

Robust and Safe Multi-Agent Reinforcement Learning with Communication for Autonomous Vehicles: From Simulation to Hardware

Keshawn Smith

Department of ECE
University of Connecticut
keshawn.smith@uconn.edu

Zhili Zhang

School of Computing
University of Connecticut
zhili.zhang@uconn.edu

H M Sabbir Ahmad

Department of ECE
Boston University
sabbir92@bu.edu

Ehsan Sabouni

Department of ECE
Boston University
esabouni@bu.edu

Maniak Mondal

School of Computing
University of Connecticut
mainak.mondal@uconn.edu

Song Han

School of Computing
University of Connecticut
song.han@uconn.edu

Wenchao Li

Department of ECE
Boston University
wenchao@bu.edu

Fei Miao

School of Computing
University of Connecticut
fei.miao@uconn.edu

Abstract: Deep multi-agent reinforcement learning (MARL) has been demonstrated effectively in simulations for multi-robot problems. For autonomous vehicles, the development of vehicle-to-vehicle (V2V) communication technologies provide opportunities to further enhance system safety. However, zero-shot transfer of simulator-trained MARL policies to dynamic hardware systems remains challenging, and how to leverage communication and shared information for MARL has limited demonstrations on hardware. This problem is challenged by discrepancies between simulated and physical states, system state and model uncertainties, practical shared information design, and the need for safety guarantees in both simulation and hardware. This paper designs RSR-RSMARL, a novel Robust and Safe MARL framework that supports Real-Sim-Real (RSR) policy adaptation for multi-agent systems with communication among agents, with both simulation and hardware demonstrations. RSR-RSMARL leverages state (includes shared state information among agents) and action representations considering real system complexities for MARL formulation. The MARL policy is trained with robust MARL algorithm to enable zero-shot transfer to hardware considering the sim-to-real gap. A safety shield module using Control Barrier Functions (CBFs) provides safety guarantee for each individual agent. Experimental results on 1/10th-scale autonomous vehicles with V2V communication demonstrate the ability of RSR-RSMARL framework to enhance driving safety and coordination across multiple configurations. These findings emphasize the importance of jointly designing robust policy representations and modular safety architectures to enable scalable, generalizable RSR transfer in multi-agent autonomy.

Keywords: Robust Multi-Agent Reinforcement Learning, Safe Multi-Agent Reinforcement Learning, MARL with Communication

1 Introduction

The U.S. Department of Transportation (USDOT) has recently outlined a national strategy to deploy Vehicle-to-Everything (V2X) technologies on U.S. roadways, with the goal of reducing traffic-related fatalities and improving overall transportation safety [1]. As the potential of Connected and Autonomous Vehicles (CAVs) continues to attract interest, a growing body of research has explored

their applications and societal benefits [2, 3, 4]. Multi-Agent Reinforcement Learning (MARL) has emerged as a promising paradigm for decision-making in autonomous driving, demonstrating the ability to learn cooperative and adaptive strategies in dynamic traffic environments [2, 5, 6, 7, 8]. By leveraging inter-agent communication, MARL-based CAV coordination strategies can enhance road safety, mitigate congestion, and improve efficiency. Despite these advances, no open-source testbed currently supports the full development and evaluation of MARL methods for CAVs. Existing testbeds are often closed-source, incomplete, or lack the necessary components to validate end-to-end multi-agent frameworks [9, 3, 10, 11]. This gap significantly limits progress toward validating robust and safe MARL algorithms in realistic CAV contexts.

Deep reinforcement learning has also been widely applied in robotics [11], yet the sim-to-real gap remains a fundamental challenge. Policies trained in simulation often degrade in performance when deployed in real-world systems due to unmodeled noise, uncertainties, and diverse operating scenarios. Domain randomization and related techniques [12, 13] can mitigate this discrepancy but at the cost of substantially higher computational requirements. For multi-agent CAV systems, these challenges are compounded by the critical importance of safety: unsafe actions may cause irreversible failures in densely populated environments. Robustness is further challenged by state estimation errors, communication delays, and model uncertainties. Therefore, frameworks for MARL in CAVs must incorporate safety guarantees not only during training but also throughout deployment in real-world environments.

