41	−0.37	−0.31	−8.7	−0.33	−0.17	−12.33	−0.38	−0.28	−10.35

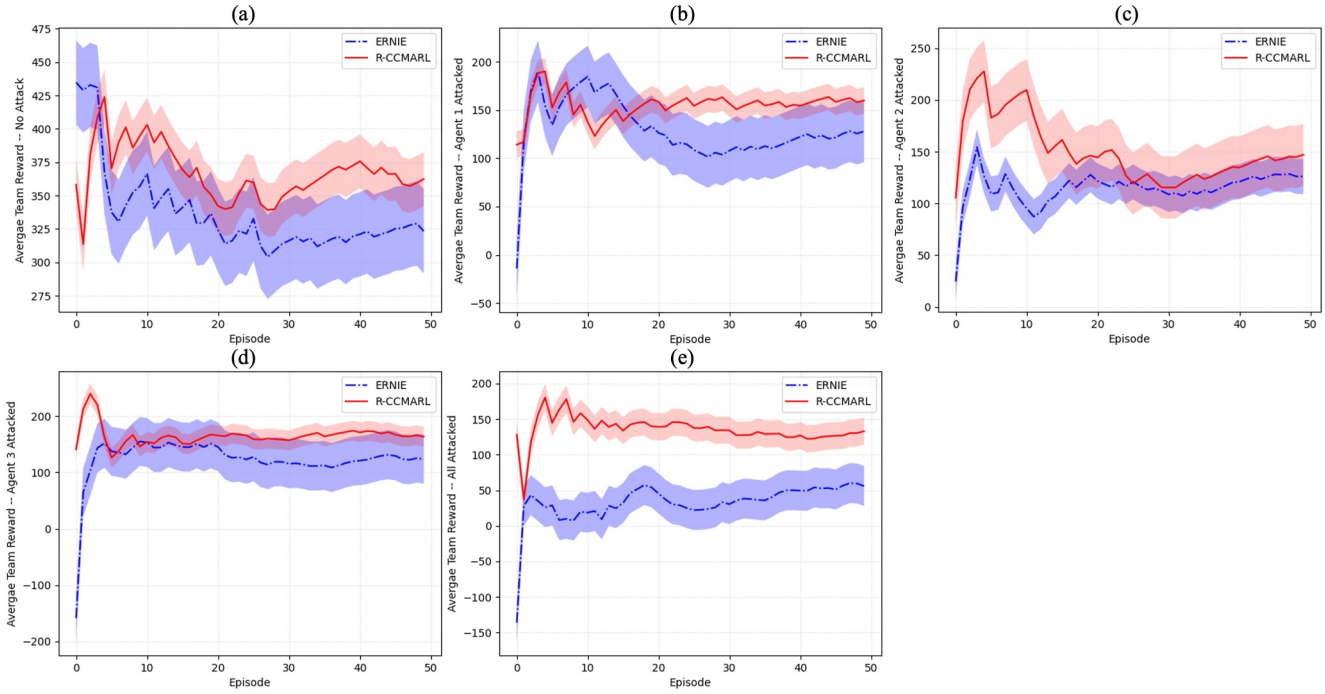

FIGURE 7 | Detailed comparisons between our proposed method R-CCMARL and ERNIE in Scenario 2. We evaluate the multi-agent system by attacking different agents ((a) No agent attacked, (b) Agent 1 attacked, (c) Agent 2 attacked, (d) Agent 3 attacked and (e) All agents attacked) with the same strength $\varepsilon = 0.1$, $iteration = 20$. The curve line is the average team reward while transparent area indicates standard deviation.

