A Multi-Agent Reinforcement Learning Approach for Safe and Efficient Behavior Planning of Connected Autonomous Vehicles

Songyang Han, *Student Member, IEEE*, Shanglin Zhou, Jiangwei Wang, *Student Member, IEEE*, Lynn Pepin, Caiwen Ding, *Member, IEEE*, Jie Fu, *Member, IEEE*, and Fei Miao, *Member, IEEE*

***Abstract*— The recent advancements in wireless technology enable connected autonomous vehicles (CAVs) to gather information about their environment by vehicle-to-vehicle (V2V) communication. In this work, we design an information-sharing-based multi-agent reinforcement learning (MARL) framework for CAVs, to take advantage of the extra information when making decisions to improve traffic efficiency and safety. The safe actor-critic algorithm we propose has two new techniques: the truncated $\mathcal{Q}$-function and safe action mapping. The truncated $\mathcal{Q}$-function utilizes the shared information from neighboring CAVs such that the joint state and action spaces of the $\mathcal{Q}$-function do not grow in our algorithm for a large-scale CAV system. We prove the bound of the approximation error between the truncated-$\mathcal{Q}$ and global Q-functions. The safe action mapping provides a provable safety guarantee for both the training and execution based on control barrier functions. Using the CARLA simulator for experiments, we show that our approach improves the CAV system's efficiency in terms of average velocity and comfort under different CAV ratios and different traffic densities. We also show that our approach avoids the execution of unsafe actions and always maintains a safe distance from other vehicles. We construct an obstacle-at-corner scenario to show that the shared vision can help CAVs to observe obstacles earlier and take action to avoid traffic jams. The experiment video is on https://songyanghan.github.io/cavmarl/.**

***Index Terms*— Autonomous vehicle, multi-agent reinforcement learning, convolutional neural network, control barrier function.**

I. Introduction

WIRELESS communication technologies such as WiFi and 5G cellular networks enable vehicle-to-vehicle (V2V) communication [1], [2]. The U.S. Department of Transportation (DOT) estimated that V2V communication could address up to 82% of crashes in the U.S. every year [3], [4]. Sharing basic safety messages (BSMs) benefits the coordination of connected autonomous vehicles (CAVs) at intersections and lane-merging scenarios [5], [6], [7], [8]. However, when CAVs get extra environment knowledge via V2V communication, how to make prudent decisions to improve traffic efficiency, and whether communication can bring benefits are still unsolved challenges.

In this work, we consider utilizing V2V communication among CAVs to make better behavior planning and control decisions, such as lane-changing and lane-keeping, while meeting safety requirements and improving traffic efficiency. This problem is not well-studied. Most existing CAV frameworks assume that the lane-changing or keeping decision has already been made, such as platooning [4], adaptive cruise control (ACC) [9], and cooperative adaptive cruise control (CACC) [10]. Reinforcement learning (RL) has shown advantages in decision-making for a single autonomous vehicle in complex driving environments [8], [11], [12], [13], [14], [15], and learning-based method for traffic flow management of autonomous vehicle systems [16], [17]. However, the RL-based decision-making method for multiple CAVs' considering both the system-level objective and individual CAV's safety requirements has not been thoroughly researched [13]. To address this challenge, we design a multi-agent reinforcement learning (MARL) approach. Existing MARL algorithms usually require a centralized critic with access to the global state and the global action [18], which poses three significant challenges for real-world implementation:

- It is difficult for each CAV to get the global state and the global action considering the communication overheads.
- The joint state and action spaces of the critic (Q-function) grow combinatorially with the total number of CAVs. The computational overheads become burdensome when the total number of CAVs becomes very large.
- The action used for both training and execution may be unsafe for the safety-critical CAV system [19].

To tackle these challenges, we design a new algorithm, called the safe actor-critic algorithm, with two new techniques: truncated $\mathcal{Q}$-function and safe action mapping.

To the best of our knowledge, this is the first attempt to design a safe MARL algorithm to solve the behavior planning challenges for CAVs considering the dynamic control process. We design a truncated $\mathcal{Q}$-function with neighboring vehicles' states and actions to approximate the global action

Manuscript received 3 September 2022; revised 4 June 2023; accepted 30 October 2023. This work was supported by NSF under Grant 1849246, Grant 1952096, Grant 2047354, and Grant 2144113. The Associate Editor for this article was A. Bucchiarone. *(Corresponding author: Songyang Han.)*

Songyang Han is with the Department of Computer Science and Engineering, University of Connecticut, Storrs, Mansfield, CT 06268 USA, and also with Sony AI, New York, NY 10010 USA (e-mail: songyang.han@uconn.edu).

Shanglin Zhou, Lynn Pepin, Caiwen Ding, and Fei Miao are with the Department of Computer Science and Engineering, University of Connecticut, Storrs, Mansfield, CT 06268 USA (e-mail: shanglin.zhou@uconn.edu; lynn.pepin@uconn.edu; caiwen.ding@uconn.edu; fei.miao@uconn.edu).

Jiangwei Wang is with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, Mansfield, CT 06268 USA (e-mail: jiangwei.wang@uconn.edu).

Jie Fu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32605 USA (e-mail: fujie@ufl.edu).

Digital Object Identifier 10.1109/TITS.2023.3336670

value function with global states and actions in the MARL. We design a safe action mapping algorithm based on control barrier functions (CBFs) to make sure that the actions trained and executed by our proposed MARL algorithm are safe. In experiments, we show that our approach increases traffic efficiency and guarantees safety.

In summary, the main contributions of this work are:

- We propose a novel safe and efficient actor-critic algorithm for behavior planning of CAVs based on two new techniques: 1) *Truncated $\mathcal{Q}$-function* (for the first two challenges above): Each vehicle learns a truncated $\mathcal{Q}$-function as a critic that only needs the states and actions from neighboring vehicles. The joint state and action spaces of the truncated $\mathcal{Q}$-function do not grow in a large-scale CAV system. 2) *Safe action mapping* (for the third challenge above): We map any action in the action space to the safe action set so that the training and execution have provable safety guarantees.
- To support the learning process of the truncated $\mathcal{Q}$-function, we propose a weight-pruned convolutional neural network (CNN) technique to process the images from the camera and point clouds from LIDAR fast enough such that the vision information is always available for the learning of the truncated $\mathcal{Q}$-function.
- We validate our algorithms in the CARLA simulator [20] that can simulate complicated mixed traffic environments including both autonomous and human-driven vehicles. The experiments show that the safe actor-critic algorithm can improve traffic efficiency with safety guarantees. We also validate our MARL algorithm in challenging driving scenarios like obstacle-at-corner, and the shared vision with our algorithm helps vehicles to avoid traffic jams.

The rest of this paper is organized as follows. We introduce the related work in Section II. In Section III, we introduce the V2V communication setting, formulate the behavior planning and control problem, and give an overview of our novel solutions. In Section IV we propose the truncated $\mathcal{Q}$-function with provable approximation error. In Section V, we design a safe action mapping technique and propose a safe actor-critic algorithm for behavior planning. In Section VI, we show the experimental results with safety and efficiency improvements. Section VII concludes the work.

II. Related Work

A. Deep Learning Framework For Autonomous Driving

Autonomous vehicles can learn the steering angle and acceleration directly based on vision perception, such as end-to-end imitation learning [21], and end-to-end reinforcement learning [11], [12], [22]. However, end-to-end frameworks usually only show effectiveness in lane-keeping scenarios without lane-changing behaviors [11], [12], [22], [23]. Moreover, they do not have a provable safety guarantee. We also show in our experiments that vehicles have zigzag trajectories using end-to-end learning without explicitly deciding to change lanes but directly deciding steering angle and acceleration. Therefore, the end-to-end framework does not fit our CAV problem.

When considering lane-changing behaviors, it is popular to separate the learning and control modules [24], [25], [26], [27]. The learning module makes a high-level decision, such as "go straight", "go left" [25], or "yield to another vehicle" [24]. Then the control module implements the high-level decision with a safety guarantee [25]. Therefore, we design a discrete action space with continuous control inputs to execute the lane-keeping or lane-changing maneuver in our proposed MARL algorithm. Existing literature only considers the learning and control for a single autonomous vehicle, while in this work we solve the more challenging learning and control problem for a multi-vehicle system.

B. Multi-Agent Reinforcement Learning

Existing multi-agent reinforcement learning (MARL) literature [28] has not fully solved the challenges for CAVs. How communication among agents will improve systems' safety or efficiency in policy learning has not been addressed. Recent advancements like multi-agent deep deterministic policy gradient (MADDPG) [18], the attention mechanism [29], cooperative MARL [30], [31], [32], [33], [34] and league training [35], do not specify communication among agents or safety guarantees for the learned policy. A recent MARL work designs a scalable actor-critic algorithm for networked systems [36], but the method cannot be applied to physical systems like CAVs or provide critical safety guarantees. Moreover, their localized policy only relies on the local state, while in our design the localized policy utilizes the information sharing capability of CAVs and the shared states from neighbors. Therefore, we consider a novel problem of CAVs' planning with information sharing and design a safe MARL algorithm with a scalable critic function and provable safety guarantees.

C. Safe RL and MARL

Safe reinforcement learning is an increasingly important research area for real-world safety-critical applications [37], [38]. The existing safe RL methods for single-agent RL cannot be directly used to solve the challenges in MARL considered in this work [22], [39], [40]. The Monte Carlo tree search method is used to search for safe actions in single-agent RL [39], but the search space grows combinatorially large without a formal safety guarantee in MARL. Control barrier functions (CBFs) have been applied to guarantee the safety learning of continuous action space for a single vehicle [22]. The safe RL in [40] aims to maximize the expected reward while keeping safety constraints for a single agent. However, we consider a more challenging multi-agent problem with discrete action and continuous control inputs.

The existing safe MARL methods either cannot solve the CAV challenges or interrupt the learning process of the MARL agents. Safe MARL methods mainly have two types: constrained MARL or shielding for exploration. In constrained MARL [41], agents maximize the total expected return while keeping the costs lower than the designed bounds. However, the constraints in these methods cannot explicitly represent the safety requirement at every timestep of a physical dynamic system like CAVs, and safety constraints can be violated in practice. The model predictive shielding (MPS) algorithm

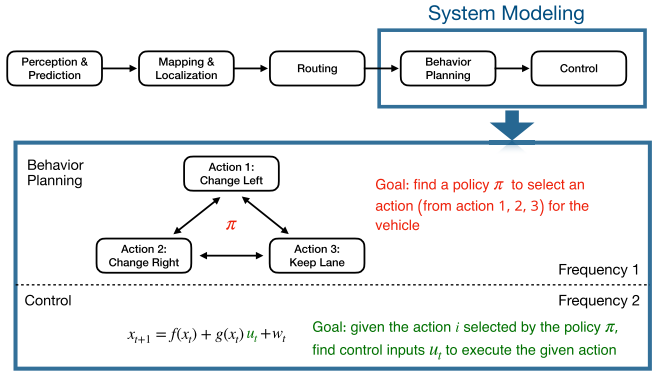


Fig. 1. The system modeling of the behavior planning and control problem considered in this work. We design a novel safe behavior planning and control framework with decentralized training and decentralized execution to tackle the new challenges for CAVs.

provably guarantees safety for any learned MARL policy [42], [43]. The basic idea is to use a backup controller to override the learned policy by dynamically checking whether the learned policy can maintain safety. However, this overriding interrupts the learning process. Our proposed algorithm maps any action in the action space to a safe action such that the MARL agent can keep learning without being interrupted, and guarantees the safety of both training and execution of MARL considering vehicles' physical dynamics.

III. Problem Description and Solution Overview

We consider behavior planning (such as deciding when to change or keep lanes) and control of CAVs to improve traffic efficiency with safety guarantees. The driving environment is a multiple-lane freeway mixed with CAVs and human-driven vehicles, and our decision-making framework is designed for the CAVs. Human-driven vehicles are considered part of the environment and can be observed by the onboard sensors.

We assume that each CAV receives the following information from its neighboring CAVs by V2V communication:

- Each neighboring CAV's position, velocity, and acceleration.
- Each neighboring CAV's vision information (detection results) captured by the onboard camera and LIDAR sensors and processed by a convolutional neural network.
- Each neighboring CAV's action in the action set {Change Left, Change Right, Keep Lane}.

Sharing vehicle's states and actions has already been used in many CAV applications [6], [8], [44], [45], our V2V communication demand is satisfied by the recent V2V communication advances [2], [4].