In this work, we present a novel **Real-Sim-Real** Multi-Agent Reinforcement Learning (RSR-MARL) framework that explicitly addresses these challenges. Our approach minimizes the sim-to-real gap by designing MARL problem formulations and training algorithms that closely align with the physical systems in which they are deployed. To ensure safety, we integrate a Control Barrier Function (CBF)-based *Safety Shield*. High-level decision-making tasks such as lane-keeping and lane-changing are handled by robust MARL policies [14, 15], while low-level safe control actions are enforced through the safety shield. The shield dynamically filters unsafe actions, guaranteeing adherence to safety constraints during both training and execution. This modular design allows the *Safety Shield* framework to adapt seamlessly across different dynamical systems (such as those governed by PID or MPC) without requiring fundamental changes to the training pipeline. Furthermore, we incorporate communication delays directly into the training process, simulating realistic shared observation latencies between neighboring vehicles. This explicit modeling and robust MARL training process ensures that the resulting policies are zero-shot transferable to real-world platforms.

The key contributions of this work are summarized as follows:

1. We propose **RSR-RSMARL**, a robust and safe MARL framework with inter-agent communication, specifically designed for **Real-Sim-Real** transfer of MARL-based policies. Our framework explicitly addresses challenges such as model uncertainties, state estimation errors, and communication delays while enforcing safety guarantees.
2. We conduct extensive experiments both in the CARLA simulator and on real-world 1/10th-scale autonomous vehicles. In simulation, we evaluate MARL policies under controlled conditions using safety and efficiency metrics, while in hardware, we validate zero-shot policy transfer under diverse driving scenarios and ablation settings.
3. To the best of our knowledge, this is the first demonstration of robust and safe MARL with communication on physical CAV platforms. Our real-world experiments highlight the framework’s ability to generalize simulation-trained policies to hardware while ensuring safe and adaptive operations.

Our results demonstrate that the proposed **RSR-RSMARL** pipeline provides an effective and reliable pathway for safely transferring MARL-based decision-making strategies from simulation to physical systems. This work represents a step toward enabling practical deployment of multi-agent CAV coordination in real-world environments.

2 Related Work

In this section, we review the existing literature in this area, highlighting its limitations to establish the motivation for our proposed approach.

Deep Reinforcement Learning in Robotics Training RL or MARL policies in simulation ensures safety and efficiency by mitigating risks to hardware and its surroundings. While imitation learning is commonly used for policy transfer [16, 17], its reliance on real-world data often incurs significant costs. Our approach focuses on simulator-based training to reduce this dependency while maintaining robust real-world performance. Addressing the challenges of sim-to-real transfer, prior studies have introduced techniques such as domain randomization, state normalization, and noise injection to bridge the sim-to-real gap [18, 19, 20]. Building on these advancements, our proposed RSR-RSMARL framework aligns simulator and real-world environments by designing state and action spaces based on hardware capabilities, enabling efficient policy deployment for execution.

Multi-Agent Systems and CAV Testbeds Existing multi-agent system and CAV vehicular testbeds [9, 3, 10, 11, 21] address diverse research areas such as planning and control, computer vision, collective behavior, autonomous racing, and human-computer interaction. For instance, Blumenkamp et al. [21] introduced the Cambridge RoboMaster platform, which leverages customized DJI RoboMaster S1 robots with a tightly integrated hardware, control, and simulation stack. While effective, their approach requires a bespoke simulation environment tailored specifically to their robot platform in order to train MARL policies, limiting portability to other systems. Moreover, their framework does not incorporate explicit safety filtering during policy execution, whereas our work introduces a CBF-based Safety Shield (via QP formulations) and CBF/CLF-constrained MPC backend to enforce real-time safety guarantees. By contrast, our proposed testbed provides a fully open-source and extensible framework, supports Real-Sim-Real transfer without reliance on robot-specific simulators, and integrates robust MARL with modular safety mechanisms to ensure reliable multi-agent autonomy across diverse scenarios.

Safe and Robust RL and MARL Safety has become a critical focus in RL and MARL, with prior work exploring safety shields, barrier functions, and CBF-PID controllers [22, 23, 24, 2, 25, 8], as well as robust RL methods for uncertain observations [15, 14, 26]. However, these approaches often overlook the combined challenges of communication latency, sensing uncertainty, and explicit safety guarantees in multi-agent deployment. Our work advances this area by introducing **RSR-RSMARL**, a Real-Sim-Real framework that aligns simulator states and actions with hardware, incorporates V2V delays during training, and enforces safety through a CBF-based Safety Shield with pluggable PID or MPC controllers. Experiments in both CARLA and on 1/10th-scale vehicles demonstrate how this integration supports safer and more generalizable MARL-based coordination compared to existing approaches.