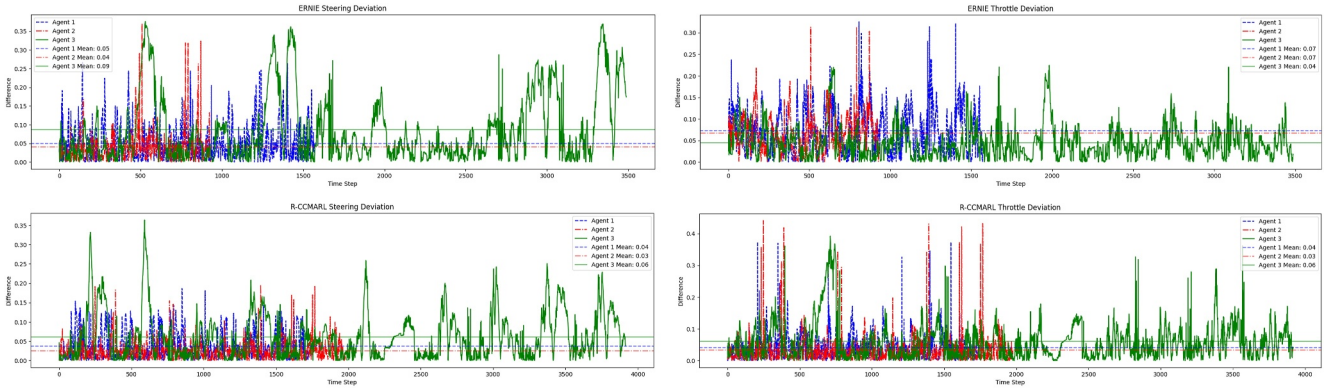

FIGURE 8 | Action deviation comparisons under all agents attacked condition and no attack condition in Scenario 2.

negotiation of the MARL system, rather than being predefined. It is evident that, in both steering and throttle actions, R-CCMARL exhibits smaller deviations between attacked and normal actions compared to ERNIE, particularly in steering control. In throttle control, while R-CCMARL shows some higher spikes than ERNIE, its overall deviation remains at a relatively lower level. These lower action deviations indicate stronger resilience to adversarial attacks, contributing to enhanced robustness in multi-agent scenarios.

Overall, the results from both scenarios highlight the robustness and effectiveness of the proposed R-CCMARL across varying levels of environmental complexity and adversarial intensity, demonstrating its efficiency in handling challenging multi-agent tasks.

4.2.2 | Risk Minimisation Analysis

The long-term risk minimisation with risk assessment and CBF constraint optimisation together with robust MARL shows significant robustness to perturbations in comparison with the nonrobust MAPPO. When focussing solely on the risk metric, R-CCMARL and its variant risk-only model consistently demonstrates superior performance in mitigating risk compared to MAPPO across all attack scenarios. As demonstrated in Table 2, in the 'Attack on Agent 1' scenario, MAPPO shows the highest risk at −10.26, while the risk-only model reduces it to −6.10, and the proposed R-CCMARL achieves the lowest risk at −5.61. Similarly, in the more challenging 'Attack on All' scenario, the risk of MAPPO remains high with values like −9.89 for Agent 1, while the risk-only model shows improvement, reducing it to

−8.37. However, our approach significantly outperforms both by further reducing the risk to −6.97. Combining the results shown in Tables 2 and 3, across all situations, whether under heavy attacks or in normal conditions, R-CCMARL consistently shows lower risk values, indicating its greater resilience to adversarial conditions and ability to minimise performance degradation more effectively than the other models. This highlights the strength of our proposed constrained objective function in enhancing the robustness of the multi-agent reinforcement learning system under adversarial attacks.

Figure 9 illustrates the team risks of the multi-agent system under no-attack and all-attack conditions in scenario 1 and scenario 2. It can be observed that while certain agents in R-CCMARL tend to have higher individual risks compared to the other two models, the overall team risk of R-CCMARL consistently remains the smallest. This highlights the efficiency of the risk minimisation mechanism. Both R-CCMARL and its variant risk-only model demonstrate superior risk control capabilities compared to MAPPO. Notably, in Figure 9a–c, R-CCMARL shows even greater effectiveness, as the CBF constraint introduced identifies a smaller, safer action subset compared to simple risk minimisation. In Figure 9b, the risk-only model outperforms MAPPO under heavy attack conditions, indicating that risk minimisation identifies an action subset that overlaps with the adversarial toleration subset. This reveals the relevance between our proposed risk assessment to adversarial defence mechanism, with the CBF further enhancing its effectiveness. It is worth noting that in Figure 9d, R-CCMARL and ERNIE exhibit nearly identical average long-term risk. This occurs because ERNIE fails in most episodes, leading to shorter episode lengths. As the agents stop, the risk becomes zero, which results in a similar team risk value to R-CCMARL.

