2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Spatial-Temporal-Aware Safe Multi-Agent Reinforcement Learning of Connected Autonomous Vehicles in Challenging Scenarios

Zhili Zhang Songyang Han Jiangwei Wang Fei Miao

***Abstract*— Communication technologies enable coordination among connected and autonomous vehicles (CAVs). However, it remains unclear how to utilize shared information to improve the safety and efficiency of the CAV system in dynamic and complicated driving scenarios. In this work, we propose a framework of constrained multi-agent reinforcement learning (MARL) with a parallel Safety Shield for CAVs in challenging driving scenarios that includes unconnected hazard vehicles. The coordination mechanisms of the proposed MARL include information sharing and cooperative policy learning, with Graph Convolutional Network (GCN)-Transformer as a spatial-temporal encoder that enhances the agent's environment awareness. The Safety Shield module with Control Barrier Functions (CBF)-based safety checking protects the agents from taking unsafe actions. We design a constrained multi-agent advantage actor-critic (CMAA2C) algorithm to train safe and cooperative policies for CAVs. With the experiment deployed in the CARLA simulator, we verify the performance of the safety checking, spatial-temporal encoder, and coordination mechanisms designed in our method by comparative experiments in several challenging scenarios with unconnected hazard vehicles. Results show that our proposed methodology significantly increases system safety and efficiency in challenging scenarios.**

I. Introduction

Wireless communication technologies such as WiFi and 5G cellular networks enable vehicle-to-everything (V2X) communication and help the autonomous vehicle to get extra information about the driving environment beyond its sensing capability [1], [2]. Shared information captured by the onboard sensors such as cameras and LIDARs can be used to improve connected autonomous vehicles' (CAVs) decision-making [3], [4], [5]. Shared basic safety messages benefit the coordination and control decisions of CAVs in scenarios such as intersections and lane-merging [6], [7].

However, it is not clear how information sharing benefits connected autonomous vehicles in challenging scenarios. Without communication or coordination, it is difficult for CAVs to react to traffic-rule-violating behaviors or sudden speeding/braking maneuvers taken by unconnected hazard vehicles in mixed traffic conditions as in Fig. 1. When an autonomous vehicle gets extra knowledge about the environment via coordinated V2X communication, how to design a neural network structure to utilize the shared information with spatial and temporal features and how to make prudent decisions to improve collaborative safety remain unsolved.

This work was supported by NSF 1849246, NSF 1932250, NSF 2047354 grants. Z. Zhang, S. Han, and F. Miao are with the Department of Computer Science and Engineering, J. Wang is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs Mansfield, CT, USA 06268. Email: {zhili.zhang, songyang.han, jiangwei.wang, fei.miao}@uconn.edu.

Fig. 1. *Intersection* ((a),(b),(c)) and *Highway* ((d),(e),(f)) scenarios: one hazard vehicle runs the red light in *Intersection* scenario and one takes a sudden hard-brake in *Highway* scenario. 1a, 1d: scenario initialization; 1b, 1e: successful cases of collaborative collision-avoidance from test runs of our method; 1c, 1f: collision cases from test runs of baseline model. Connected autonomous vehicles (CAVs) are in green; unconnected vehicles (UCVs) are in red; unconnected hazard vehicles (HAZV) are in red with yellow triangle marks. Without the safety shield or coordination, CAVs are likely to collide with HAZV or other vehicles as in 1c, 1f.

In this work, we design a spatial-temporal-aware constrained MARL framework with *Safety Shield* for cooperative policy-learning of CAVs, to improve safety and efficiency of the system utilizing V2X communication-based information-sharing. In particular, we consider challenging driving scenarios with potential unconnected hazard vehicles (HAZV). We adopt the prevailing Graph Convolutional Network (GCN) and Transformer networks as spatial-temporal scene encoders (Fig. 2a) for agents to raise their situation awareness (Sec. III), and the actor-critic-cost neural network structure of the proposed Constrained Multi-Agent Advantage Actor-Critic (CMAA2C) method (Sec. IV-A). The complicated dynamics and interactions among CAVs under challenging scenarios provide strong motivations to design a *Safety Shield* based on Control Barrier Functions (CBFs) for the policy (Sec. IV-B). We further introduce the cooperative training scheme of the CMAA2C algorithm in Sec. IV-C (Fig 2b). In summary, the main contributions of this work are:

- We propose a Constrained Multi-Agent Advantage Actor Critic method with a *Safety Shield* to improve safety and efficiency of the CAV system in challenging scenarios. The coordination mechanisms include information-sharing and cooperative policy-learning in CMAA2C.
- We design a GCN-Transformer encoder for the neural

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10161216

network structure of CMAA2C to utilize the shared spatial and temporal information among CAVs and improve the situation awareness of CAVs.
- We validate that the proposed CMAA2C MARL framework significantly improves the collision-free rate and overall returns of the CAV system with experiments. Our results show that cooperation among CAVs, the *Safety Shield*, and the GCN-Transformer encoder design all contribute to the improvement.

II. Related Work

a) Planning and Control of Autonomous Vehicles: To learn the output controls for steering and throttle directly based on observed environment, end-to-end learning is designed in CNN-based supervised learning [8] and CBF-based Deep Reinforcement Learning [9], when considering only lane-keeping behaviors. Another popular way is separating the learning and control phases. Learning methods can give high-level decisions, such as "go left", "go straight" [10], or "yield" [11]. It also works to first extract image features and then apply control upon these features [12]. However, the mentioned works do not consider connections between CAVs, while we consider how CAVs should use information sharing to improve the safety and efficiency of the system, and design an MARL-based algorithm such that CAVs cooperatively take actions under challenging driving scenarios.