One typical workflow for an autonomous vehicle includes perception, prediction, mapping and localization, routing, behavior planning, and control [25]. We focus on the last two modules: the behavior planning module to determine whether to change or keep lanes; the control module to control the steering angle and the acceleration. The system modeling is shown at the bottom of Fig. 1. In our framework, the behavior planning module selects one of the discrete actions. We aim to find a policy for each CAV to decide the action to improve traffic efficiency with safe control inputs. For an autonomous vehicle, the behavior planning module and the control module work in different frequencies. One example is that the behavior planning module updates at 2Hz and the control module updates at 100Hz [46]. This is one reason why we separate the behavior learning and control modules.

In summary, our goal is to design a behavior planning and control framework for CAVs to improve the transportation system's safety and efficiency.

A. Problem Formulation

In this section, we formulate the problem for the MARL-based behavior planning and the continuous space physical dynamic control module in Fig. 1.

1) Behavior Planning: We consider n CAVs with an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ as the communication connections. The nodes $\mathcal{N} = \{1, \dots, n\}$ is the set of CAVs and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the edge set. Each vehicle i is associated with an action $a^i \in \mathcal{A}^i$ and a state $s^i \in \mathcal{S}^i$. The action set $\mathcal{A}^i$ is {Keep Lane (KL), Change Left (CL), and Change Right (CR)}. The state s^i of each vehicle is {position, velocity, acceleration, vision information}, where the vision information is captured by the onboard camera and LIDAR sensors and processed according to the weight-pruned CNN to be introduced in Section IV-C. The global state is $s = (s^1, \dots, s^n) \in \mathcal{S} := \mathcal{S}^1 \times \cdots \times \mathcal{S}^n$. The global action is $a = (a^1, \dots, a^n) \in \mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$.

Each vehicle has a stage-wise reward function $r^i(s^i, a^i)$ that depends on its local state and action. The global stage-wise reward is $r(s, a) = \frac{1}{n}\sum_{i=1}^{n} r^i(s^i, a^i)$. Each vehicle is associated with a localized policy $\pi^i(a^i|s^{N^i})$ given the joint states s^{N^i} of its neighbors (including itself). We use $\pi(a|s)$ to denote the joint policy of n CAVs. The action-value function for the joint policy π is

$$\begin{aligned} Q(s,a) &= \mathbb{E}_{a_k \sim \pi(a|s)} \left[\sum_{k=0}^{\infty} \gamma^k r_{k+1}(s_k, a_k) | s_0 = s, a_0 = a \right] \\ &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{a_k \sim \pi(a|s)} \\ &\quad \left[\sum_{k=0}^{\infty} \gamma^k r^i_{k+1}(s^i_k, a^i_k) | s_0 = s, a_0 = a \right] \\ &:= \frac{1}{n} \sum_{i=1}^{n} Q^i(s,a), \end{aligned} \tag{1}$$

where $Q^i(s, a)$ is the action value function for each vehicle i and $\gamma \in (0, 1)$ is a discount factor.

The objective is to find a policy π to maximize the total expected return of n CAVs:

$$G = \mathbb{E}_{s_0 \sim p_0} \mathbb{E}_{a_k \sim \pi(a|s)} \left[\sum_{k=0}^{\infty} \gamma^k r_{k+1}(s_k, a_k) \mid s_0 \right], \tag{2}$$

where p_0 is the initial state distribution.

2) Control: As shown in Fig. 1, we consider the physical dynamics for each vehicle i in the control affine form:

$$x_{t+1} = f(x_t) + g(x_t)u_t + w_t.^{1} \tag{3}$$

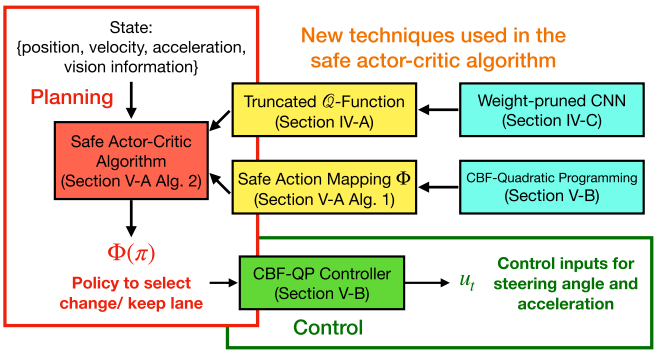


Fig. 2. Our proposed learning and control framework for the behavior planning and control problem. We design a safe actor-critic MARL algorithm to learn a policy to select actions. We use two new techniques in our algorithm: truncated $\mathcal{Q}$-function and safe action mapping. We also introduce a CBF-QP controller to generate control inputs for steering angle and acceleration with provable safety guarantees.

with f and g locally Lipschitz ,[2] $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ is the control state and $u \in \mathcal{U} \subset \mathbb{R}^{m_u}$ is the control input, w is the bounded noise that satisfies $\|w_t\| \leq W$ for all $t \geq 0$.

We want to find control inputs such that x_t is always in a safe set $\mathcal{C} \subseteq \mathcal{X}$. In other words, the ego vehicle stays within a given range or a bounding box of its planned trajectory and does not collide with other vehicles or obstacles.

Note that we use t and k to distinguish between the controller timestep and reinforcement learning timestep. We want to find a policy $\pi(a_k|s_k)$ to maximize the total expected return G; meanwhile, there should exist control inputs $u_t \in \mathcal{U}$ for each vehicle to execute the action and guarantee that $x_t \in \mathcal{C}$ for all $t \geq 0$ given the initial state $x_0 \in \mathcal{C}$.

B. Our Novel Solution Overview

In this work, we design a novel behavior learning and control framework with *a safe actor-critic algorithm for decentralized training and decentralized execution* as shown in Fig. 2. We assume that all CAVs share their states (including weight-pruned CNN processed vision information) and actions with others. Each CAV uses minibatch gradient descent to learn the truncated $\mathcal{Q}$-function as a critic to be introduced in Section IV. Then each CAV learns a localized policy $\Phi(\pi(a^i|s^{N^i}))$ based on the learned critic with a safe action mapping Φ to be introduced in Section V. The details of this safe actor-critic algorithm will be introduced in Section V-A.

IV. Truncated $\mathcal{Q}$-Function

In this section, we introduce the design of truncated action value function $\mathcal{Q}$ to solve the first two challenges mentioned in the introduction such that the training process utilizes the information sharing capability of neighboring CAVs instead of relying on the global states and actions of all agents. Moreover, the joint state and action spaces of the truncated $\mathcal{Q}$-function do not grow with the total number of CAVs.

We first define a truncated $\mathcal{Q}$-function in subsection A to approximate the global Q-function and prove that the approximation error is bounded in Theorem 1 and Theorem 2. This is one main result of this work. Then, we introduce the truncated $\mathcal{Q}$-network design. At last, we introduce how to overcome the challenges of sharing vision information (required in the input of truncated $\mathcal{Q}$-function) using the weight-pruned CNN.

A. Truncated $\mathcal{Q}$-Function Approximation

To tackle the new challenges for the CAVs, we introduce how to use the truncated $\mathcal{Q}$-function to approximate the global Q-function. The key idea is that the further the vehicles are away from the ego vehicle,[3] the less impact they will have on the ego vehicle. To illustrate this idea, we first introduce one assumption:

Assumption 1: At each time instant k, each vehicle i's next state s^i_{k+1} is independently generated and only depends on its neighbors, hence,

$$P(s_{k+1}|s_k, a_k) = \prod_{i=1}^{n} P(s^i_{k+1}|s^{N^i}_k, a^{N^i}_k), \tag{4}$$

where N^i means the 1-hop neighborhood of vehicle i including itself, s^{N^i} and a^{N^i} are the joint states and joint actions of N^i.

This assumption follows the idea that vehicles far away from the ego vehicle do not have a direct influence on the ego vehicle. Then we introduce the definition of the exponential decay property. We use N^i_κ to denote the κ-hop neighborhood of vehicle i for $\kappa \geq 1$ (including all vehicles whose graph distance to i is less than or equal to κ). We use N^{-i}_κ to denote the vehicles that are outside of vehicle i's κ-hop neighborhood. Then the global state s can be divided into two parts $(s^{N^i_\kappa}, s^{N^{-i}_\kappa})$. Similarly, we divide the global action a into $(a^{N^i_\kappa}, a^{N^{-i}_\kappa})$. The exponential decay property is defined for the global Q^i-function as follows.

Definition 1 (Exponential decay property [36]): The (c, ρ)-exponential decay property holds for the global Q^i-function in (1) if there exists some $c > 0$ and $0 < \rho < 1$ such that for any $i \in \mathcal{N}$, $\forall s^{N^i_\kappa} \in \mathcal{S}^{N^i_\kappa}$, $\forall s^{N^{-i}_\kappa} \in \mathcal{S}^{N^{-i}_\kappa}$, $\forall s'^{N^{-i}_\kappa} \in \mathcal{S}^{N^{-i}_\kappa}$, $\forall a^{N^i_\kappa} \in \mathcal{A}^{N^i_\kappa}$, $\forall a^{N^{-i}_\kappa} \in \mathcal{A}^{N^{-i}_\kappa}$, $\forall a'^{N^{-i}_\kappa} \in \mathcal{A}^{N^{-i}_\kappa}$, it holds that $|Q^i(s^{N^i_\kappa}, s^{N^{-i}_\kappa}, a^{N^i_\kappa}, a^{N^{-i}_\kappa}) - Q^i(s^{N^i_\kappa}, s'^{N^{-i}_\kappa}, a^{N^i_\kappa}, a'^{N^{-i}_\kappa})| \leq c\rho^\kappa$.

This property shows that the impact of the states and the actions of the κ-hop neighborhood on the global Q^i-function decreases exponentially as κ increases. The following Theorem 1 shows that the exponential decay property holds for the global Q^i-function in our CAV MARL problem. The proof of this theorem is postponed to Appendix B. This property is utilized for CAVs to get rid of the dependence on the global states and actions.

[1] The controller uses the same design for each vehicle i. We drop the superscript i in control states and control inputs when there is no confusion.

[2] A function $f : \mathbb{R}^{n_x} \to \mathbb{R}$ is called Lipschitz continuous over $\mathcal{X}$ if and only if there exists a constant $K \geq 0$ such that for any $x_1, x_2 \in \mathcal{X}$,

$$|f(x_1) - f(x_2)| \leq K |x_1 - x_2| .$$

A function $f : \mathbb{R}^{n_x} \to \mathbb{R}$ is called locally Lipschitz continuous if for every $x \in \mathcal{X}$ there exists a neighborhood $\mathcal{X}_x$ of x such that f is Lipschitz continuous over $\mathcal{X}_x$.

[3] We refer to the vehicle we focus on (vehicle i) as the ego vehicle.

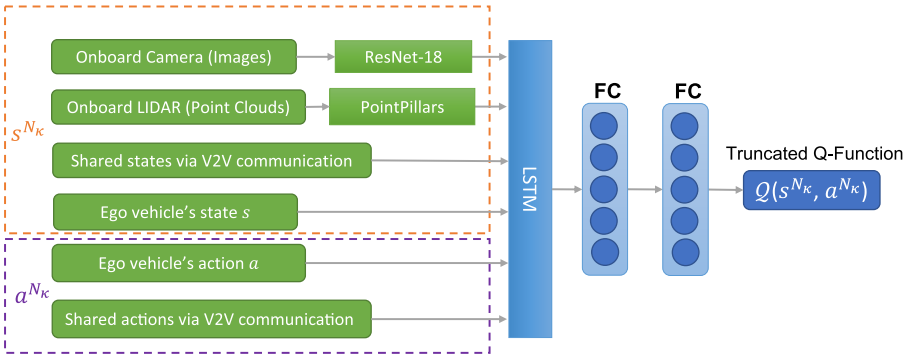


Fig. 3. The truncated $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$ network for the behavior planning with the LSTM (long short-term memory) layer and FC (fully connected) layers. We use truncated $\mathcal{Q}$-function to approximate the centralized critic such that the training process utilizes the information sharing capability of CAVs instead of relying on the global states and actions.

Theorem 1: If for all $i \in \mathcal{N}$, the reward r^i is upper bounded by $\bar{r}$, then the $(\frac{\bar{r}}{1-\gamma}, \sqrt{\gamma})$-exponential decay property holds for Q^i in (1) under Assumption 1, where γ is the discount factor.

Proof: See Appendix B. ∎

Theorem 1 shows that with the exponential decay property, the states and actions of vehicles far away have limited contribution to estimating the global Q^i-function for the agent i. Based on this property, we define a truncated $\mathcal{Q}$-function as follows to approximate the global Q^i-function.