When considering the combined performance across all three metrics—success rate, return, and risk, it is evident that our method R-CCMARL offers the most robust results. The lower risk values, paired with higher success rates and long-term returns, demonstrate the effectiveness of our risk assessment approach. Notably, the risk is positively correlated with both the success rate and long-term return: as the risk decreases, the success rate and returns improve. This relationship highlights how well R-CCMARL handles adversarial conditions, with reduced risks translating into better overall performance.

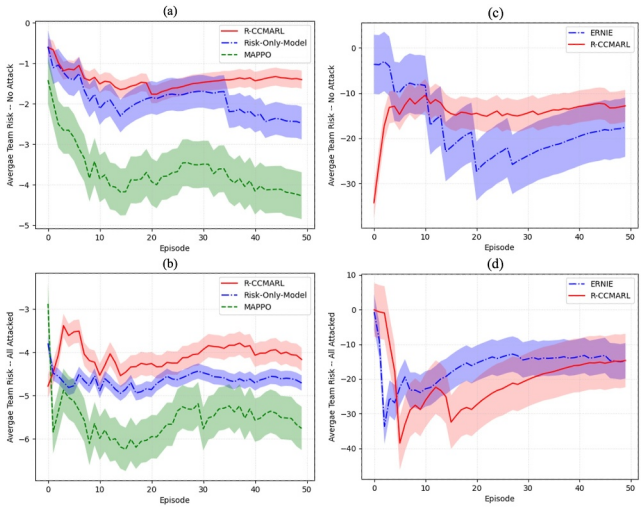

FIGURE 9 | The long-term episodic average team risk comparison in no agent attacked and all agents attacked situations. (a) and (b) indicate the risks in experiment scenario 1, while (c) and (d) show the risks in experiment scenario 2.

5 | Conclusion

In this paper, we introduce a novel approach for deep reinforcement learning based robust cooperative multi-agent autonomous driving against observation perturbations. The realm of autonomous driving and decision-making models based on deep reinforcement learning is constantly challenged by safety issues arising from inevitable sensor failures or transitions between domains. These challenges bear a resemblance to adversarial attacks, as they potentially disrupt or compromise the agent's ability to make accurate decisions in critical situations. To approximate the worst-case perturbation, we develop a white-box optimal method for generating adversaries, which has full access to the parameters of the RL agent model, enabling the creation of training samples for robust reinforcement learning. This method is based on fast gradient sign method (FGSM), incorporating with our collision risk maximum formulation. To more efficiently attack the RL agent, we use iteration on FGSM rather than simply enlarge the attack strength parameter ε to reduce the visibility of the adversarial examples. After obtaining the perturbed observation, we introduce a Stochastic Games with perturbation to formulate the multi-agent intersection passing for autonomous driving. We develop a mean-filed theory supported information sharing structure to enable global state and interaction awareness. An efficient risk assessment is proposed and utilised in the long-term risk minimisation and as the control barrier function to tolerated bounded perturbations also provides safety rewards as feedback to MARL and help shape the policy. Furthermore, a divergence-based regulariser term is applied to mimic the performance gap between nonadversarial states and adversarial states. Experiment verifies the effectiveness of our proposed method, and the advantage of the risk minimisation module and the constraint optimisation in solving the multi-agent intersection problems.

Author Contributions

Chuyao Wang: conceptualization, formal analysis, investigation, methodology, software, validation, visualisation, writing. **Ziwei Wang:** conceptualization, software, writing. **Nabil Aouf:** conceptualization, supervision.

Conflicts of Interest

One of the author Nabil Aouf is the editor for the special issue Trust through eXplAInability (XAI), Robustness and Verification of Autonomous Systems.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. J. Chen, B. Yuan, and M. Tomizuka, "Model-Free Deep Reinforcement Learning for Urban Autonomous Driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (IEEE, 2019), 2765–2771.