b) GCN, Transformer and Deep MARL: It has not been addressed how to design a specific neural network structure utilizing communications among CAVs to improve the system safety or efficiency. Recent advances like GCN [13] and Transformer [14], [15] show their advantages in acquiring spatial and temporal properties from data and we use them to encode the spatial-temporal information of driving scenarios. To the best of our knowledge, we are the first to design a GCN-Transformer-based deep constrained MARL framework using shared information among CAVs. We validate that this design improves the safety rates and total rewards for CAVs in challenging scenarios with traffic hazards.

c) Constrained MDP and Safe RL: Existing multi-agent reinforcement learning (MARL) literature [16], [17], [18], [19] has not fully solved the challenges for CAVs. Constrained Markov Decision Process (CMDP) [20], [21] learns a policy maximizing the total reward while maintaining the total cost under certain constraints. However, the cost or the constraint does not explicitly represents all the safety requirements of the physical dynamic systems and cannot be directly applied to solve CAV challenges. The recent advance with a formal safety guarantee is the model predictive shielding (MPS) that also works for multi-agent systems [22], [23]. However, their safety guarantee assumes an accurate model of vehicles which is difficult to find in reality. CBFs are used to map unsafe actions to a safe action set in MARL [24], but they do not consider how to design a spatial-temporal-aware network structure for challenging scenarios. In this work, we first integrate the strengths of both constrained MARL and CBF-based *Safety Shield* to further improve the safety of CAVs under threats of traffic hazards.

III. Problem Formulation

A. Problem Description

We consider the cooperative policy-learning problem for CAVs in challenging scenarios occurred on a multi-lane urban intersection or on a multi-lane highway (as shown in Fig.1). Other traffic participants include unconnected vehicle (UCVs) and a hazard vehicle (HAZV). Meanwhile infrastructures that have sensing, communication and computation abilities also play a supportive role to CAVs.

A CAV agent or ego vehicle i is primarily supported with its own observation o_i, the shared observation $o_{\mathcal{N}_i}$ from neighboring agents $\mathcal{N}_i$ based on V2V communication and the shared observation o_{inf} from the road infrastructures. Specifically, ego vehicle's neighbors $\mathcal{N}_i$ provides extra sensor measurements and sensor-detection data, such as lane-detection with camera images and object detection with LiDARs [25]. The CAV neighbors $\mathcal{N}_i$ also share their action histories which are used by the Safety Shield in IV-B to avoid merging conflict. o_{inf} is broadcasted messages to CAVs from road infrastructures, such as Radar that can broadcast the detected speed and location of nearby vehicles.

B. Constrained MARL Problem Formulation

A Constrained MARL is defined as a tuple $G = (\mathcal{S}, \mathcal{A}, P, \{r_i\}, \{c_i\}, \mathcal{G}, \gamma)$ where $\mathcal{G} := (\mathcal{N}, \mathcal{E})$ is the communication network of all CAV agents; $\mathcal{S}$ is the joint state space of all agents: $\mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. The state space of agent i: $\mathcal{S}_i = \{o_i, o_{\mathcal{N}_i}, o_{\text{inf}}\}$ contains information from three sources: self-observation o_i by vehicle i's own odometers and sensors, observation $o_{\mathcal{N}_i}$ shared by other connected agents and observation o_{inf} shared by infrastructure. The observation of CAV i is $o_i = \{(\boldsymbol{l}_i, \boldsymbol{v}_i, \boldsymbol{\alpha}_i), \det_i\}$, where $(\boldsymbol{l}_i, \boldsymbol{v}_i, \boldsymbol{\alpha}_i)$ is the GPS location, velocity and acceleration of agent i, $\det_i$ is the vision-based sensors (on-board camera and 3D point-cloud LiDAR) object detection results. The joint action set is $\mathcal{A} := \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ where $\mathcal{A}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,4+k}\}$ is the discrete finite action space for agent i. KEEP-LANE-SPEED ($a_{i,1}$): CAV i maintains current speed in the current lane. CHANGE-LANE-LEFT ($a_{i,2}$): CAV i changes to its left lane. In experiment, by taking $a_{i,2}$ we set a target waypoint trajectory onto its left neighboring lane [26]. CHANGE-LANE-RIGHT ($a_{i,3}$): CAV i changes to its right lane. In experiment, by taking $a_{i,3}$ we set a target waypoint trajectory onto its right neighboring lane. BRAKE ($a_{i,4}$): in the experiment, CAV i's actuator will compute a brake value within range $brake_i^t \in [0, 0.5]$ at time t. THROTTLE: $a_{i,5}, a_{i,6}, \ldots, a_{i,4+k}$ are k discretized throttle intervals. Given the available throttle value set in the simulator as $[0, 1]$, we set $a_{i,4+j} = [\frac{j-1}{k}, \frac{j}{k}]$. By choosing the action $a_{i,5}$, for example, the actuator of the vehicle i will maintain in current lane and compute a throttle value $throttle_i \in [\frac{j-1}{k}, \frac{j}{k}]$ according to controller's approach.

The state transition function is $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$. The reward function $r := \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. With agent j's velocity defined as $\boldsymbol{v}_j$, agent i's reward function is $r_i(s, a) = \sum_j \mu_{i,j} \|\boldsymbol{v}_j\|_2$, with $\mu_{i,j}$ as non-negative weights. Every agent aims to maximize the weighted sum of all agents'

speed. The cost function $c_i := \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is defined as $c_i(s,a) = \min(\|\boldsymbol{l}_i - \boldsymbol{l}_\kappa\|, \|\boldsymbol{l}'_i - \boldsymbol{l}'_\kappa\| \mid \forall \kappa \in \Gamma_i)$, in which we consider the ego vehicle's distance to its closest neighbors and all the detected environment vehicles Γ_i for the current step location $\boldsymbol{l}_i$ and next step location $\boldsymbol{l}'_i$. The local policy with parameter $\boldsymbol{\theta}_i$ used by agent i is defined as: $\pi^{\boldsymbol{\theta}_i}(a_i|s_i)$. As shown in Fig. 2a, selected actions are examined by the *Safety Shield* discussed in IV-B in order to guarantee the satisfaction of safety constraints and only safe actions will be implemented by lower level controllers.