Definition 2 (Truncated $\mathcal{Q}$-function): The truncated $\mathcal{Q}$-function for each vehicle i only takes in the states and actions of the κ-hop neighborhood of vehicle i as follows:

$$\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i}) := Q^i(s^{N_\kappa^i}, \mathbf{0}, a^{N_\kappa^i}, \mathbf{0}), \tag{5}$$

where $s^{N_\kappa^{-i}}$ and $a^{N_\kappa^{-i}}$ are selected as $\mathbf{0}$.[4]

The benefit of using the truncated $\mathcal{Q}$-function to approximate the global Q-function is that the truncated $\mathcal{Q}$-function only needs the states and actions of the κ-hop neighborhood of vehicle i instead of the global states and global actions. The joint state and action spaces of the truncated $\mathcal{Q}$-function do not grow with the total number of CAVs. We have the following theorem to give a bound of the approximation error by using the truncated $\mathcal{Q}$-function.

Theorem 2: If for all $i \in \mathcal{N}$, the reward r^i is upper bounded by $\bar{r}$, then, under Assumption 1, it holds that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\left|\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i}) - Q^i(s, a)\right| \leq \frac{\bar{r}}{1-\gamma}\gamma^{\frac{\kappa}{2}}. \tag{6}$$

Proof: Since $\forall i \in \mathcal{N}$, $r^i \leq \bar{r}$, the $(\frac{\bar{r}}{1-\gamma}, \sqrt{\gamma})$-exponential decay property holds using Theorem 1. According to Definition 1, $\forall i \in \mathcal{N}, \forall s^{N_\kappa^i} \in \mathcal{S}^{N_\kappa^i}, \forall (s^{N_\kappa^{-i}}, s_*^{N_\kappa^{-i}}) \in \mathcal{S}^{N_\kappa^{-i}}, \forall a^{N_\kappa^i} \in \mathcal{A}^{N_\kappa^i}, \forall (a^{N_\kappa^{-i}}, a_*^{N_\kappa^{-i}}) \in \mathcal{A}^{N_\kappa^{-i}}$, we have $|Q^i(s^{N_\kappa^i}, s_*^{N_\kappa^{-i}}, a^{N_\kappa^i}, a_*^{N_\kappa^{-i}}) - Q^i(s^{N_\kappa^i}, s^{N_\kappa^{-i}}, a^{N_\kappa^i}, a^{N_\kappa^{-i}})| \leq \frac{\bar{r}}{1-\gamma}\gamma^{\kappa/2}$. That is to say $|Q^i(s^{N_\kappa^i}, s_*^{N_\kappa^{-i}}, a^{N_\kappa^i}, a_*^{N_\kappa^{-i}}) - Q^i(s, a)| \leq \frac{\bar{r}}{1-\gamma}\gamma^{\kappa/2}$. By plugging in $s_*^{N_\kappa^{-i}}$ and $a_*^{N_\kappa^{-i}}$ as $\mathbf{0}$, we have $\left|\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i}) - Q^i(s, a)\right| \leq \frac{\bar{r}}{1-\gamma}\gamma^{\kappa/2}$. ∎

Theorem 2 shows that the approximation error of the truncated action value function $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$ is bounded by $\frac{\bar{r}}{1-\gamma}\gamma^{\kappa/2}$. As the communication technology develops, the κ becomes larger and this approximation error decreases exponentially small. Information shared from neighbors shows benefit in getting a scalable and well-approximated $\mathcal{Q}$-function for MARL-based CAV behavior learning. Results in Theorem 1 and Theorem 2 show that the ego vehicle only needs to get the states and actions of its neighboring vehicles rather than that of all CAVs to learn the truncated $\mathcal{Q}$-function.

B. Truncated $\mathcal{Q}$-Network

The $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$-network we use for the behavior planning is shown in Fig. 3. We consider the action set $\mathcal{A}$ includes {Keep Lane (KL), Change Left (CL), and Change Right (CR)}. The inputs of the truncated $\mathcal{Q}$-network include:

- Ego vehicle's vision information including onboard camera images and LIDAR data (point clouds);
- Shared states (including shared vision information) and actions from neighboring CAVs via V2V;
- Ego vehicle's state s^i and action a^i.

The input design is based on Theorem 2. Since we can use truncated $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$ to approximate $Q^i(s, a)$ with bounded approximation error, the input of the $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$-network does not rely on the global s and the global action a. Meanwhile, the joint state and action spaces do not grow even in a large-scale system. In this $\mathcal{Q}$-network, we adopt the ResNet-18 [47] structure to process the images and the PointPillars [48] to process the point clouds. The implementation details will be introduced in the experiment section.

Remark 1: Information sharing can be very useful for micro control as it enables coordination and collaboration among different entities. However, we emphasize that to achieve system optimality, accurate prediction for decision-making is equally crucial. While information sharing is an important initial step toward achieving system optimality, it must be followed by accurate prediction and decision-making. Therefore, the goal should not be limited to sharing information, but rather to use this information to make informed decisions.

C. Weight-Pruned CNN For Truncated $\mathcal{Q}$-Function Learning

When updating the truncated $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$-network, we need the κ-hop neighborhood's states including vision

[4]We assume that $\mathbf{0}$ represents a valid state or action in $\mathcal{S}^{N_\kappa^{-i}}$ and $\mathcal{A}^{N_\kappa^{-i}}$.

information captured by onboard cameras and LIDAR sensors. Therefore, we assume all CAVs share their states by V2V communication. However, it is unrealistic for CAVs to share the raw camera images and point clouds with others due to the following two reasons: (i) the limited bandwidth of a V2V link prevents sharing the raw camera and LIDAR information among vehicles [49]; (ii) the same shared information on neighboring vehicles needs to be repeatedly learned for lane information extraction, resulting in computing resource waste.

To overcome these challenges, we first process the vision information locally using CNNs and then share the extracted features with neighboring vehicles. We observe that the processing time of point cloud data is significantly longer ($11.16\times$) than that of images. The point cloud data processing becomes the "critical path" of the overall vision process since the raw images and point clouds need to be synchronously processed and shared for behavior planning. Previous research has shown that there exists redundancy in CNN model parameters [50], [51]. Hence, we further develop a weight pruning technique to speed up the slower process.

CNN weight pruning can be used to exploit the redundancy in the parameterization of deep architectures while maintaining the CNN model accuracy. The key idea is to represent a neural network with a simpler model through pruning the redundant weights (whose magnitudes are below a threshold). We formulate the weight pruning problem in an N-layer CNN as:

$$\begin{aligned} \underset{\{\mathbf{W}_i\}}{\text{minimize}} \quad & \mathcal{F}\big(\{\mathbf{W}_i\}_{i=1}^{N}\big), \\ \text{subject to} \quad & \text{cardinality}(\mathbf{W}_i) \le l_i,\ i = 1, \ldots, N, \end{aligned} \tag{7}$$

where $\mathbf{W}_i$ represents the weights in the i-th layer. $\mathcal{F}$ is the CNN loss function with respect to $\mathbf{W}_i$. Here, we use the cross entropy loss as the CNN loss to measure the difference between the real label and the predicted label, and minimize the loss function to obtain a candidate solution that has the lowest error [52]. We use l_i to represent the desired numbers of non-zero weights [53]. Then we retrain the CNN to fine-tune the weights of the remaining connections. Our vision processing method is used to extract features of both autonomous and human-driven vehicles in the mixed traffic environment. The implementation details of the weight-pruned CNN will be introduced in the experiment section.

V. Behavior Planning and Safety

In this section, we introduce our main contribution, the safe actor-critic MARL algorithm, to learn a behavior planning policy $\Phi(\pi)$ to select actions. A safe action mapping Φ based on the control barrier function (CBF) is designed to solve the last challenge mentioned in the introduction to guarantee that the action explored in training and execution is safe with feasible control inputs. The novel safe actor-critic algorithm and the resulting localized policy design can utilize the unique strength of CAVs to gather more information for decision making based on V2V communication. We also introduce the control barrier function (CBF)-based quadratic programming (QP) controller in subsection B to generate control inputs for steering angle and acceleration with provable safety.

Algorithm 1 Safe Action Mapping Φ

```
Input: action set A, action a^i ∈ A ;
Output: safe action a^i ∈ C_A ⊆ A ;
while True do
    if a^i is safe, i.e., the CBF-QP (introduced in
     subsection B) has a feasible solution when
     plugging in a^i then
        return a^i;
    else
        Remove a^i from A;
        if A is not empty then
            find action in A with the highest Q value
             and assign it to a^i;
        else
            return a^i = ES, emergency stop [54];
        end
    end
end
```

A. Behavior Planning Algorithm

Since autonomous driving is a safety-critical application, the behavior policy used to explore different actions must be designed rigorously to avoid potential accidents. The action a generated by the traditional policy for exploration, like the ϵ-greedy method [55], may not be safe to execute without additional checking. For example, lane-changing may lead to collisions with neighboring vehicles. To ensure that there are no collisions during the training process, we design a safe action mapping function Φ to map any action from the action set $\mathcal{A}$ to the safe action set $\mathcal{C}_\mathcal{A} \subseteq \mathcal{A}$.

1) Safe Action Mapping: The safe action mapping Φ is shown in Alg. 1. We use a CBF-QP controller (it is to be introduced in subsection B) to evaluate whether an action is safe or not. If a is safe, then return the safe action a; if not, the controller will search other actions in $\mathcal{A}$ in descending order according to their action value and find a safe one. If all the actions in $\mathcal{A}$ are not safe in the worst case, then the controller will apply the emergency stop (ES) process. In case when the safe action set $\mathcal{C}_\mathcal{A}$ is empty, an emergency stop will be executed, and the corresponding state will be marked unsafe and receive a large penalty. The emergency stop will only be performed in an emergency scenario where all normal actions are not safe [54].

2) Safe Actor-Critic Algorithm: In our behavior planning algorithm design, we use a decentralized training and decentralized execution paradigm. Each CAV learns a truncated critic $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$ and then use it to train a localized policy $\Phi(\pi^i(a^i|s^{N^i}))$ for decentralized execution. We adopt an actor-critic structure so that the actor and the critic can use different information sets. We do not derive a policy directly from the truncated $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$ function but design an actor besides the critic, because we want the policy for each CAV to only depend on its available states information excluding the κ-hop neighbors' actions. Otherwise, there will be a deadlock during execution, and CAVs will wait for each other's actions.

Our main design, the safe actor-critic algorithm, is shown in Alg. 2. We mainly design two new techniques in our algorithm:

Algorithm 2 Safe Actor-Critic Algorithm

Randomly initialize the critic network $\mathcal{Q}^i$ and the actor network π^i for agent i. Initialize target networks $\mathcal{Q}'^i, \pi'^i$. Initialize replay buffers $\mathcal{D}^i$;
for *each episode* **do**
  Initialize the global state s;
  for *each timestep* **do**
    With probability ϵ, select action $a^i = \Phi(\pi^i(s^{N^i}))$ for each agent i, where Φ is the safe action mapping in Alg. 1. With probability $1-\epsilon$, select an action a^i for each agent i in $\{\Phi(a^i)|a^i \in \mathcal{A}\}$ randomly;
    Execute actions $a = (a^1, \ldots, a^n)$ and observe reward r and new state s';
    for *each agent* i **do**
      Store $(s^{N^i_\kappa}, a^{N^i_\kappa}, r^i, s'^{N^i_\kappa})$ in replay buffer $\mathcal{D}^i$;
      Sample a random minibatch with size K from $\mathcal{D}^i$;
      Set $y^i_k = r^i_k + \gamma \mathcal{Q}'^i(s'^{N^i_\kappa}_k, a'^{N^i_\kappa}_k)|_{a'^i=\pi'^i(s'^{N^i})}$;
      Update critic by minimizing the loss $\mathcal{L}(\theta^i) = \frac{1}{K}\sum_k (y^i_k - \mathcal{Q}^i(s^{N^i_\kappa}_k, a^{N^i_\kappa}_k))^2$;
      Update actor using the gradient $\nabla_{\theta^i} J = \frac{1}{K}\sum_k \nabla_{\theta^i}\pi^i(s^{N^i})\nabla_{a^i}\mathcal{Q}^i(s^{N^i_\kappa}_k, a^{N^i_\kappa}_k)$ where $a^i = \pi^i(s^{N^i})$;
      Update target networks: $\theta'^i \leftarrow \tau\theta^i + (1-\tau)\theta'^i$.
    end
    Set $s \leftarrow s'$;
  end
end

- *Truncated Q-function*: Each vehicle learns a truncated $\mathcal{Q}$-function in decentralized training such that the training process does not rely on the global states and the global actions. The joint state and action spaces of the truncated $\mathcal{Q}$-function do not grow in a large-scale CAV system.
- *Safe action mapping*: We map any action in the action space to the safe action set so that the training and execution have provable safety guarantees.