2. C. Wang and N. Aouf, "Deep Reinforcement Learning Based Planning for Urban Self-Driving With Demonstration and Depth Completion," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)* (IEEE, 2021), 962–967.

3. J. Chen, S. E. Li, and M. Tomizuka, "Interpretable End-To-End Urban Autonomous Driving With Latent Deep Reinforcement Learning," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 6 (2021): 5068–5078, https://doi.org/10.1109/tits.2020.3046646.

4. I. B. Viana, H. Kanchwala, K. Ahiska, and N. Aouf, "A Comparison of Trajectory Planning and Control Frameworks for Cooperative Autonomous Driving," *Journal of Dynamic Systems, Measurement, and Control* 143, no. 7 (2021): 071002, https://doi.org/10.1115/1.4049554.

5. H. Kanchwala, I. Bezerra Viana, and N. Aouf, "Cooperative Path-Planning and Tracking Controller Evaluation Using Vehicle Models of Varying Complexities," *Proceedings of the Institution of Mechanical Engineers - Part C: Journal of Mechanical Engineering Science* 235, no. 16 (2021): 2877–2896, https://doi.org/10.1177/0954406220945468.

6. M. A. Shah and N. Aouf, "3D Cooperative Pythagorean Hodograph Path Planning and Obstacle Avoidance for Multiple Uavs," in *2010 IEEE 9th International Conference on Cyberntic Intelligent Systems* (IEEE, 2010), 1–6.

7. R. Lowe, Y. I. Wu, A. Tamar, et al., "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," *Advances in Neural Information Processing Systems* 30 (2017): 6382–6393.

8. P. Sunehag, G. Lever, A. Gruslys, et al., "Value-Decomposition Networks for Cooperative Multi-Agent Learning," *arXiv Preprint* arXiv:1706.05296 (2017).

9. C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: A Modular Learning Framework for Mixed Autonomy Traffic," *IEEE Transactions on Robotics* 38, no. 2 (2021): 1270–1286, https://doi.org/10.1109/tro.2021.3087314.

10. S. Omidshafiei, J. Pazis, C. Amato, et al., *Deep Decentralised Multi-Task Multi-Agent Reinforcement Learning Under Partial Observability International Conference on Machine Learning* (PMLR, 2017), 2681–2690.

11. J. Hao, T. Yang, H. Tang, et al., "Exploration in Deep Reinforcement Learning: From Single-Agent to Multiagent Domain," *IEEE Transactions on Neural Networks and Learning Systems* 35, no. 7 (2023): 8762–8782, https://doi.org/10.1109/tnnls.2023.3236361.

12. T. Rashid, M. Samvelyan, C. S. De Witt, et al., "Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning," *Journal of Machine Learning Research* 21, no. 178 (2020): 1–51.

13. Z. Zhou, G. Liu, and M. Zhou, "A Robust Mean-Field Actor-Critic Reinforcement Learning Against Adversarial Perturbations on Agent States," *IEEE Transactions on Neural Networks and Learning Systems* 35, no. 10 (2023): 14370–14381, https://doi.org/10.1109/tnnls.2023.3278715.

14. C. Yu, A. Velu, E. Vinitsky, et al., "The Surprising Effectiveness of Ppo in Cooperative Multi-Agent Games," *Advances in Neural Information Processing Systems* 35 (2022): 24611–24624.

15. B. R. Kiran, I. Sobh, V. Talpaert, et al., "Deep Reinforcement Learning for Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 6 (2021): 4909–4926, https://doi.org/10.1109/tits.2021.3054625.

16. A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," *arXiv Preprint* arXiv:1611.01236 (2016).

17. A. Madry, A. Makelov, L. Schmidt, et al., "Towards Deep Learning Models Resistant to Adversarial Attacks," *arXiv Preprint* arXiv:1706.06083 (2017).