C. Spatial-Temporal Encoding

Graph Convolutional Network and Transformer have shown advantages in modeling spatial and sequential information [14]. GCN has been utilized [13], [27], [28] to decode interactions between vehicles for collision and trajectory predictions. Transformer for sequential learning also achieves superior performance in some trajectory prediction solutions [15]. As in Fig. 2a, we design a GCN-Transformer module utilizing shared information to encode spatial-temporal features of driving environments for MARL. Ego vehicle's observation o_i, shared observations $o_{\mathcal{N}_i}$ and o_{inf} from V2X communication are used to construct graphs comprised of vehicles, roads and intersections as nodes, and edges among them [13]. Agents are enabled by GCN to characterize the complex communication with other vehicles or road nodes; we also down-sample the significant nodes in large graphs with excessive nodes for scalability. With such graphs in consecutive time steps as input, the Transformer module encodes each agent i's dynamic $(\boldsymbol{l}_i, \boldsymbol{v}_i, \boldsymbol{\alpha}_i)$ with their graph neighbors and generates the spatial-temporal representations of environment as inputs to the MARL model.

IV. Methodology

In this section, we introduce our major contribution, the Constrained Multi-Agent Advantage Actor-Critic (C-MAA2C) framework with a CBF-based *Safety Shield* and a collaborative policy learning scheme. The proposed methodology improves the safety and efficiency of CAV systems and defines the coordination mechanisms in both information-sharing and the policy-learning process. We will introduce the main Algorithm 1 CMAA2C in subsection IV-A, followed by details of the safety checking and training process in subsections IV-B and IV-C.

In particular, we adopt the GCN-Transformer neural networks to approximate the Q-function, cost function [20] and policy for each agent. We use advantage to optimize the policy under the constraint of the cost. During training, agents also exchange the policy network parameters with their neighboring agents [21] to approximate a global optimal solution with distributed policy optimization.

We design a *Safety Shield* based on control barrier function and quadratic programming (CBF-QP) for each agent to check the safety of the action selected by the reinforcement learning process and make corrections for the control input. We design a barrier function that considers the acceleration in both front and rear vehicles for changing lane maneuver [26]. We utilize shared action histories by neighboring agents to build barrier functions for the rear vehicles in the CBF-QP, to avoid merging conflict to the same lane by multiple CAVs.

A. Constrained Multi-Agent Advantage Actor-Critic

In Algorithm 1, we use centralized training decentralized execution design. Each agent maintains a policy network $\pi^{\boldsymbol{\theta}_i}$ ("actor") with parameter $\boldsymbol{\theta}_i$, a $Q(s,a)$ network with parameter $\boldsymbol{\phi}_i$ ("critic") for the reward $r_i(s,a)$ and another $Q^C(s,a)$ network with parameter $\boldsymbol{\omega}_i$ ("cost") for the cost $c_i(s,a)$ (as in Fig. 2a). $\boldsymbol{\theta}$ is defined as the parameter of the joint policy taken by all agents. The algorithm operates in forward view as agents interact within the environment. After observing the state s_i, the agent's stochastic policy computes for the probability over action set $\mathcal{P}(\mathcal{A}_i)$. Meantime, the safety checking based on s_i generates the safe action set $\mathcal{A}_i^{\text{safe}}$ including all the safe candidate actions $a_{i,\kappa}$. The eventual behavior will be sampled from $\mathcal{P}(\mathcal{A}_i^{\text{safe}})$ based on ϵ-greedy. After all agents instruct their selected behavior a_i to controller and have them executed, the algorithm synchronously goes to the next step by observing the reward $r_i(s,a)$, cost $c_i(s,a)$ and the new state s'_i. Specifically, all the CAVs want to collaboratively optimize the total expected return of the system defined as $J^R(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i\in\mathcal{N}} J_i^R(\boldsymbol{\theta})$:

$$J^R(\boldsymbol{\theta}) = \sum_{i\in\mathcal{N}_i} \mathbb{E}_{a^k\sim\pi^{\boldsymbol{\theta}}(\cdot|s^k)}[\sum_{k=0}^{\infty}(\gamma_r)^k r_i(s^k,a^k)] \quad (1)$$

where $\gamma = (\gamma_r, \gamma_c)$ are the discount factors for reward and cost respectively. Maximizing objective (1) is equivalent to minimizing the negative of such value, subject to the cost's lower-bound constraint ζ_i that each agent i's expected accumulated cost $J_i^C(\boldsymbol{\theta}) = \mathbb{E}_{a^k\sim\pi^{\boldsymbol{\theta}}(\cdot|s^k)}[\sum_{k=0}^{\infty}(\gamma_c)^k c_i(s^k,a^k)]$ should satisfy. Thus the constrained MARL problem is defined as the following optimization problem

$$\min_{\boldsymbol{\theta}} -J^R(\boldsymbol{\theta}) \quad s.t. \quad J_i^C(\boldsymbol{\theta}) \geq \zeta_i, \forall i \in \mathcal{N}; \Theta_i = \Theta j, j \in \mathcal{N}_i \quad (2)$$