The transition experience in Alg. 2 is represented by $(s^{N^i_\kappa}, a^{N^i_\kappa}, r^i, s'^{N^i_\kappa})$, where $s^{N^i_\kappa}$ and $a^{N^i_\kappa}$ are the current states and actions of the κ-hop neighborhood including vehicle i, r^i is the reward, $s'^{N^i_\kappa}$ is the next states of the κ-hop neighborhood including vehicle i. This algorithm learns a truncated critic $\mathcal{Q}^i(s^{N^i_\kappa}, a^{N^i_\kappa})$ by minimizing the Bellman loss:

$$\mathcal{L}(\theta^i) = \mathbb{E}_{s^{N^i_\kappa}, a^{N^i_\kappa}, r^i, s'^{N^i_\kappa}}\left[(y^i - \mathcal{Q}^i(s^{N^i_\kappa}, a^{N^i_\kappa}; \theta^i))^2\right], \quad (8)$$

where $y^i = r^i + \gamma \cdot \mathcal{Q}'^i(s'^{N^i_\kappa}, a'^{N^i_\kappa}; \theta'^i)|_{a'^i=\pi'^i(s'^{N^i})}$, γ is the discount factor, $\mathcal{Q}'^i$ is the target network for the critic, π'^i is the target network for the actor. The safe action mapping assures the policy used to produce a new transition experience is safe. The generated experience is stored in a replay buffer $\mathcal{D}^i$. When training the $\mathcal{Q}^i$-network, a mini-batch is sampled from $\mathcal{D}^i$ to decorrelate data. Then we use this truncated critic to train a localized actor using the following gradient

$$\nabla_{\theta^i} J = \mathbb{E}_{s^{N^i_\kappa}, a^{N^i_\kappa} \sim \mathcal{D}^i}\left[\nabla_{\theta^i}\pi^i(s^{N^i})\nabla_{a^i}\mathcal{Q}^i(s^{N^i_\kappa}, a^{N^i_\kappa})\right]. \quad (9)$$

Remark 2: The safe actor-critic algorithm in Alg. 2 is scalable for a large-scale CAV system for the following two reasons: (i) The Alg. 2 learns a truncated action value function $\mathcal{Q}^i(s^{N^i_\kappa}, a^{N^i_\kappa})$ that only requires the states and actions of vehicle i's κ-hop neighborhood, i.e., the joint state and action spaces of the truncated $\mathcal{Q}$-function do not grow with the total number of CAVs; (ii) The policy $\Phi(\pi^i(a^i|s^{N^i}))$ only depends on the local states.

B. Safety For the Ego Vehicle

This section introduces the control barrier function (CBF)-based quadratic programming (QP) controller design for the ego vehicle to find the safe control inputs for the steering angle and the acceleration. It is also used in the safe action mapping algorithm (Alg. 1) to check the safety of CAVs' actions. Since the controller design is the same for each vehicle i, we drop the superscript i when there is no confusion in this section. Before we give the formulation of the CBF-QP controller, we first introduce the concept of control barrier function, safe set, and forward invariant.

Definition 3 (Control barrier function (CBF) [22], [56]): A mapping $h : \mathcal{X} \to \mathbb{R}$ is a discrete-time exponential control barrier function for dynamic system (3) if

1) $h(x_0) \geq 0$ and,
2) there exists a control input $u_t \in \mathcal{U}$ and $\eta \in (0, 1]$ such that for all $t \in \mathbb{N}_+$,

$$h(f(x_t) + g(x_t)u_t + w_t) + (\eta - 1)h(x_t) \geq 0, \quad (10)$$

Definition 4 (Safe set [56], [57]): Denote the safe set $\mathcal{C}$ as

$$\mathcal{C} : \{x \in \mathcal{X} \in \mathbb{R}^{n_x} \mid h(x) \geq 0\}, \quad (11)$$

where $h : \mathcal{X} \to \mathbb{R}$ is a discrete-time exponential CBF.

Definition 5 (Forward invariant [56], [57]): For any initial state $x_0 \in \mathcal{C}$, if the state x_t is in $\mathcal{C}$ for all $t \geq 0$, then the set $\mathcal{C}$ is said to be forward invariant.

The system is safe with respect to set $\mathcal{C}$ if $\mathcal{C}$ is forward invariant, i.e., the system state x always remains within the safe set. The CBF is a condition to add the forward invariant property to the system's state x. For the CAV problem, we consider an affine barrier function with the form $h = p^\mathsf{T}x + q$, $(p \in \mathbb{R}^{n_x}, q \in \mathbb{R})$. This restriction means the set $\mathcal{C}$ is a polytope constructed by the intersecting of half-spaces since the safe area for a vehicle is usually represented by a bounding box [58]. By applying a Frenét frame, the bounding box also works for curve roads [59].

We formulate the following Quadratic Programming (QP) based on the CBF condition (10) that can be solved efficiently at each timestep:

$$\begin{aligned} \underset{u_t, \zeta}{\text{argmin}} \quad & \|u_t - U \cdot a\|^2 + M\zeta \\ \text{s.t.} \quad & p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t)u_t + q - p^\mathsf{T} W \cdot \mathbf{1} \\ & \geq (1-\eta)h(x_t) - \zeta \\ & \zeta \geq 0, \end{aligned} \quad (12)$$

where M is a large constant, a is the action to change/keep lane, and $U \cdot a$ is the control reference for action a. For any action selected by the policy, we use state-of-the-art trajectory tracking [58], [60] to calculate the control references. The implementation details of the CBF-QP controller will be introduced in the experiment section for our CAV problem.

The following theorem shows that the control inputs calculated by solving (12) can guarantee the safety of the system $x_{t+1} = f(x_t) + g(x_t)u_t + w_t$ in (3) with bounded noise.

Theorem 3: When the physical dynamics of each autonomous vehicle satisfy (3) with bounded noise $\|w_t\| \leq W$, *if there exists* $\eta \in (0, 1]$ *such that the QP (12) has a solution for all* $x_t \in \mathcal{C}$ *and the solution satisfies* $\zeta \leq Z$, *then the controller derived from (12) renders set* $\mathcal{C}' : \{x \in \mathbb{R}^{n_x} \mid h'(x) = h(x) + \frac{Z}{\eta} \geq 0\}$ *forward invariant.*

Proof: Since $x_t \in \mathcal{C}$, we have $h(x_t) \geq 0$ according to the Definition 4. From $Z \geq \zeta \geq 0 \geq -\eta h(x_t)$, we have $h(x_t) + \frac{Z}{\eta} \geq 0$. Thus, $x_t \in \mathcal{C}'$ with $h'(x_t) \geq 0$. Also, we have

$$\begin{aligned} h(x_{t+1}) &= p^\mathsf{T} x_{t+1} + q \\ &= p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t) u_t + p^\mathsf{T} w_t + q \\ &\geq p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t) u_t + q - p^\mathsf{T} W \cdot \mathbf{1} \\ &\geq (1-\eta) h(x_t) - \zeta \geq (1-\eta) h(x_t) - Z, \end{aligned} \tag{13}$$

where the first inequality follows $\|w_t\| \leq W$ and the second inequality follows the inequality constraints in the CBF-QP (12). Adding Z/η to both sides of (13), we have

$$\begin{aligned} h(x_{t+1}) + \frac{Z}{\eta} &\geq (1-\eta)\left[h(x_t) + \frac{Z}{\eta}\right], \\ h'(x_{t+1}) &\geq (1-\eta) h'(x_t) \geq 0. \end{aligned} \tag{14}$$

Thus, $x_{t+1} \in \mathcal{C}'$ and $\mathcal{C}'$ is forward invariant. ■

The value of Z denotes how large the CBF condition (10) is violated from the original $h(x_t)$. In this case, the safety condition should be formulated according to the set $\mathcal{C}'$.

Theorem 3 shows that the safe set $\mathcal{C}'$ is forward invariant using the control inputs calculated by the CBF-QP in (12), that is to say, the ego vehicle's control states are always in the safe set and the ego vehicle is guaranteed to be safe. We also use (12) in Alg. 1 for a safety checking by plugging in the corresponding action a. As long as there is a feasible solution to (12), we can find control inputs for this behavior action a such that the control states are always in the safe set, in other words, the action a is a safe action. In Appendix C, we show how to formulate the CBF-QP for a system when we do not need to consider the noise.

C. Safety For the Multi-Agent System

At last, we show the connection between the safety of a single agent and the multi-agent system. Note that the CBF-QP in (12) is defined for a single CAV, and x_t in (12) only includes the control state for a single CAV. Theorem 3 shows the safety guarantee of each CAV. Consider the multi-agent system with n agents, the joint control state is $\mathbf{x} = (x^1, \ldots, x^n) \in \mathbf{X} := \mathcal{X}^1 \times \cdots \times \mathcal{X}^n$, the joint control input is $\mathbf{u} = (u^1, \ldots, u^n) \in \mathbf{U} := \mathcal{U}^1 \times \cdots \times \mathcal{U}^n$. The safe set defined in Definition 4 can be extended for the joint control state as:

$$\mathbf{C} : \{\mathbf{x} \in \mathbf{X} \in \mathbb{R}^{n \times n_x} \mid \forall i \in \mathcal{N}, h^i(x^i) \geq 0\}. \tag{15}$$

The safety of a multi-agent system can be formally defined as follows:

Definition 6 (Safety of a Multi-Agent System): For any initial state in the safe set $\mathbf{x}_0 \in \mathbf{C}$, if the joint control state $\mathbf{x}_t$ is in $\mathbf{C}$ for all $t \geq 0$, then the multi-agent system is safe with respect to the safe set $\mathbf{C}$.

We use a decentralized manner to find the safe control inputs for each agent: at each time t, each agent solves its own CBF-QP in (12) to find the safe control inputs for itself. Then we have the following Proposition 1 to show the safety of the multi-agent system when all agents follow the CBF-QP in (12).

Proposition 1: If there exists $\eta^i \in (0, 1]$ such that the QP (12) for each agent $i \in \mathcal{N}$ has a solution for all $x_t^i \in \mathcal{C}^i$ and the solution satisfies $\zeta^i \leq Z^i$, then the controller derived from (12) for each agent guarantees the safety of the multi-agent system with respect to $\mathbf{C}' : \{\mathbf{x} \in \mathbb{R}^{n \times n_x} \mid \forall i \in \mathcal{N}, h^{i\prime}(x^i) = h^i(x^i) + \frac{Z^i}{\eta^i} \geq 0\}$.

Proof: According to Theorem 3, for each $i \in \mathcal{N}$, the controller derived from (12) for each agent renders the safe set $\mathcal{C}^{i\prime} : \{x^i \in \mathbb{R}^{n_x} \mid h^{i\prime}(x^i) = h^i(x^i) + \frac{Z^i}{\eta^i} \geq 0\}$ forward invariant. Therefore, starting from any safe joint state $\mathbf{x}_0 \in \mathbf{C}'$, the control state x_t^i is within $\mathcal{C}^{i\prime}$ for all $i \in \mathcal{N}$ and all $t \geq 0$ according to the forward invariant property defined in Definition 5. Thus, the joint control state $\mathbf{x}_t$ is within the $\mathbf{C}' : \{\mathbf{x} \in \mathbb{R}^{n \times n_x} \mid \forall i \in \mathcal{N}, h^{i\prime}(x^i) = h^i(x^i) + \frac{Z^i}{\eta^i} \geq 0\}$, and the multi-agent system is safe with respect to $\mathbf{C}'$. ■

Proposition 1 shows that each agent only needs to care about its own safety if all agents follow the same form of safety rules, i.e., using the CBF-QP in (12) to find control inputs, and the entire multi-agent system will be safe. This decentralized controller design can significantly improve the scalability of the CAV system and avoid using a centralized controller with a combinatorially large joint control state space.

VI. Experiment

In this section, we first show the implementation details and the evaluation of the weight-pruned CNN in subsection A. In subsection B, we show our safe actor-critic algorithm improves traffic efficiency under different CAV ratios and different traffic densities. We also evaluate the truncated $\mathcal{Q}$-function technique in this set of experiments. Then in subsection C, we show our approach guarantees the safety of the CAVs. In subsection D, we construct an obstacle-at-corner scenario to show the benefit of the shared vision. In all experiments, we use CARLA [20], an open-source simulator that supports the development training and validation of autonomous driving systems, to validate our proposed safe actor-critic algorithm. We select CARLA because it has a high-fidelity simulation and supports a wide range of sensors.