18. G. Li, Y. Yang, S. Li, X. Qu, and N. Lyu, "Decision Making of Autonomous Vehicles in Lane Change Scenarios: Deep Reinforcement Learning Approaches With Risk Awareness," *Transportation Research Part C: Emerging Technologies* 134 (2022): 103452, https://doi.org/10.1016/j.trc.2021.103452.

19. A. Dosovitskiy, G. Ros, F. Codevilla, et al., *CARLA: An Open Urban Driving Simulator Conference on Robot Learning* (PMLR, 2017), 1–16.

20. S. Iqbal and F. Sha, "Actor-Attention-Critic for Multi-Agent Reinforcement Learning," in *International Conference on Machine Learning* (PMLR, 2019), 2961–2970.

21. Y. Liu, W. Wang, Y. Hu, J. Hao, X. Chen, and Y. Gao, "Multi-Agent Game Abstraction via Graph Attention Neural Network," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, no. 5 2020): 7211–7218, https://doi.org/10.1609/aaai.v34i05.6211.

22. Y. Yang, J. Hao, G. Chen, et al., *Q-Value Path Decomposition for Deep Multiagent Reinforcement Learning International Conference on Machine Learning* (PMLR, 2020), 10706–10715.

23. J. Wang, Z. Ren, T. Liu, et al., "Qplex: Duplex Dueling Multi-Agent Q-Learning," *arXiv Preprint* arXiv:2008.01062 (2020).

24. Y. Yang, R. Luo, M. Li, et al., *Mean Field Multi-Agent Reinforcement Learning International Conference on Machine Learning* (PMLR, 2018), 5571–5580.

25. V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-Level Control Through Deep Reinforcement Learning," *Nature* 518, no. 7540 (2015): 529–533, https://doi.org/10.1038/nature14236.

26. D. Silver, A. Huang, C. J. Maddison, et al., "Mastering the Game of Go With Deep Neural Networks and Tree Search," *Nature* 529, no. 7587 (2016): 484–489, https://doi.org/10.1038/nature16961.

27. Z. Q. Zhao, P. Zheng, S. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems* 30, no. 11 (2019): 3212–3232, https://doi.org/10.1109/tnnls.2018.2876865.

28. Q. Wang, F. Ju, H. Wang, et al., "Multi-Agent Reinforcement Learning for Ecological Car-Following Control in Mixed Traffic," *IEEE Transactions on Transportation Electrification* 10, no. 4 (2024): 8671–8684, https://doi.org/10.1109/tte.2024.3383091.

29. D. Li, D. Zhao, Q. Zhang, and Y. Chen, "Reinforcement Learning and Deep Learning Based Lateral Control for Autonomous Driving," *IEEE Computational Intelligence Magazine* 14, no. 2 (2019): 83–98, https://doi.org/10.1109/mci.2019.2901089.

30. M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, and R. Ke, "Safe, Efficient, and Comfortable Velocity Control Based on Reinforcement Learning for Autonomous Driving," *Transportation Research Part C: Emerging Technologies* 117 (2020): 102662, https://doi.org/10.1016/j.trc.2020.102662.

31. H. Lin, C. Lyu, Y. He, Y. Liu, K. Gao, and X. Qu, "Enhancing State Representation in Multi-Agent Reinforcement Learning for Platoon-Following Models," *IEEE Transactions on Vehicular Technology* 73, no. 8 (2024): 12110–12114, https://doi.org/10.1109/tvt.2024.3373533.

32. Z. Huang, J. Wu, and C. Lv, "Efficient Deep Reinforcement Learning With Imitative Expert Priors for Autonomous Driving," *IEEE Transactions on Neural Networks and Learning Systems* 34, no. 10 (2022): 7391–7403, https://doi.org/10.1109/tnnls.2022.3142822.

33. A. Boloor, K. Garimella, X. He, C. Gill, Y. Vorobeychik, and X. Zhang, "Attacking Vision-Based Perception in End-To-End Autonomous Driving Models," *Journal of Systems Architecture* 110 (2020): 101766, https://doi.org/10.1016/j.sysarc.2020.101766.