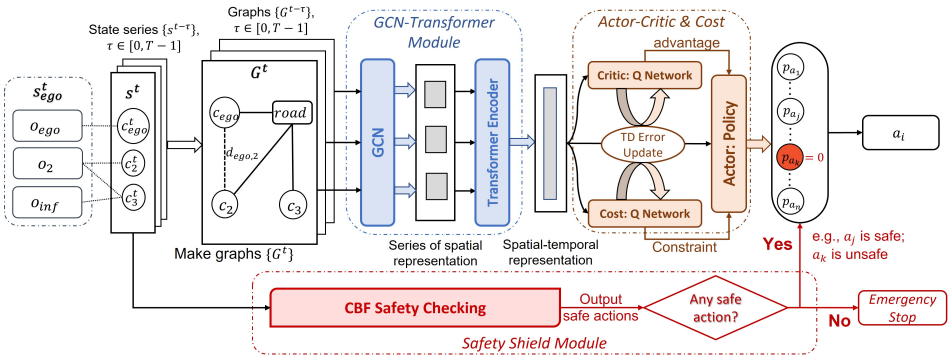
where $\Theta_i := \boldsymbol{\theta}_i \times \boldsymbol{\theta}_{-i}$ is defined as the local copy of the policy $\boldsymbol{\theta}$ owned by agent i according to [21]. By the Lagrangian method [29], the problem (2) can be written as the following problem solved through the training process:

$$\begin{aligned} \min_{\boldsymbol{\theta}_i\in\boldsymbol{\theta}} \max_{\boldsymbol{\lambda}\geq 0} \quad & \mathcal{L}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}, \boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{-i}) \\ s.t. \quad & \Theta_i = \Theta j, \quad \forall j \in \mathcal{N}_i, \quad \forall i \end{aligned} \quad (3)$$

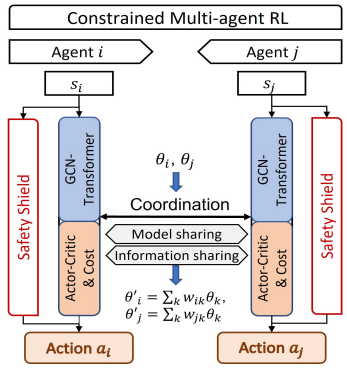
where $\boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{-i}$ denote the dual variables, and $\mathcal{L}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{-i}, \boldsymbol{\lambda}_i, \boldsymbol{\lambda}_{-i}) \triangleq \frac{1}{n}\sum_{i\in\mathcal{N}}[J_i^R(\boldsymbol{\theta}) + \langle \zeta_i - J_i^C(\boldsymbol{\theta}), \boldsymbol{\lambda}_i\rangle]$.

B. Safety Shield and Safety Checking

To enhance the safety of agents during their interactions, we design a *Safety Shield* module to identify potential unsafe actions that violate the safety requirements and update the safe action set for the constrained MARL in Algorithm 1. Given agent i's state s_i, safety checking will loop through all candidate actions $a_{i,\kappa} \in \mathcal{A}_i$ and judge if $a_{i,\kappa}$ is safe based on control barrier functions and quadratic programming (CBF-QP). CBFs have been introduced to ensure set invariance



(a) Model pipeline for a single agent. The state information as time series $\{s^{t-\tau}\}_\tau$ will be processed as graphs first and sequentially enter the GCN-Transformer module and the Actor's policy network; meanwhile, s^t is input to the CBF safety checking module for computing safe actions. During training, the outputs of GCN-Tranformer will be input to the Critic and Cost network for advantage, constraint and TD error calculation.



(b) Constrained MARL and *Safety Shield* framework, with coordinated information-sharing and policy-learning among agents.

Fig. 2. Single agent's model pipeline in 2a; Constrained MARL framework in 2b

Algorithm 1: Constrained Multi-Agent A2C

Initialize replay memory $M = \bigcup_i M_i$; *Initialize* actor, critic and cost networks $\boldsymbol{\theta}_i^0, \boldsymbol{\phi}_i^0, \boldsymbol{\omega}_i^0$; *Initialize* $\boldsymbol{\vartheta}_i^0 = \mathbf{0}, \boldsymbol{\lambda}_i^0 = 0$;
for *each episode* ϵ **do**
  Initialize $s = \prod_i s_i \in \mathcal{S}$;
  Initialize safe action set $\mathcal{A}^{\text{safe}} = \prod_i \mathcal{A}_i^{\text{safe}} = \mathcal{A}$;
  for *each training cycle* τ **do**
    for *each step* **do**
      Choose $a_i \in \mathcal{A}_i^{\text{safe}}$ based on ϵ-greedy, $a = \prod a_i$;
      Execute action a, observe rewards $r = \{r_i\}$, costs $c = \{c_i\}$, and the new state $s' = \prod_i s_i'$;
      Store $(s_i, a_i, r_i, c_i, s_i'), \forall i$ in M_i;
      if *collision* **then** continue;
      Update $\mathcal{A}^{\text{safe}} =$ **Safety_Checking**(s');
      $s \leftarrow s'$;
    end
    Perform **Training**$(\tau, M, \boldsymbol{\theta}_i^\tau, \boldsymbol{\phi}_i^\tau, \boldsymbol{\omega}_i^\tau, \boldsymbol{\vartheta}_i^\tau, \boldsymbol{\lambda}_i^\tau)$;
  end
end

Algorithm 2: Safety_Checking

Input: $s = \prod s_i$; initialize $\mathcal{A}^{\text{safe}} = \emptyset$;
for *each agent* i **do**
  for *each action* $a_{i,\kappa} \in \mathcal{A}_i$ **do**
    if $a_{i,\kappa}$ *is safe, i.e. CBF-QP has a feasible solution* **then** append $a_{i,\kappa}$ to $\mathcal{A}_i^{\text{safe}}$;
  end
  if $\mathcal{A}_i^{safe} = \emptyset$ **then** $\mathcal{A}_i^{\text{safe}} = [Emergency\_stop]$;
end

with system dynamics knowledge [30], [31] and ensure safe controller design of vehicles [26], [32], [33], [24].