All vehicles run on a 3-lane freeway as shown in Fig. 4. Each CAV in CARLA has a camera for capturing RGB images and a LIDAR for generating point clouds. The resolution of the camera image is $375 \times 1,242$ pixels. We set row anchors

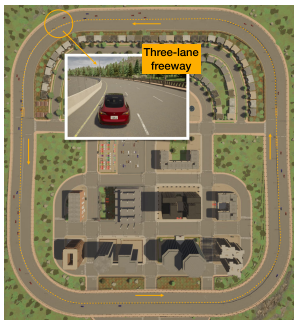


Fig. 4. The example scenario of a 3-lane freeway in CARLA [20]. Vehicles are scattered on the outer loop of the map "Town 05". The environment is mixed with both autonomous and human-driven vehicles.

TABLE I
SAFE ACTOR-CRITIC HYPERPARAMETERS

Parameter	Value
optimizer	Adam
learning rate	0.01
discount factor	0.9
replay buffer size	10^6
hidden size in FC and LSTM layers	128
minibatch size	64
activation function	ReLU

ranging from 100 to 370 with intervals of 10, and the number of grids is 100. We scale each image to 288×800. Each point cloud from LIDAR is stored with the 3 coordinates (the ego vehicle being the origin), representing forward, left, and up respectively, and an additional reflectance value. To augment the dataset, we apply a random mirroring flip along the forward axis, and a global rotation and scaling. In evaluation, we set forward and up axes' resolution to 0.16 m, max number of pillars to 12,000, and max number of points per pillar to 100. Images and point clouds are paired under the same timestamp and are processed jointly under the CNN.

We assume all CAVs share their states (including vision information) and actions with others. Each CAV uses mini-batch gradient descent to learn the truncated $\mathcal{Q}^i(s^{N_\kappa^i}, a^{N_\kappa^i})$ critic introduced in Section IV. Then each CAV learns a localized policy $\Phi(\pi^i(a^i|s^{N^i}))$ based on the learned critic with a safe action mapping Φ as our proposed Alg. 2. The hyperparameters of Alg. 2 are shown in Table I.

The host machine adopted in our experiments is a server configured with Intel Core i9-10900X processors and four NVIDIA RTX2080Ti GPUs. Our experiments are performed on Python 3.7.6, GCC7 7.5, PyTorch 1.6.0, and CUDA 11.0.

A. Safe Actor-Critic Algorithm Implementation Details

1) Reward Function: In the implementation of our safe actor-critic algorithm, we consider a stage-wise reward for the ego vehicle to improve velocity v^i and comfort c^i. The comfort of a vehicle (for passenger's experience) is defined based on its acceleration and action as follows:

$$c^i(\dot{v}^i, a^i) = \begin{cases} 3, & \text{if } |\dot{v}^i| < \Theta \text{ and } a^i = KL; \\ 2, & \text{if } |\dot{v}^i| \geq \Theta \text{ and } a^i = KL; \\ 1, & \text{if } a^i = CL/CR; \\ 0, & \text{if in } ES. \end{cases} \tag{16}$$

Fig. 5. The ego vehicle should keep a safe distance from the front vehicle when keeping lane. To show the idea of safe distance, each vehicle is treated as a mass point at the CoG in this figure.

where $\dot{v}^i$ is the acceleration, $a^i \in \mathcal{A}$ is the MARL action, Θ is a predefined threshold. The reward function for vehicle i is defined as:

$$r^i(s^i, a^i) = \omega \cdot v^i + c^i(\dot{v}^i, a^i), \tag{17}$$

where ω is a positive trade-off weight.

2) The CBF-QP: In the implementation of the CBF-QP controller (12), the details of the system model are in Appendix A. The control state $x_t = [p_x, p_y, \psi, v]^\mathsf{T} \in \mathcal{X}$ includes the two coordinates of the center of gravity (CoG), its orientation and velocity. The control input $u_t = [\tan(\delta), \dot{v}] \in \mathcal{U}$ includes the tangent of the steering angle and the acceleration. The action $a = \{a_j | a_j \in \mathcal{A}\}$ is selected based on the policy learned by the safe actor-critic algorithm. In the controller module, we use standard basis vectors in $\mathbb{R}^{|\mathcal{A}|}$ to denote each discrete action, where $|\mathcal{A}|$ is the number of total actions in the action set $\mathcal{A}$. In our problem, the three actions are represented as $(1,0,0)^\mathsf{T}, (0,1,0)^\mathsf{T}, (0,0,1)^\mathsf{T}$.

For each action selected by the behavior planning module, we can use the state-of-the-art trajectory planning methods to generate trajectories $\sigma = [\sigma_1, \sigma_2]^\mathsf{T} = [p_x, p_y]^\mathsf{T}$, such as potential fields, cell decomposition, model predictive control (MPC) [58], [60]. We use endogenous transformation to compute the controller's references to track trajectories:

$$u_{ref1} = \frac{\dot{\sigma}_1\ddot{\sigma}_1 + \dot{\sigma}_2\ddot{\sigma}_2}{\sqrt{\dot{\sigma}_1^2 + \dot{\sigma}_2^2}}, \quad u_{ref2} = \frac{\dot{\sigma}_1\ddot{\sigma}_2 - \dot{\sigma}_2\ddot{\sigma}_1}{(\dot{\sigma}_1^2 + \dot{\sigma}_2^2)^{\frac{3}{2}}}. \tag{18}$$

Assembling them in the matrix U as follows for all actions $a \in \mathcal{A}$:

$$U = \begin{bmatrix} u_{1,ref1} & u_{2,ref1} & \cdots & u_{|\mathcal{A}|,ref1} \\ u_{1,ref2} & u_{2,ref2} & \cdots & u_{|\mathcal{A}|,ref2} \end{bmatrix}, \tag{19}$$

where $u_{j,ref1}, u_{j,ref2}$ is the references for j-th action a_j. We can retrieve the references for a_j by multiplying it with U in the objective of the CBF-QP in Eq. (12).

Then we show how we define the CBF for each agent. With respect to the ego vehicle, vehicle fc represents the vehicle immediately in front of the ego vehicle in its *current lane* as shown in Fig. 5. When the action of the ego vehicle is changing lane, vehicle bt represents the vehicle immediately behind the ego vehicle in the *target lane*, vehicle ft denotes the vehicle immediately in front of the ego vehicle in the target lane as shown in Fig. 6. We use $\triangle p_{x,k} = \|p_x - p_{x,k}\|$ to denote the distance between ego vehicle and vehicle $k \in \{fc, ft, bt\}$. Then we define the CBF along the p_x-axis as follows: when the action of the ego vehicle is keeping lane, the CBF is:

$$h_{fc}(x) = \triangle p_{x,fc} - P_{safe}, \tag{20}$$

where P_{safe} is the safe distance between the ego vehicle and other vehicles. When the action of the ego vehicle is changing

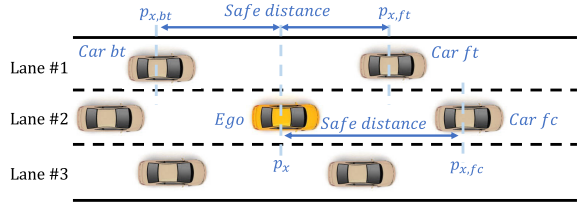


Fig. 6. The ego vehicle is changing from lane #2 to lane #1. Its safety distance is considered on both the current lane (#2) and the target lane (#1). In this figure, the length of each car is ignored for simplicity and the value of safe distance already considers the length of the cars.

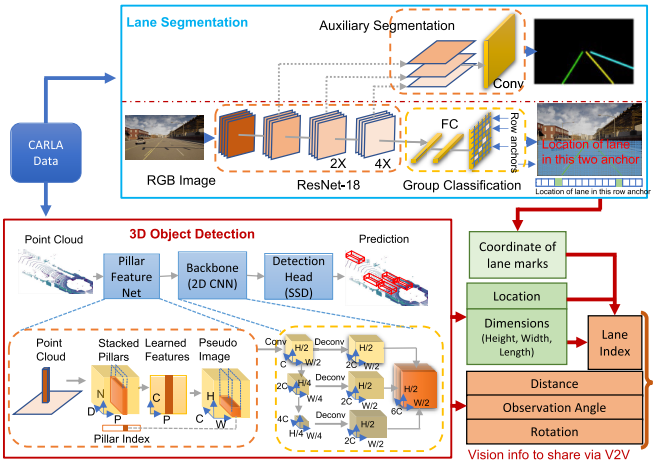


Fig. 7. Detailed structure of vision information processing. We combine the lane segmentation and 3D object detection results to extract neighboring vehicles' features. The processed information is included in the state of each CAV.

lanes, the CBF is:

$$\begin{aligned} h_{fc}(x) &= \triangle p_{x,fc} - P_{safe}, \\ h_{ft}(x) &= \triangle p_{x,ft} - P_{safe}, \\ h_{bt}(x) &= \triangle p_{x,bt} - P_{safe}, \end{aligned} \tag{21}$$

For the p_y-axis, we use CBF to limit the ego vehicle to move within the range $[P_y^{min}, P_y^{max}]$ determined by the width of the lanes:

$$\begin{aligned} h_{y,max}(x) &= -p_y + P_y^{max}, \\ h_{y,min}(x) &= p_y - P_y^{min}. \end{aligned} \tag{22}$$

In the implementation of the CBF-QP controller (12), we use a vectorized CBF. When the action of the ego vehicle is keeping lane, $h(x) = [h_{fc}(x), h_{y,max}(x), h_{y,min}(x)]^\mathsf{T}$. When the action of the ego vehicle is changing lane, $h(x) = [h_{fc}(x), h_{ft}(x), h_{bt}(x), h_{y,max}(x), h_{y,min}(x)]^\mathsf{T}$.

B. CNN-Driven Shared Vision

We develop a CNN-driven shared vision solution to process and share the vision information to support the learning of the truncated Q-function, as shown in Fig. 7. The shared vision model has two main branches. The first one is lane segmentation and the second one is 3D object detection (OD). We use the network architecture from Ultra-Fast-Lane-Detection [61] for lane segmentation. It aggregates auxiliary segmentation tasks to utilize multi-scale features. Table II summarizes the evaluation metrics of several popular CNN backbones for object detection and segmentation that are applied on ImageNet [62], including the total number of parameters, test accuracy, and multiply-accumulate (MAC) operation (accumulate the product of two numbers in a CNN and can evaluate the computational complexity of each network [63]). We adopt ResNet-18 [47] as the backbone since it not only achieves high accuracy but also keeps a small number of parameters and MAC operations. The lightweight feature leads the model easier to be deployed for fast inference in processing environment information.

TABLE II

SPEED AND ACCURACY COMPARISON OF SEVERAL POPULAR CNN BACKBONES. WE ADOPT RESNET-18 BECAUSE IT ACHIEVES HIGH ACCURACY WITH FEWER PARAMETERS FOR FAST INFERENCE

CNN Backbone	No. of Parameters	MACs	Accuracy
AlexNet [64]	61.1M	0.72	56.55
ResNet-18 [47]	11.69M	1.82	71.78
ResNet-50 [47]	25.56M	4.12	76.15
VGG-16 [65]	138.36M	15.5	71.59
GoogLeNet [66]	42.71M	1.52	69.78

In the lane segmentation model, the RGB images from the onboard camera are divided into equal-sized grids, where grids in the same rows are defined as row anchors. Grids with the appearance of the lane mark on this row anchor will be colored green. Detailed pixel coordinates of all four-lane marks of the 3-lane freeway will be generated by the neural network and saved in a hash table.

The 3D OD takes LIDAR point cloud as input and uses the architecture from PointPillars [48] with PointNets [67] as backbone. The inputs are converted to sparse pseudo-images through an encoder, which can learn a set of features from the stacked pillars and then scatter those features back to 2D pseudo-images. A network similar to [68] is used to process the pseudo-image into high-level representation by first continuously down-sampling learned features to small spatial resolution, then up-sampling and concatenating each down-sampled feature. The detection head, single shot multi-box detector (SSD) [69], is used to predict 3D bounding boxes for neighboring vehicles with the features extracted from the backbone. Finally, the vision information (including the location of vehicles in camera coordinates, dimensions, observation angle, distance, and rotation) will be generated.

As the camera RGB image and LIDAR point cloud are paired under the same timestamp, we derive the lane index of each neighboring vehicle by fusing the location and dimension of the neighboring vehicle (obtained by 3D OD) with the coordinate of lane marks obtained by lane segmentation. We then send all vision information including lane index, distance, observation angle, and rotation info to the safe actor-critic algorithm for behavior planning.