34. X. He, H. Yang, Z. Hu, and C. Lv, "Robust Lane Change Decision Making for Autonomous Vehicles: An Observation Adversarial Reinforcement Learning Approach," *IEEE Transactions on Intelligent Vehicles* 8, no. 1 (2022): 184–193, https://doi.org/10.1109/tiv.2022.3165178.

35. V. Behzadan and A. Munir, "Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks Machine Learning and Data Mining," in *Pattern Recognition: 13th International Conference, MLDM 2017, New York, NY, USA, July 15–20, 2017, Proceedings 13* (Springer International Publishing, 2017), 262–275.

36. A. Pattanaik, Z. Tang, S. Liu, et al., "Robust Deep Reinforcement Learning With Adversarial Attacks," *arXiv Preprint* arXiv:1712.03632 (2017).

37. S. Huang, N. Papernot, I. Goodfellow, et al., "Adversarial Attacks on Neural Network Policies," *arXiv Preprint* arXiv:1702.02284 (2017).

38. J. Kos and D. Song, "Delving Into Adversarial Attacks on Deep Policies," *arXiv Preprint* arXiv:1705.06452 (2017).

39. Y. C. Lin, Z. W. Hong, Y. H. Liao, M. L. Shih, M. Y. Liu, and M. Sun, "Tactics of Adversarial Attack on Deep Reinforcement Learning Agents," *arXiv Preprint* arXiv:1703.06748 (2017): 3756–3762, https://doi.org/10.24963/ijcai.2017/525.

40. H. Zhang, H. Chen, C. Xiao, et al., "Robust Deep Reinforcement Learning Against Adversarial Perturbations on State Observations," *Advances in Neural Information Processing Systems* 33 (2020): 21024–21037.

41. H. Zhang, H. Chen, D. Boning, et al., "Robust Reinforcement Learning on State Observations With Learned Optimal Adversary," *arXiv Preprint* arXiv:2101.08452 (2021).

42. T. Oikarinen, W. Zhang, A. Megretski, et al., "Robust Deep Reinforcement Learning Through Adversarial Loss," *Advances in Neural Information Processing Systems* 34 (2021): 26156–26167.

43. A. Kumar, A. Levine, and S. Feizi, "Policy Smoothing for Provably Robust Reinforcement Learning," *arXiv Preprint* arXiv:2106.11420 (2021).

44. J. Lin, K. Dzeparoska, S. Q. Zhang, et al., "On the Robustness of Cooperative Multi-Agent Reinforcement Learning," in *2020 IEEE Security and Privacy Workshops (SPW)* (IEEE, 2020), 62–68.

45. L. S. Shapley, "Stochastic Games," *Proceedings of the National Academy of Sciences* 39, no. 10 (1953): 1095–1100, https://doi.org/10.1073/pnas.39.10.1953.

46. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv Preprint* arXiv:1412.6572 (2014).

47. A. Kurakin, I. J. Goodfellow, and S. Bengio, *Adversarial Examples in the Physical World Artificial Intelligence Safety and Security* (Chapman and Hall/CRC, 2018), 99–112.

48. J. Schulman, P. Moritz, S. Levine, et al., "High-Dimensional Continuous Control Using Generalised Advantage Estimation," *arXiv Preprint* arXiv:1506.02438 (2015).

49. C. Wang and N. Aouf, "Explainable Deep Adversarial Reinforcement Learning Approach for Robust Autonomous Driving," *IEEE Transactions on Intelligent Vehicles* (2024): 1–13, https://doi.org/10.1109/tiv.2024.3379367.

50. I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems* (Cambridge University Press, 2011).

51. I. Csiszár, "I-Divergence Geometry of Probability Distributions and Minimisation Problems," *Annals of Probability* (1975): 146–158.

52. A. Bukharin, Y. Li, Y. Yu, et al., "Robust Multi-Agent Reinforcement Learning via Adversarial Regularisation: Theoretical Foundation and Stable Algorithms," *Advances in Neural Information Processing Systems* 36 (2024): 68121–68133.

53. A. D. Ames, X. Xu, J. W. Grizzle, et al., "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control* 62, no. 8 (2016): 3861–3876.