Consider a nonlinear affine control system: $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$ with state $\boldsymbol{x} \in \mathbb{R}^n$, input $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^m$, $\mathcal{U}$ is the admissible input set of the system, f and g are locally Lipschitz. Define a superlevel set $\mathcal{C} \subset \mathbb{R}^n$ of a differentiable function h: $\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}, t) \geq 0\}$. A set $\mathcal{C} \subset \mathbb{R}^n$ is *forward invariant* if for every $\boldsymbol{x}_0 \in \mathcal{C}$, the solution $\boldsymbol{x}(t)$ to the system satisfies $\boldsymbol{x}(t) \in \mathcal{C}$ for all $t \geq 0$. The system is *safe* with respect to the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant [30]. The function h is a control barrier function (CBF) for the system on $\mathcal{C}$ if there exists $\gamma \in \mathcal{K}_\infty$ [34]:

$$\sup_{\boldsymbol{u} \in \mathcal{U}} [\frac{\partial h(\boldsymbol{x}, t)}{\partial t} + L_f h(\boldsymbol{x}, t) + L_g h(\boldsymbol{x}, t)\boldsymbol{u}] \geq -\gamma h(\boldsymbol{x}, \boldsymbol{t})$$

CBF evaluating the safety of an action $a_{i,\kappa}$ focuses on the relevant vehicles given $a_{i,\kappa}$ will be executed. As is shown in Fig. 3, if a change-lane action is evaluated, target vehicles are the nearest neighbors from the front vehicles, front and rear vehicles on the target lane, and front and rear CAVs on the left/right other lane if such vehicle is also changing to the lane that ego vehicle is targeting. Otherwise, only front and rear neighbors in the current lane are concerned.

We adopt the kinematic bicycle model for its simplicity while considering the non-holonomic vehicle behaviors [35]. The state of the system $\boldsymbol{x} = [x, y, v, \psi]^T$ are the coordinates, velocity, orientation of the vehicle's center of gravity (c.g.) in an inertial frame (X,Y). The inputs $\boldsymbol{u}$ to the system are acceleration at the vehicle's c.g. α and the steering angle of the vehicle φ.

We consider the function of *safety following distance* $\mathcal{D}_f(v, v_f) := c_1 v + c_2(\frac{v^2}{2|\max(\alpha)|} - \frac{v_f^2}{2|\max(\alpha_f)|}) + D$, if the target vehicle is in the front on any lane, as in Fig. 3. It takes the front and rear vehicles' velocity as input, and considers both the reaction delay term $c_1 v$ and the hard-braking term $c_2(\frac{v^2}{2|\max(\alpha)|} - \frac{v_f^2}{2|\max(\alpha_f)|})$ which is proportional to the difference of hard-braking distances between the front and following vehicles, with an extra buffer distance as constant D. If target vehicle is behind, *safety leading distance* is defined as $\mathcal{D}_l(v, v_b) := c_1 v_b + c_2(\frac{v_b^2}{2|\max(\alpha_b)|} - \frac{v^2}{2|\max(\alpha)|}) + D$. Barrier function $h(\boldsymbol{x}, t)$ can then be respectively given as $h_f(\boldsymbol{x}, t) := (x_f - x) - \mathcal{D}_f(v, v_f)$ and $h_b(\boldsymbol{x}, t) := (x - x_b) - \mathcal{D}_l(v, v_b)$. For each candidate action $a_{i,\kappa}$, $\boldsymbol{u}_i$ is the corresponding control input generated by a nominal controller, e.g. PID controller. Then the safe candidate action can be evaluated by solving the below quadratic program *CBF-QP*. If *CBF-QP* is solvable, $a_{i,\kappa}$ is safe; otherwise it is unsafe. As in Algorithm 2, an emergency stop will be taken if no candidate action is safe.

$$\textit{CBF-QP:} \quad \min_{\boldsymbol{u} \in \mathbb{R}^m} \frac{1}{2} \parallel \boldsymbol{u} - \boldsymbol{u}_i \parallel^2 \tag{4}$$
$$\text{s.t.} \quad \frac{\partial h(\boldsymbol{x}, t)}{\partial t} + L_f h(\boldsymbol{x}, t) + L_g h(\boldsymbol{x}, t)\boldsymbol{u} \geq -\gamma h(\boldsymbol{x}, t)$$

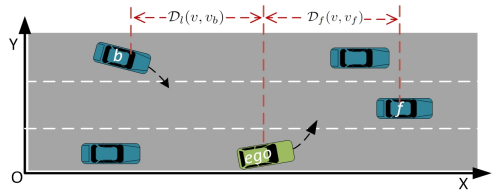


Fig. 3. Safety checking for ego vehicle's lane-change. We consider both the vehicle f in the target lane and the vehicle b entering the target lane.

Algorithm 3: Training

Input: $\tau, M, \boldsymbol{\theta}_i^\tau, \boldsymbol{\phi}_i^\tau, \boldsymbol{\omega}_i^\tau, \boldsymbol{\vartheta}_i^\tau, \boldsymbol{\lambda}_i^\tau$;
for *each agent* i **do**
  Update the $\boldsymbol{\theta}_i^{\tau+1}$ with (7);
  Sample a batch B_i^τ from M, Update $\boldsymbol{\phi}_i, \boldsymbol{\omega}_i$ with (9) respectively;
  Calculate $\widehat{\nabla}_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^\tau, \boldsymbol{\lambda}_i^\tau)$, Update the $\boldsymbol{\vartheta}_i^{\tau+1}$ by (8);
  Calculate $(\hat{J}_i^C)(\boldsymbol{\theta}_i^{\tau+1})$, Update the $\boldsymbol{\lambda}_i^{\tau+1}$ by (10);
end

C. Training

As introduced in (IV-A), the algorithm operates in forward view and updates the model parameters in every training cycle. During the training, the algorithm loops through agents and sequentially updates their policy parameters $\boldsymbol{\theta}_i$, the critic and cost network parameters $\boldsymbol{\phi}_i, \boldsymbol{\omega}_i$, the auxiliary policy gradient variables $\boldsymbol{\vartheta}_i$ and the dual variable $\boldsymbol{\lambda}_i$ with the training batch B_i sampled from memory. Steps are given in algorithm 3.