For evaluation, PointPillars uses mean average precision (mAP) as standard, while Ultra-Fast-Lane-Detection uses "accuracy": $\frac{\sum_{\text{clip}} C_{\text{clip}}}{\sum_{\text{clip}} S_{\text{clip}}}$, where C_{clip} is the number of lane points predicted correctly and S_{clip} is the total number of ground truth in each clip. We show the performance (accuracy or mAP) and running speed in Table III. The lane segmentation achieves high accuracy with low latency. It can process 313 images

TABLE III
SUMMARY OF RESULTS FROM VISION NETWORK. OUR WEIGHT-PRUNED CNN INCREASES THE SPEED BY 3.5× WITH A VERY SMALL ACCURACY DEGRADATION TO SATISFY THE REQUIREMENT OF INFORMATION SHARING FOR SAFE MARL BEHAVIOR PLANNING

CNN Model	performance (%)	speed (img/s)
Lane Segmentation	accuracy: 95.82	313
3D object detection (OD)	mAP: 76.5	28
3D OD after weight pruning	mAP: 73.6	99

per second. The 3D object detection has a running time of 28 images per second (img/s), which means 100 images can be processed in around 3.6 seconds. Our weight pruning technique significantly increases the speed by 3.5× with a very small accuracy degradation. It decreases the processing time of 100 images from 3.6 seconds to one second. Overall, for parallel lane segmentation and 3D object detection, our weight pruning technique significantly speeds up the CNN-driven shared vision process to achieve real-time processing. Our vision processing method can do lane segmentation and object detection in mixed traffic with both autonomous and human-driven vehicles.

C. System Efficiency Improvement

In this section, we show our algorithm improves the CAVs' system efficiency in terms of the average velocity and the average comfort as defined in the reward function (17).

1) Comparison Under Different CAV Ratios: Our approach improves the average velocity and the average comfort as the CAV ratios (the total CAV number divided by the total number of all vehicles) get higher. In this set of experiments, the total number of CAVs ranges from 0 to 30 as listed in Table IV. We compare the average velocity and comfort for all vehicles under different CAV ratios. The comfort of a single vehicle is defined in Eq. (16). The velocity and comfort are averaged over all the 40000 timesteps and all vehicles in the simulation. The result of our approach is shown at the top of Table IV. All CAVs use our safe actor-critic (Alg. 2) introduced in Section V. The result in Table IV shows the average velocity and comfort of the entire mixed traffic. We use CARLA's built-in human-driven vehicle in the mixed traffic [20]. In the result of Table IV, the average velocity and comfort increase when the CAV ratio gets higher. This gives us insights that the penetration of the CAVs and the communication capabilities can improve traffic efficiency in the future.

The truncated $\mathcal{Q}$-function technique introduced in Section IV is a good approximation of the centralized critic $Q(s, a)$ with the global s and the global a. We compare our approach with the MADDPG [18] plus our safe action mapping technique introduced in Section V. The reason for adding the safe action mapping is that the MADDPG algorithm stops early in each episode due to collisions. We will further explain this in safety experiments. The MADDPG uses the centralized critic $Q(s, a)$. Hence, we use it as a baseline to evaluate our truncated $\mathcal{Q}$-function technique. Compared with the baseline, our average velocity is 6.0% smaller, and the average comfort is 3.9% smaller. Though the system efficiency is sacrificed a little bit, our truncated $\mathcal{Q}$-function increases the scalability since it does not rely on the global states or actions.

TABLE IV
THE SYSTEM EFFICIENCY COMPARISON UNDER DIFFERENT CAV RATIOS. OUR APPROACH IMPROVES THE AVERAGE VELOCITY AND THE AVERAGE COMFORT AS THE CAV RATIOS GET HIGHER

CAV ratio	CAV number	human-driven vehicle number	average velocity (mph)	average comfort
Algorithm	Safe Actor-Critic (Truncated $\mathcal{Q}$ + Safe Action Mapping)			
0	0	30	60.06	2.61
0.17	5	25	61.82	2.64
0.33	10	20	64.70	2.68
0.5	15	15	65.14	2.72
0.67	20	10	65.18	2.74
0.83	25	5	65.53	2.77
1	30	0	66.15	2.81
Algorithm	MADDPG + Safe Action Mapping			
0	0	30	60.06	2.61
0.17	5	25	62.80	2.65
0.33	10	20	66.19	2.72
0.5	15	15	67.71	2.83
0.67	20	10	67.80	2.83
0.83	25	5	69.31	2.85
1	30	0	70.34	2.89

2) Comparison Under Different Traffic Densities: Our approach improves the traffic flow and average comfort under different traffic densities. The traffic density ρ is the ratio between the total number of vehicles and the road length. We compare the traffic flow and average comfort under different traffic densities between our safe MARL approach using Alg. 2 and an intelligent driving model (IDM) [70]. The traffic flow reflects the quality of the road throughout with respect to the traffic density. It is calculated as $\rho \times \bar{v}$, where $\bar{v}$ is the average velocity of all the vehicles [71]. The IDM is a common baseline in autonomous driving. In IDM, the vehicle's acceleration is a function of its current speed, current and desired spacing, and the leading and following vehicles' speed [70]. Building on top of these IDM agents, we add lane-changing functionality using the gap acceptance method in [72]. In this set of experiments, we keep CAV ratios at 0.6. As shown in Fig. 8, the safe MARL agent gets both larger traffic flow and better driving comfort when traffic density ρ is low. When ρ grows, the result of the safe MARL agent gets worse, but it is still comparable with the IDM. When the road is saturated, lane-changing tends to downgrade passengers' comfort but cannot bring higher speed. Consequently, a better choice is to keep lanes when ρ is high, and there is no significant difference between the safe MARL and the IDM. We also add the result using MADDPG in Fig. 8. Both the traffic flow and the driving comfort are very small (a positive number but close to 0) using MADDPG, and many collisions occurred during the simulation. This is because the MADDPG does not have a safety guarantee.

3) Cost-Benefit Analysis of Communication Range: As presented in Table V, we evaluate the learned policy of our safe actor-critic algorithm using various communication ranges under the CAV ratio of 0.9. The total episode reward in the table is averaged over five random episodes for each

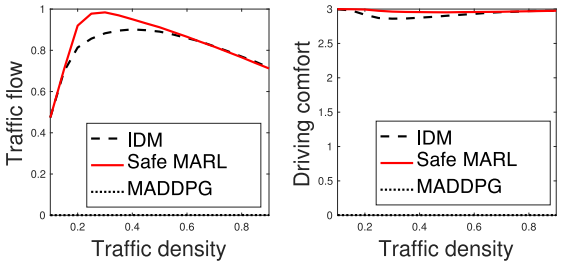


Fig. 8. Our approach (safe MARL) improves the traffic flow and driving comfort under different traffic densities.

TABLE V

THE COST-BENEFIT ANALYSIS OF COMMUNICATION RANGE AND THE SYSTEM'S REWARD IMPROVEMENT

Range (m)	0	50	100	150
Total Episode Reward	1782.93	1847.98	1899.24	1937.24
Range (m)	200	250	300	350
Total Episode Reward	1948.23	1954.63	1970.98	1978.23

TABLE VI

TOTAL NUMBER OF UNSAFE ACTIONS EXECUTED BY THE SAFE MARL AND MADDPG UNDER DIFFERENT TRAFFIC DENSITY ρ. OUR APPROACH HAS 0 UNSAFE ACTION BY USING THE SAFE ACTION MAPPING

ρ	0.1	0.2	0.3	0.4	0.5
Safe MARL	0	0	0	0	0
MADDPG	4416	9510	21897	43491	61689
ρ	0.6	0.7	0.8	0.9	
Safe MARL	0	0	0	0	
MADDPG	91154	133135	191404	242976	

policy. Each episode has a maximum timestep of 40000. As the communication range increases, the vehicles can obtain more information from their neighboring vehicles with higher communication costs. As a result, we can observe that more communication costs can lead to further enhancements and more benefits in the traffic system's rewards.

D. Safety Guarantee

In this section, we show our algorithm has a safety guarantee. We compare our safe actor-critic algorithm with the MADDPG [18] algorithm. We assign a negative reward in MADDPG for each collision. Our approach avoids the execution of unsafe actions that can lead to collisions. Table VI shows the total number of unsafe actions executed by the safe MARL and MADDPG under different traffic densities. The traffic density ρ is the ratio between the total number of vehicles and the road length. The number in Table VI is averaged over the last 10 episodes, which has a maximum timestep of 40000. When $\rho = 0.9$, our approach has 0 unsafe actions while the MADDPG has 242976 unsafe actions because it does not have a safety module.

Our approach can maintain a safe headway with neighboring vehicles while the MADDPG cannot. The headway is the distance between two consecutive vehicles following each other. In Fig. 9, the minimum headway across all the vehicles is shown in the first 500 timesteps of one episode when

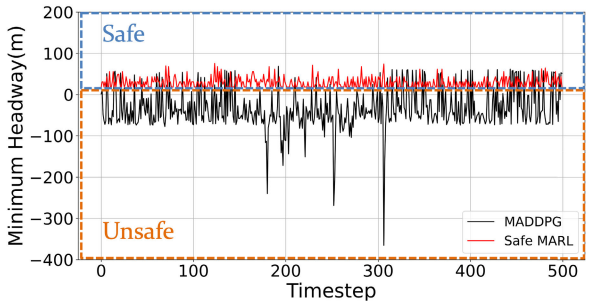


Fig. 9. Our approach (Safe MARL) can maintain a safe headway with neighboring vehicles while the MADDPG cannot. The figure shows the minimum headway across all the vehicles in the first 500 timesteps of one episode when $\rho = 0.6$.

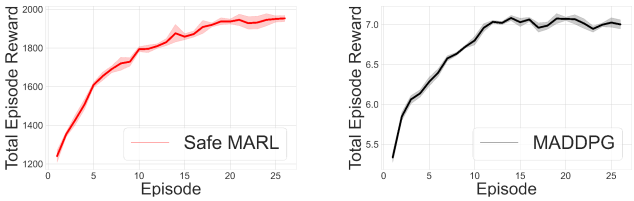


Fig. 10. Our approach (safe MARL) gets a higher total episode reward compared to the MADDPG during the training process, because our approach can guarantee a safe training process and run for the maximum episode length.

$\rho = 0.6$. We see that the headway is always greater than 0 using our safe actor-critic algorithm with a minimum value of 18.5 meters. Nevertheless, the MADDPG is likely to have a negative headway, which means collisions in reality. Note that we set the minimum car-following distance of CAVs to be 18.5 meters following the study of the safe car-following distance of autonomous vehicles [73], but this value can be set differently to satisfy the requirements in different scenarios.

Our approach gets a much larger total episode reward than the MADDPG. The total episode reward is the summation of all stage-wise rewards defined in Eq. 17 for each episode. As shown on the left of Fig. 10, the maximum total episode reward using our approach is about 1940, which is first reached in the 20th episode. In the right figure, the maximum total episode reward using the MADDPG is about 7, because some collision terminates the episode. They have different initial rewards because the neural networks are randomly initialized and the action is selected by the ϵ-greedy method with randomness. Our approach runs about 30 minutes in each episode as it has a safety guarantee and runs for the maximum episode length. Yet, each episode stops quickly in about 5 seconds using the MADDPG.

E. Obstacle-At-Corner Scenario and Benefit of Shared Vision

We construct a scenario called obstacle-at-corner to show how sharing vision information can help autonomous vehicles make wise lane-changing decisions ahead of time. As shown in Fig. 11, there are obstacles at a left-turning corner (represented by two stationary vehicles). The right bottom figure shows the view of a vehicle that comes in the direction of this curved road. It is quite difficult to observe the obstacles merely relying on its own sensors. In this case, if the white front vehicle can share its observation, the coming vehicle can get to know there are obstacles before entering the turning corner.

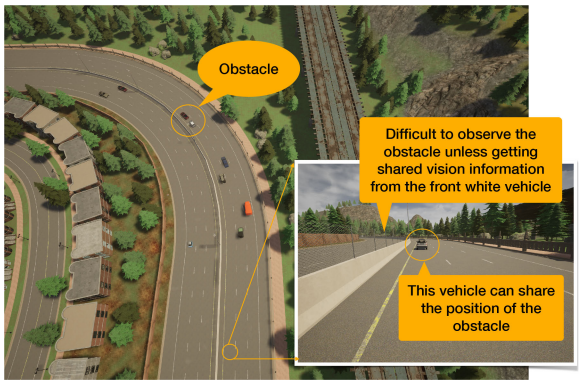


Fig. 11. The obstacle-at-corner scenario where there are obstacles in a left-turning corner. The vehicles on the road are autonomous vehicles. The coming vehicles' view is blocked such that they cannot observe the obstacles.