Let $F_i(\boldsymbol{\theta}, \boldsymbol{\lambda}_i) \triangleq J_i^R(\boldsymbol{\theta}) + \langle \zeta_i - J_i^C(\boldsymbol{\theta}), \boldsymbol{\lambda}_i \rangle$,$\forall i$. The estimated policy gradients regarding primal variables are

$$\widehat{\nabla}_{\boldsymbol{\theta}_i} F_i(\boldsymbol{\theta}, \boldsymbol{\lambda}_i) = \widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^R(\boldsymbol{\theta}) - \langle \widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^C(\boldsymbol{\theta}), \boldsymbol{\lambda}_i \rangle, \forall i \tag{5}$$

and the policy gradients with respect to dual variables are

$$\widehat{\nabla}_{\boldsymbol{\lambda}_i} F_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i) = \zeta_i - \hat{J}_i^C(\boldsymbol{\theta}_i), \forall i \tag{6}$$

The update of actor's policy network follows the approach from safe-Dec policy gradient algorithm [21], in which every agent maintains a local policy with parameters $\boldsymbol{\theta}_i$ and a copy of auxiliary policy gradients $\boldsymbol{\vartheta}_i$ computed based on local and neighbors' gradients. $\boldsymbol{\vartheta}_i$ is initialized as $\boldsymbol{\vartheta}_i^0 = \mathbf{0}$. For each training round τ, the new local policy $\boldsymbol{\theta}_i^{\tau+1}$ is updated with the current local policy $\boldsymbol{\theta}_i^\tau$, the local copy of policy gradients $\boldsymbol{\vartheta}_i$ and policies shared by its neighbors $\{\boldsymbol{\theta}_j^\tau\}_{j \in \mathcal{N}_i}$ as in (7). The update of policy gradient follows (8). In (7), (8), σ^τ is the step size; $\mathbf{W}$ is weight matrix characterizing relations among nodes in $\mathcal{G}$ introduced in [21].

$$\boldsymbol{\theta}_i^{\tau+1} = \textstyle\sum_{j \in \mathcal{N}_i} \mathbf{W}_{i,j} \boldsymbol{\theta}_j^\tau - \sigma^\tau \boldsymbol{\vartheta}_i^\tau \tag{7}$$

$$\boldsymbol{\vartheta}_i^{\tau+1} = \textstyle\sum_{j \in \mathcal{N}_i} \mathbf{W}_{i,j} \boldsymbol{\vartheta}_j^\tau + \widehat{\nabla}_{\boldsymbol{\theta}_i} F_i(\boldsymbol{\theta}_i^{\tau+1}, \boldsymbol{\lambda}_i^\tau) - \widehat{\nabla}_{\boldsymbol{\theta}_i} F_i(\boldsymbol{\theta}_i^\tau, \boldsymbol{\lambda}_i^\tau), \forall i \tag{8}$$

We use the temporal difference error defined in (9) for critic and cost respectively, to compute the loss for two networks. Specifically, $R_i - V_i(s)$, $R_i^C - V_i^C(s)$ are advantages [36] of the return and cost to compute gradients of policy network in $\widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^R(\boldsymbol{\theta})$ and $\widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^C(\boldsymbol{\theta})$ respectively in (5).

$$L_{i,critic} = (R_i^t - V_i(s^t))^2, L_{i,cost} = (R_i^{C,t} - V_i^C(s^t))^2. \tag{9}$$

The update of dual variable $\boldsymbol{\lambda}$ follows the approach in [21] as (10), where $\mathcal{P}_\Lambda$ is the projection operator mapping $\boldsymbol{\lambda}_i$ to a non-negative value and $\Lambda = \{\boldsymbol{\lambda}_i | \boldsymbol{\lambda}_i \geq 0\}, \forall i$ stands for the feasible set of $\boldsymbol{\lambda}_i$; ρ is the stepsize.

$$\boldsymbol{\lambda}_i^{\tau+1} = \mathcal{P}_\Lambda((1 - \rho\gamma^\tau)\boldsymbol{\lambda}_i^\tau + \rho \widehat{\nabla}_{\boldsymbol{\lambda}_i} F_i(\boldsymbol{\theta}_i^{\tau+1}, \boldsymbol{\lambda}_i^\tau)) \tag{10}$$

V. Experiments and Evaluations

We deploy our experiment in the CARLA Simulator environment [37], where each vehicle is configured with inborn GPS and IMU sensors and a collision sensor that detects the collision with other objects. We set the communication range of all CAVs in simulation as $100m$. The k-discretized throttle ranges in action space $\mathcal{A}_i$ is set as $k = 3$. We set the training cycle as 16 steps, the discount factors as $\gamma_r = 0.99, \gamma_c = 0.9$, and each model was trained 200 episodes in two scenarios. The constraints ζ_i for agents are 10; the weight matrix $\mathbf{W}$ in training generally balances the weights between ego and others' policies and takes different values based on the number of agents. The training and testing of our algorithm and baselines took place in a server configured with AMD Ryzen 3970X 32-Core processor and four NVIDIA Quadro RTX 6000 GPUs. The experiments are performed with CARLA 0.9.11, Python 3.7, PyTorch 1.10, and CUDA 11.4.

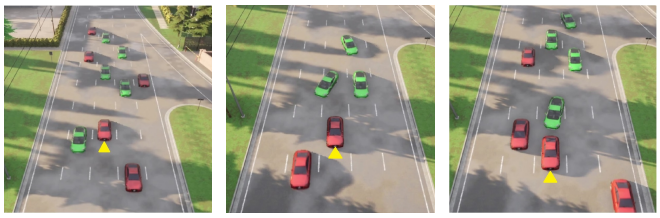
A. Simulation with Challenging Scenario

We aim to deal with challenging scenarios in real life. Specifically, safety-critical events such as ***running a red light*** at an intersection, and ***hard-braking*** in highway traffic incurred by another vehicle are usually immediate life threats to drivers and passengers. In the experiment, apart from the connected autonomous vehicles (*CAVs*) and unconnected vehicles (*UCVs*), we explicitly define a hazard vehicle (*HAZV*) taking the aforementioned dangerous behaviors in 3 respective scenarios as illustrated in Fig. 1 and 4.

1) Intersection: The first row of Fig. 1 are challenging intersection scenarios where three CAVs (green) are driving through the intersection and the HAZV (red) from the crossing direction recklessly passes the intersection at the same time. The throttle values taken by the HAZV in simulator are randomly sampled from $[0.65, 0.85]$ plus a tiny step-wise perturbation for continuous acceleration. In Fig. 1 we present samples of initialization, success and failure cases of collision avoidance in the experiment.

2) Highway: Figures in the second row of Fig. 1 illustrate the challenging highway scenario, in which three CAVs, a UCV (red) and a HAZV (yellow mark) are spawned to ride on a multi-lane highway. The HAZV suddenly hard-brakes, taking the step-wise brake values in simulator randomly sampled from $[0.9, 1.0]$ and causing an immediate threat to its rear CAVs. Meanwhile the UCV stays in its lane and takes throttles in $[0.3, 0.7]$ securing its smooth driving. Success and failure of collision-avoidance cases are given in Fig. 1.

3) Highway-Hard: For testing, we also devised a more difficult *Highway-Hard* scenario shown in Fig. 4. Ten vehicles including 5 CAVs, 4 UCVs (red) and 1 HAZV (yellow mark) are spawned in a compact traffic. The HAZV and UCVs behave similarly as in *Highway*. The *Highway-Hard* is comprehensively more challenging as it contains more agents and UCVs, and the compact vehicles' configuration produces complex interactions.



(a) (b) (c)

Fig. 4. *Highway-Hard* scenario for testing, with 10 vehicles. 4a: *Highway-Hard* initialization; 4b: collision-free case with our method where agents collaboratively change to different neighboring lanes to avoid the hard-braking HAZV; 4c: CAV agent in baseline collides with HAZV.

TABLE I

TRAINING RESULTS IN TWO SCENARIO

Scenario	Baselines		Ours
	w/o *SS*[1]	FC-CA2C[2]	GT-CA2C[3]
Intersection	21%; 430.8	93%; 572.8	**96%; 624.8**
Highway	0%; 166.4	91%; 920.1	**95%; 955.6**

[1]w/o *SS*: Ours GT-CA2C without *Safety Shield*; [2]FC-CA2C: Fully-Connected Constrained Advantage Actor-Critic; [3]GT-CA2C: GCN-Transformer Constrained Advantage Actor-Critic

Each entry above is (*collision-free rate*; *mean episode return*). Our method achieves **highest** safety and efficiency in the training phase.

TABLE II

TESTING RESULTS IN THREE SCENARIOS.

Scenario	Baselines		Ours
	w/o *SS*	FC-CA2C	GT-CA2C
Intersection	20%; 444.8	86%; 579.6	**94%; 586.8**
Highway	2%; 185.3	**90%**; 922.6	**90%; 926.7**
Highway-Hard	0%; 108.6	70%; 706.4	**78%; 724.3**
Intersection w/o Communication	20%; 432.3	44%; 473.7	44%; 513.9
Highway-Hard w/o Communication	0%; 110.8	46%; 567.5	48%; 565.6

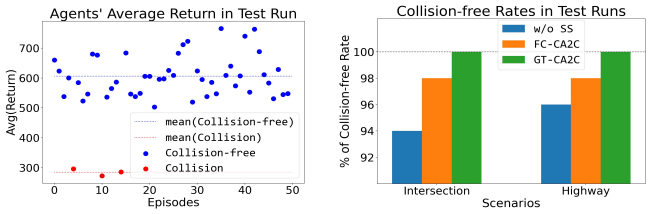
Each entry above is (*collision-free rate*; *mean episode return*). Our method **outperforms** baselines in two metrics, proving the improved safety and efficiency with GCN-Transformer and *Safety Shield*.

B. Experiment Results

We trained our model (GCN-Transformer Constrained Advantage Actor-Critic; 'GT-CA2C' in the table I, II), a baseline using our model without *Safety Shield* ('w/o *SS*' in tables) and another baseline 'FC-CA2C' with fully-connected layers (replacing GCN-Transformer), constrained advantage actor-critic and *Safety Shield*, each on *Intersection* and *Highway* scenarios. Our method and baselines are all under the multi-agent framework in Alg. 1. Training and testing experiment results are presented in table I and II. We **highlight** our method's top leading performance among all solutions. For each entry in tables, the left percentage is the collision-free rate in simulation; the right number is the mean episode return defined as the mean of agents' sums over stepwise rewards in every episode: $\sum_{\epsilon=1}^{m} \text{Avg}_i(\sum_t r_i^t)/m$. Examples of episode return values from testing our model in *Intersection* are given in the scatter plot in Fig. 5a.

1) Effectiveness of Safety Shield: In all scenarios, our approach outperforms baselines in collision-free rate and overall return. Compared with the baseline 'w/o *SS*', the huge gaps in both metrics demonstrate improved safety and efficiency with our CBF-based safety checking method.

2) GCN-Transformer and Improved Environment Awareness: With the GCN-Transformer module applied compared



(a) (b)

Fig. 5. 5a: Scatter plots of episode returns in *Intersection*; collision could affect agents' return greatly. 5b: Collision-free rates in normal driving scenarios without hazard; our method achieves 100% safety in both scenarios and leads all solutions.

to 'FC-CA2C', our method has leading performance in collision-free rates and mean episode returns in all three testing scenarios. In *Highway-Hard* particularly, we find the advantage of our method is enlarged compared with the easier *Highway*, and this verifies the significance of enhanced environment awareness with our approach under the more challenging and hazardous scenarios.

3) Benefits of Coordination under Challenging Scenarios: To verify the benefits of the coordination mechanisms, we test our model against the absence of V2X communication and observe the cascading performance without it in all solutions. In *Intersection* scenario, the HAZV information becomes unavailable until it appears in CAVs' vision. In *Highway-Hard* scenario, an ego vehicle is unaware of another CAV's intention to change lanes. From table II we could see, although our method surpasses the baselines in both metrics, the performance cannot match the excellence in test runs with coordinated communications. The collision-free rate drops from 94% to 44% in *Intersection* scenario, and from 78% to 48% in *Highway-Hard*, and this also applies to the baseline 'FC-CA2C'. The above results could prove the major contribution of coordination through information-sharing based on V2X communication.