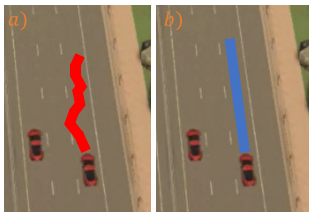


Fig. 12. The trajectory is smooth when we separate the learning and control modules in (b), but the trajectory is zigzag when we use the steering angle and acceleration as the action space of our safe actor-critic algorithm in (a).

Separating the learning and control modules in our approach is necessary. One alternative design is to output the control inputs (steering angle and acceleration) directly [11], [12], [23], [74]. But it is not suitable for our CAV problem considering the lane-changing behavior. As the subfigure (a) shown in Fig. 12, we use the same truncated Q-network structure in Fig. 3, but the action space of both the critic and the actor networks is replaced by steering angle and acceleration. The trajectory is zigzag because the learned policy does not have a concept of changing/keeping lanes. The learned policy directly outputs the steering angles and accelerations to maximize the vehicle's rewards. Since the steering angle and acceleration are selected by the MARL agents, there is no need to use a controller in subfigure (a). In subfigure (b), the trajectory is smooth when we first find whether to change/keep lanes with the policy learned by Alg. 2 and then implement this action by a CBF-QP controller in Eq. (12).

Shared vision with our proposed safe MARL algorithm can help CAVs avoid traffic jams. We test our learned policy in the obstacle-at-corner scenario. When no vehicle can share information in subfigure (a) of Fig. 13, there are more vehicles blocked on the left-most lane causing a traffic jam. Subfigures (b) and (c) are the screenshots taken in two close timesteps. In subfigure (b), vehicle A is blocked in the left lane because there is vehicle B in the middle lane and A does not realize there are obstacles before it enters the corner (there is no neighboring vehicle that can share the obstacle information in advance in this scene; two obstacles are not CAVs). In subfigure (c), the coming vehicle C in the left-most lane starts to change to the middle lane before it observes the obstacles because it gets the shared vision information either from A or B. We use the safe action mapping introduced in Sec. V to ensure the safety of this lane-changing.

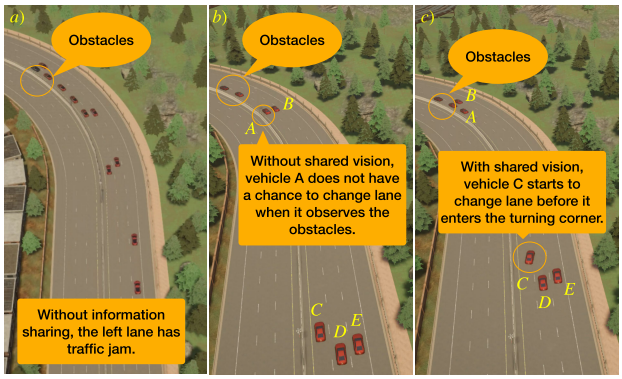


Fig. 13. Without any information sharing, there is a traffic jam in the left lane. With shared vision from A or B, using our safe MARL policy, vehicle C can change its lane before it enters the left-turning corner.

VII. Conclusion

How to utilize V2V communication to improve traffic efficiency while satisfying safety requirements is a challenging problem for the behavior planning and control of CAVs. We design a safe behavior learning and control framework, which utilizes shared states and actions to safely explore a behavior planning policy. We design two new techniques: truncated Q-function and the safe action mapping. The truncated Q-function utilizes the information-sharing capability of the CAVs instead of depending on the global states and actions, and we prove that the approximation error is bounded. The safe action mapping guarantees the safety of the training and execution process of the proposed MARL framework. In experiments, our weighted-pruned CNN technique increases the speed of the 3D object detection (OD) by 3.5× with small accuracy degradation to support MARL. We also show that our approach improves the average velocity and average comfort under different CAV ratios and different traffic densities. Our approach avoids unsafe actions and maintains a safe distance from neighboring vehicles. We also construct the obstacle-at-corner scenario to show that the shared vision can help vehicles avoid traffic jams. It is considered future work to improve the robustness of the MARL policy under state uncertainties caused by communication or sensor measurement errors.

Appendix A
Physical Dynamic Model

The physical dynamics of a vehicle are described by a kinematic bicycle model that achieves a good balance between accuracy and complexity [75], [76]. The kinematic bicycle model is a widely-used model in literature [26], [58], [77] for the MPC-based and CBF-based controller design. The discrete-time equations are obtained by applying an explicit Euler method with a sampling time $T_s = 0.01s$:

$$x_{t+1} = \underbrace{\begin{bmatrix} x_1(t) + x_4(t)\cos(x_3(t))T_s \\ x_2(t) + x_4(t)\sin(x_3(t))T_s \\ x_3(t) \\ x_4(t) \end{bmatrix}}_{f(x_t)} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{x_4(t)T_s}{L} & 0 \\ 0 & T_s \end{bmatrix}}_{g(x_t)} u_t + w_t, \tag{23}$$

where $x = [p_x, p_y, \psi, v]^\mathsf{T} \in \mathcal{X}$ includes the two coordinates of the center of gravity (CoG), its orientation and velocity. The input $u = [\tan(\delta), \dot{v}] \in \mathcal{U}$ is the tangent of the steering angle and acceleration. The parameter $L = 2.51\ m$ is the distance between the front and rear axles.

Appendix B
Exponential Decay Property

As defined in Definition 1, the (c, ρ)-exponential decay property for Q^i-function means there exists some $c > 0$ and $0 < \rho < 1$ such that for any $i \in \mathcal{N}, \forall s^{N_\kappa^i} \in \mathcal{S}^{N_\kappa^i}$, $\forall s^{N_\kappa^{-i}} \in \mathcal{S}^{N_\kappa^{-i}}, \forall s'^{N_\kappa^{-i}} \in \mathcal{S}^{N_\kappa^{-i}}, \forall a^{N_\kappa^i} \in \mathcal{A}^{N_\kappa^i}, \forall a^{N_\kappa^{-i}} \in \mathcal{A}^{N_\kappa^{-i}}$, $\forall a'^{N_\kappa^{-i}} \in \mathcal{A}^{N_\kappa^{-i}}$, it holds that $|Q^i(s^{N_\kappa^i}, s^{N_\kappa^{-i}}, a^{N_\kappa^i}, a^{N_\kappa^{-i}}) - Q^i(s^{N_\kappa^i}, s'^{N_\kappa^{-i}}, a^{N_\kappa^i}, a'^{N_\kappa^{-i}})| \le c\rho^\kappa$.

Here we prove the following theorem that gives the condition when the exponential decay property holds.

Theorem 4: If for all $i \in \mathcal{N}$, r^i is upper bounded by $\bar{r}$, then the $(\frac{\bar{r}}{1-\gamma}, \sqrt{\gamma})$-exponential decay property holds for Q^i in (1) under Assumption 1, where γ is the discount factor.

Proof: Denote $s = (s^{N_\kappa^i}, s^{N_\kappa^{-i}})$, $a = (a^{N_\kappa^i}, a^{N_\kappa^{-i}})$, $s' = (s^{N_\kappa^i}, s'^{N_\kappa^{-i}})$, $a' = (a^{N_\kappa^i}, a'^{N_\kappa^{-i}})$. We use p_k^i to represent the distribution of the state-action pair (s_k^i, a_k^i) under some localized policy $\pi^i(a^i|s^{N^i})$ for vehicle i with the initial condition $(s_0, a_0) = (s, a)$. Here the policy π^i can be any feasible policy that is not required to be the optimal policy. We use $p_k'^i$ to represent the distribution of the state-action pair (s_k^i, a_k^i) under the same policy π^i for vehicle i with the initial condition $(s_0, a_0) = (s', a')$. Because each vehicle i's next state s_{k+1}^i is independently generated and only depends on its 1-hop neighbors, that is to say,

$$P(s_{k+1}|s_k, a_k) = \prod_{i=1}^{n} P(s_{k+1}^i | s_k^{N^i}, a_k^{N^i}), \tag{24}$$

and the localized policy π^i only depends on s^{N^i}, we have $p_k^i = p_k'^i$ for $k \le \lfloor \kappa/2 \rfloor$.

Based on the definition of the action-value function, we have

$$\begin{aligned}
&\left|Q^i(s,a) - Q^i(s',a')\right| \\
&\le \sum_{k=0}^{\infty} \Big| \mathbb{E}_{a_k^i \sim \pi(a^i|s^{N^i})} \left[\gamma^k r_{k+1}^i(s_k^i, a_k^i) \mid s_o = s, a_0 = a\right] \\
&\quad - \mathbb{E}_{a_k^i \sim \pi(a^i|s^{N^i})} \left[\gamma^k r_{k+1}^i(s_k^i, a_k^i) \mid s_o = s', a_0 = a'\right] \Big| \\
&= \sum_{k=0}^{\infty} \big|\gamma^k \mathbb{E}_{(s^i,a^i)\sim p_k^i} \left[r_{k+1}^i(s_k^i, a_k^i)\right] \\
&\quad - \gamma^k \mathbb{E}_{(s^i,a^i)\sim p_k'^i} \left[r_{k+1}^i(s_k^i, a_k^i)\right] \big| \\
&= \sum_{k=\lfloor \kappa/2 \rfloor + 1}^{\infty} \left|\gamma^k \mathbb{E}_{p_k^i} \left[r_{k+1}^i(s_k^i, a_k^i)\right] - \gamma^k \mathbb{E}_{p_k'^i} \left[r_{k+1}^i(s_k^i, a_k^i)\right]\right| \\
&\le \sum_{k=\lfloor \kappa/2 \rfloor + 1}^{\infty} \gamma^k \bar{r} \times \mathrm{TV}(p_k^i, p_k'^i) \\
&\le \frac{\bar{r}}{1-\gamma} \gamma^{\lfloor \kappa/2 \rfloor + 1} \le \frac{\bar{r}}{1-\gamma} \gamma^{\kappa/2},
\end{aligned} \tag{25}$$

where $\mathrm{TV}(p_k^i, p_k'^i)$ is the total variation distance between p_k^i and $p_k'^i$ and upper bounded by 1 based on the definition of total variation [78]. This shows the $(\frac{\bar{r}}{1-\gamma}, \sqrt{\gamma})$-exponential decay property holds for Q^i according to the definition of the exponential decay property. ■

Appendix C
CBF-Based Quadratic Programming

In this section, we show how we can formulate a Quadratic Programming (QP) based on the CBF condition (10) to guarantee the safety of a system without noise defined as follows:

$$x_{t+1} = f(x_t) + g(x_t)u_t, \tag{26}$$

The CBF-QP for this system that can be solved efficiently at each timestep is formulated as:

$$\begin{aligned}
&\underset{u_t}{\operatorname{argmin}} \|u_t - U \cdot a\|^2 \\
&\text{s.t.} \quad p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t) u_t + q \ge (1-\eta) h(x_t).
\end{aligned} \tag{27}$$

Lemma 1: For system (26), if there exists $\eta \in (0, 1]$ such that the QP (27) has a solution for all $x_t \in \mathcal{C}$ ($\mathcal{C}$ is defined in (11)), then the controller derived from (27) renders set $\mathcal{C}$ forward invariant.

Proof: For any $x_t \in \mathcal{C}$, we have $h(x_t) \ge 0$ according to the Definition 4. Therefore,

$$\begin{aligned}
h(x_{t+1}) &= p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t) u_t + q \\
&\ge (1-\eta) h(x_t) \ge 0.
\end{aligned} \tag{28}$$

Thus, $x_{t+1} \in \mathcal{C}$ and $\mathcal{C}$ is forward invariant. ■

We can relax the constraint in (27), and the forward invariant property holds for a larger set. Consider:

$$\begin{aligned}
&\underset{u_t, \zeta}{\operatorname{argmin}} \|u_t - U \cdot a\|^2 + M\zeta \\
&\text{s.t.} \quad p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t) u_t + q \ge (1-\eta) h(x_t) - \zeta \\
&\quad \zeta \ge 0,
\end{aligned} \tag{29}$$

where M is a large constant.

Theorem 5: For system (26), if there exists $\eta \in (0, 1]$ such that the QP (29) has a solution for all $x_t \in \mathcal{C}$ and the solution satisfies $\zeta \le Z$, then the controller derived from (29) renders set $\mathcal{C}' : \{x \in \mathbb{R}^{n_x} \mid h'(x) = h(x) + \frac{Z}{\eta} \ge 0\}$ forward invariant.