4) Performance in Normal Driving Scenario: Lastly, we show results from testing in the remake hazard-free scenarios *Intersection-Normal* and *Highway-Normal* in Fig. 5b, in which the HAZV doesn't break into the intersection or brake abruptly. Our method can still perform well in the normal driving scenario as it achieved 100% collision-free rate, while Baselines 'w/o *SS*' and 'FC-CA2C' both have collision without hazard.

VI. CONCLUSION

In this work, we study the connected autonomous vehicles' cooperative policy-learning problem in challenging driving scenarios. We propose a constrained MARL coordinated policy learning framework with a *Safety Shield* for CAVs based on information-sharing. The GCN-Tranformer encoder is introduced to MARL to raise agents' spatial-temporal awareness of the environment. In experiments, we verify the effectiveness and advantage of our method and each of its modules in both safety and efficiency by comparing results with baseline models or settings, in challenging driving scenarios with hazard vehicles in traffic. Future work could extend to enhance the robustness of MARL algorithm and CBF *Safety Shield* with noisy and erroneous shared observations or models.

REFERENCES

[1] D. Martín-Sacristán, S. Roger, D. Garcia-Roger, J. F. Monserrat, P. Spapis, C. Zhou, and A. Kaloxylos, "Low-latency infrastructure-based cellular v2v communications for multi-operator environments with regional split," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1052–1067, 2020.

[2] H. Mun, M. Seo, and D. H. Lee, "Secure privacy-preserving v2v communication in 5g-v2x supporting network slicing," *IEEE Trans. Intell. Transp. Syst.*, 2021.

[3] N. Buckman, A. Pierson, S. Karaman, and D. Rus, "Generating visibility-aware trajectories for cooperative and proactive motion planning," in *ICRA*. IEEE, 2020, pp. 3220–3226.

[4] A. Miller and K. Rim, "Cooperative perception and localization for cooperative driving," in *ICRA 2020*. IEEE, 2020, pp. 1256–1262.

[5] S. Han, H. Wang, S. Su, Y. Shi, and F. Miao, "Stable and efficient shapley value-based reward reallocation for multi-agent reinforcement learning of autonomous vehicles," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8765–8771.

[6] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.

[7] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 81–90, March 2012.

[8] M. Bojarski and D. Testa, "End to end learning for self-driving cars," *arXiv:1604.07316*, 2016.

[9] R. Cheng and G. Orosz, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *AAAI*, vol. 33, 2019, pp. 3387–3395.

[10] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," *arXiv:1704.03952*, 2017.

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

[12] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *ICCV*, 2015, pp. 2722–2730.

[13] A. V. Malawade, S.-Y. Yu, B. Hsu, D. Muthirayan, P. P. Khargonekar, and M. A. Al Faruque, "Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9379–9388, 2022.

[14] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *ICML*, 2019, pp. 2961–2970.

[15] J. Zhao, X. Li, Q. Xue, and W. Zhang, "Spatial-channel transformer network for trajectory prediction on the traffic scenes," *arXiv:2101.11472*, 2021.

[16] K. Zhang and Z. Yang, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *arXiv:1911.10635*, 2019.

[17] J. Foerster and G. Farquhar, "Counterfactual multi-agent policy gradients," in *AAAI*, 2018.

[18] T. Rashid and M. Samvelyan, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *ICML*, 2018, pp. 4295–4304.

[19] R. Lowe and Y. I. Wu, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *NeurIPS*, 2017, pp. 6379–6390.

[20] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng, "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–7.

[21] S. Lu, K. Zhang, T. Chen, T. Başar, and L. Horesh, "Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, 2021, pp. 8767–8775.

[22] S. Li and O. Bastani, "Robust model predictive shielding for safe reinforcement learning with stochastic dynamics," in *ICRA*, 2020, pp. 7166–7172.

[23] W. Zhang, O. Bastani, and V. Kumar, "Mamps: Safe multi-agent reinforcement learning via model predictive shielding," *arXiv:1910.12639*, 2019.

[24] S. Han, S. Zhou, J. Wang, L. Pepin, C. Ding, J. Fu, and F. Miao, "A multi-agent reinforcement learning approach for safe and efficient behavior planning of connected autonomous vehicles," *arXiv:2003.04371*, 2022.

[25] Y. Li, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: A virtual collaborative perception dataset for autonomous driving," *arXiv preprint arXiv:2202.08449*, 2022.

[26] S. He, J. Zeng, B. Zhang, and K. Sreenath, "Rule-based safety-critical control design using control barrier functions with application to autonomous lane change," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 178–185.

[27] J. Schmidt, J. Jordan, F. Gritschneder, and K. Dietmayer, "Crat-pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention," *arXiv preprint arXiv:2202.04488*, 2022.

[28] Y. Ban, X. Li, G. Rosman, I. Gilitschenski, O. Meireles, S. Karaman, and D. Rus, "A deep concept graph network for interaction-aware trajectory prediction," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8992–8998.

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

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

[31] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3420–3431.

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

[33] G. Notomista, M. Wang, M. Schwager, and M. Egerstedt, "Enhancing game-theoretic autonomous car racing using control barrier functions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5393–5399.

[34] G. Wu and K. Sreenath, "Safety-critical control of a 3d quadrotor with range-limited sensing," in *Dynamic Systems and Control Conference*, vol. 50695. American Society of Mechanical Engineers, 2016, p. V001T05A006.

[35] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Autonomous driving using model predictive control and a kinematic bicycle vehicle model," in *Intelligent Vehicles Symposium, Seoul, Korea*, 2015.

[36] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1928–1937.

[37] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.