Proof: Since $x_t \in \mathcal{C}$, we have $h(x_t) \ge 0$ according to the Definition 4. From $Z \ge \zeta \ge 0 \ge -\eta h(x_t)$, we have $h(x_t) + \frac{Z}{\eta} \ge 0$. Thus, $x_t \in \mathcal{C}'$ with $h'(x_t) \ge 0$. Also, we have

$$\begin{aligned}
h(x_{t+1}) &= p^\mathsf{T} f(x_t) + p^\mathsf{T} g(x_t) u_t + q \\
&\ge (1-\eta) h(x_t) - \zeta \ge (1-\eta) h(x_t) - Z,
\end{aligned} \tag{30}$$

$$\begin{aligned}
h(x_{t+1}) + \frac{Z}{\eta} &\ge (1-\eta) \left[h(x_t) + \frac{Z}{\eta}\right], \\
h'(x_{t+1}) &\ge (1-\eta) h'(x_t) \ge 0.
\end{aligned} \tag{31}$$

Thus, $x_{t+1} \in \mathcal{C}'$ and $\mathcal{C}'$ is forward invariant. Note that it is simplified to be Lemma 1 with set $\mathcal{C} = \mathcal{C}'$ when $Z = 0$. ■

The value of Z denotes how large the CBF condition (10) is violated from the original $h(x_t)$. In this case, the safety condition should be formulated according to the set $\mathcal{C}'$.

REFERENCES

[1] D. Martín-Sacristán et al., "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, Feb. 2021.

[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.*, vol. 23, no. 9, pp. 14439–14455, Sep. 2022.

[3] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[4] M. Won, "L-Platooning: A protocol for managing a long platoon with DSRC," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5777–5790, Jun. 2022.

[5] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 433–443, Jan. 2020.

[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] B. Peng, M. F. Keskin, B. Kulcsár, and H. Wymeersch, "Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning," *Commun. Transp. Res.*, vol. 1, Dec. 2021, Art. no. 100017.

[8] Z. E. A. Kherroubi, S. Aknine, and R. Bacha, "Novel decision-making strategy for connected and autonomous vehicles in highway on-ramp merging," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12490–12502, Aug. 2022.

[9] F. Farivar, M. S. Haghighi, A. Jolfaei, and S. Wen, "On the security of networked control systems in smart vehicle and its adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3824–3831, Jun. 2021.

[10] M. A. Silgu, I. G. Erdagi, G. Göksu, and H. B. Celikoglu, "Combined control of freeway traffic involving cooperative adaptive cruise controlled and human driven vehicles using feedback control through SUMO," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11011–11025, Aug. 2022.

[11] L. Chen, X. Hu, B. Tang, and Y. Cheng, "Conditional DQN-based motion planning with fuzzy logic for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 2966–2977, Apr. 2022.

[12] J. Chen, S. E. Li, and M. Tomizuka, "Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5068–5078, Jun. 2022.

[13] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4909–4926, Jun. 2022.

[14] Y. Fu, C. Li, F. R. Yu, T. H. Luan, and Y. Zhang, "Hybrid autonomous driving guidance strategy combining deep reinforcement learning and expert system," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11273–11286, Aug. 2022.

[15] L. Zhu, L. Lu, X. Wang, C. Jiang, and N. Ye, "Operational characteristics of mixed-autonomy traffic flow on the freeway with on- and off-ramps and weaving sections: An RL-based approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13512–13525, Aug. 2022.

[16] A. R. Kreidieh, C. Wu, and A. M. Bayen, "Dissipating stop-and-go waves in closed and open networks via deep reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1475–1480.

[17] Z. Yan and C. Wu, "Reinforcement learning for mixed autonomy intersections," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 2089–2094.

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

[19] K. Muhammad, A. Ullah, J. Lloret, J. D. Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4316–4336, Jul. 2021.

[20] A. Dosovitskiy and G. Ros, "CARLA: An open urban driving simulator," 2017, *arXiv:1711.03938*.

[21] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14128–14147, Sep. 2022.

[22] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3387–3395.

[23] M. Bojarski et al., "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.

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

[25] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Feb. 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 *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 178–185.

[27] 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 *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8765–8771.

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

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

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

[31] P. Sunehag and G. Lever, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. AAMAS*, 2018, pp. 2085–2087.

[32] C. Yu et al., "Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 735–748, Feb. 2020.

[33] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, "Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. NeurIPS*, Dec. 2020, pp. 1–12.

[34] M. Zhou, Z. Liu, P. Sui, Y. Li, and Y. Y. Chung, "Learning implicit credit assignment for cooperative multi-agent reinforcement learning," in *Proc. NeurIPS*, vol. 33, 2020, pp. 11853–11864.

[35] O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.

[36] C. D. Hsu, H. Jeong, G. J. Pappas, and P. Chaudhari, "Scalable reinforcement learning policies for multi-agent control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 256–266.

[37] O. Bastani, Y. Pu, and A. Solar-Lezama, "Verifiable reinforcement learning via policy extraction," in *Proc. NeurIPS*, 2018, pp. 1–11.

[38] G. Thomas, Y. Luo, and T. Ma, "Safe reinforcement learning by imagining the near future," in *Proc. NeurIPS*, vol. 34, 2021, pp. 13859–13869.

[39] S. Mo, X. Pei, and C. Wu, "Safe reinforcement learning for autonomous vehicle using Monte Carlo tree search," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6766–6773, Jul. 2022.

[40] R. Basso, B. Kulcsár, I. Sanchez-Diaz, and X. Qu, "Dynamic stochastic electric vehicle routing with safe reinforcement learning," *Transp. Res. E, Logistics Transp. Rev.*, vol. 157, Jan. 2022, Art. no. 102496.

[41] S. Lu, K. Zhang, T. Chen, T. Basar, and L. Horesh, "Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 10, pp. 8767–8775.

[42] S. Li and O. Bastani, "Robust model predictive shielding for safe reinforcement learning with stochastic dynamics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 7166–7172.

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

[44] A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 683–711, Feb. 2021.

[45] S. Berlato, M. Centenaro, and S. Ranise, "Smart card-based identity management protocols for V2V and V2I communications in CCAM: A systematic literature review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10086–10103, Aug. 2022.

[46] G. Notomista, M. Wang, M. Schwager, and M. Egerstedt, "Enhancing game-theoretic autonomous car racing using control barrier functions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 5393–5399.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[48] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.

[49] A. Miller, K. Rim, P. Chopra, P. Kelkar, and M. Likhachev, "Cooperative perception and localization for cooperative driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1256–1262.

[50] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50 × fewer parameters and < 0.5 MB model size," 2016, *arXiv:1602.07360*.

[51] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A filter level pruning method for deep neural network compression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5068–5076.

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (Adaptive Computation and Machine Learning Series). Cambridge, MA, USA: MIT Press, 2017, pp. 321–359.

[53] T. Zhang et al., "A systematic DNN weight pruning framework using alternating direction method of multipliers," in *Proc. ECCV*, 2018, pp. 184–199.

[54] F. Khelladi, M. Boudali, R. Orjuela, M. Cassaro, M. Basset, and C. Roos, "An emergency hierarchical guidance control strategy for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4319–4330, May 2022.

[55] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[56] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation," in *Proc. Robot., Sci. Syst.*, Cambridge, MA, USA, Jul. 2017.

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

[58] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 23–35, Fall. 2017.

[59] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous driving on curvy roads without reliance on Frenet frame: A Cartesian-based trajectory planning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15729–15741, Sep. 2022.

[60] S. Dixit et al., "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annu. Rev. Control*, vol. 45, pp. 76–86, Jan. 2018.

[61] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," 2020, *arXiv:2004.11757*.

[62] A. Benali Amjoud and M. Amrouch, "Convolutional neural networks backbones for object detection," in *Image and Signal Processing*, Cham, Switzerland: Springer, 2020.

[63] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[64] A. Krizhevsky and I. Sutskever, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, vol. 25, 2012, pp. 1097–1105.

[65] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[66] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[67] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[68] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[69] W. Liu et al., "SSD: Single shot MultiBox detector," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 21–37.

[70] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transp. Res. C, Emerg. Technol.*, vol. 71, pp. 143–163, Oct. 2016.

[71] J. Rios-Torres and A. A. Malikopoulos, "Impact of partial penetrations of connected and automated vehicles on fuel consumption and traffic flow," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 4, pp. 453–462, Dec. 2018.

[72] V. A. Butakov and P. Ioannou, "Personalized driver/vehicle lane change models for ADAS," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4422–4431, Oct. 2015.

[73] N. Aréchiga, "Specifying safety of autonomous vehicles in signal temporal logic," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 58–63.

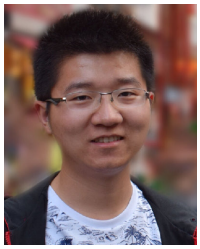
[74] A. Kendall et al., "Learning to drive in a day," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8248–8254.

[75] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4459–4466, Oct. 2019.

[76] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer, 2011.

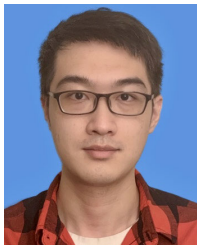
[77] Z. Lin et al., "Policy iteration based approximate dynamic programming toward autonomous driving in constrained dynamic environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5003–5013, May 2023.

[78] G. Grimmett and D. Stirzaker, *Probability and Random Processes*. Oxford, U.K.: Oxford Univ. Press, 2020.



Songyang Han (Student Member, IEEE) received the B.E. degree in automation from Nanjing University, Nanjing, China, in 2015, the M.S. degree in electrical and computer engineering from Shanghai Jiao Tong University, Shanghai, China, in 2018, and the Ph.D. degree in computer science and engineering from the University of Connecticut, Storrs, CT, USA, in 2023. He is currently a Research Scientist with Sony AI. His current research interests include reinforcement learning, artificial intelligence, and generative AI. He was a recipient of the Best Paper Award at the 12th ACM/IEEE International Conference on Cyber-Physical Systems.

Shanglin Zhou received the B.S. degree in statistics from the Minzu University of China, Beijing, China, in 2015, and the M.S. degree in statistics from the University of Connecticut, Storrs, CT, USA, in 2017, where she is currently pursuing the Ph.D. degree in computer science and engineering. Her current research interests include deep learning model compression and the application of computer vision.

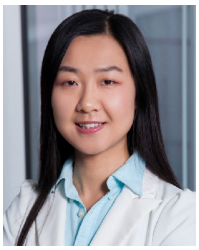


Jiangwei Wang (Student Member, IEEE) received the B.S. degree in electrical and computer engineering from Xi'an Jiaotong University, Xi'an, China, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Connecticut, Storrs, CT, USA. His current research interests are in cyber-physical systems and autonomous driving. He was a recipient of the Outstanding Reviewer for IEEE TRANSACTIONS ON SUSTAINABLE ENERGY and IEEE JOURNAL OF OCEANIC ENGINEERING in 2019.

Lynn Pepin received the B.E. degree in computer science and engineering from the University of Connecticut, Storrs, CT, USA, in 2018, where he is currently pursuing the Ph.D. degree in computer science. His current research interests include cybersecurity and machine learning research in the area of power systems and autonomous vehicles.

Caiwen Ding (Member, IEEE) received the Ph.D. degree in computer science and engineering from Northeastern University, Boston, MA, USA, in 2019. He is currently an Assistant Professor with the School of Computing, University of Connecticut, Storrs, CT, USA. His research interests include machine learning, deep neural network systems, computer vision, and natural language processing. He was a recipient of the Best Paper Award Nomination at DATE 2018 and DATE 2021.

Jie Fu (Member, IEEE) received the M.S. degree in electrical engineering and automation from the Beijing Institute of Technology, Beijing, China, in 2009, and the Ph.D. degree in mechanical engineering from the University of Delaware, Newark, DE, USA, in 2013. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. Previously, she was a Post-Doctoral Researcher with the University of Pennsylvania. Her research interests include control theory, formal methods, and machine learning, with applications to robotic systems and cyber-physical systems.



Fei Miao (Member, IEEE) received the B.S. degree in automation from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the M.A. degree in statistics and the Ph.D. degree in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2015 and 2016, respectively. She is currently an Associate Professor with the School of Computing, University of Connecticut, Storrs, CT, USA. Previously, she was a Post-Doctoral Researcher with the University of Pennsylvania. Her research interests include learning and control of cyber-physical systems under model uncertainties and CPS security. She was a recipient of the NSF CAREER Award, the Best Paper Award at the 12th ACM/IEEE International Conference on Cyber-Physical Systems, and the Best Doctoral Dissertation Award during the Ph.D. study.