3 Approach

We introduce **RSR-RSMARL**, a Real-Sim-Real Robust and Safe Multi-Agent Reinforcement Learning framework that explicitly incorporates real-world constraints into both the simulation-based MARL problem formulation and the training process. The goal is to enable zero-shot transfer of trained policies to physical testbeds, even under testing conditions not observed during training. The overall architecture, shown in Figure 1, is composed of three main components:

1. **State and action space design.** States and actions are aligned with real-world sensing and actuation capabilities to ensure policies trained in simulation are executable on hardware.
2. **Robust and safe MARL training.** Training incorporates communication delays and a safety shield mechanism to improve robustness and ensure safe coordination among connected vehicles.
3. **Real-world deployment.** Policies trained in simulation are transferred to hardware platforms, where onboard sensing and a safety shield ensure safe and reliable execution.

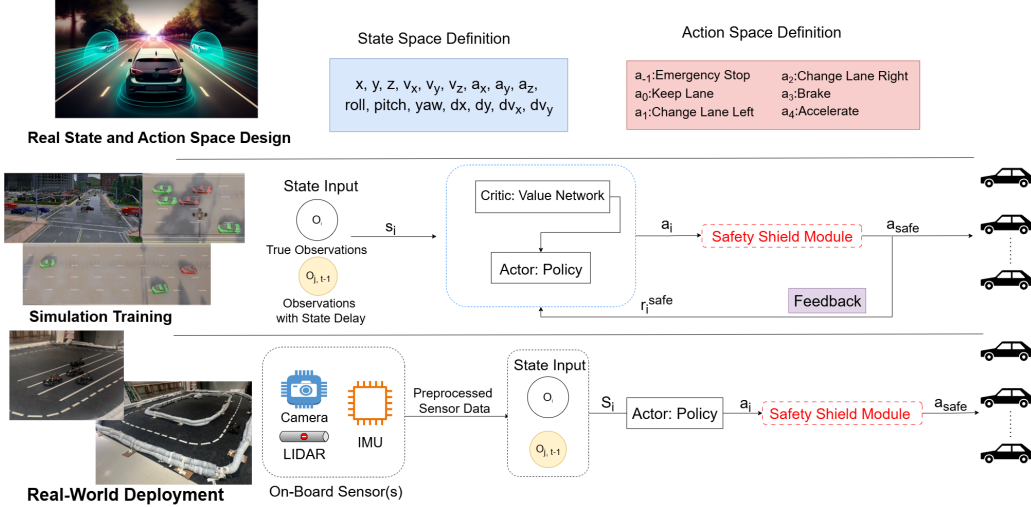


Figure 1: Overview of the RSR-RSMARL framework. Real-world sensor data informs the state and action design used in simulation training. The MARL training process incorporates safety shields and communication delays. Trained policies are then transferred back to hardware, where high-level actions are converted into safe control inputs.

3.1 MARL Formulation with Real-to-Sim Design

The design of a robust and safe MARL problem formulation, together with its policy training algorithm, begins with a grounded understanding of real-world conditions and constraints to ensure practical applicability and safety across the simulation-to-reality gap. By aligning state spaces, action spaces, and inter-agent communication protocols with hardware limitations, the algorithm avoids unrealistic assumptions often present in purely simulated settings, ensuring that state validation is consistently tied back to real-world conditions. Key real-world factors—such as sensor noise, communication and actuation delays, and environmental obstacles, are explicitly modeled to capture uncertainty and variability [27, 10, 7]. Incorporating these complexities enable the simulator-trained MARL policy to generalize effectively across diverse operating conditions while satisfying the stringent safety requirements of CAVs.

In particular, in a multi-agent setting, each CAV, referred to as the ego agent i , receives its own observation o_i and a set of shared observations o_{N_i} communicated by neighboring vehicles N_i via V2V protocols. The shared information includes complementary sensor data and environmental detections, such as obstacle presence and lane boundary features, not directly observable by the ego agent. These measurements are derived from onboard sensors including LiDAR, camera-based lane detection models [28], and others [29]. Additionally, CAVs exchange local state-action histories to mitigate the effects of partial observability and communication delays, enabling more informed action selection. This cooperative approach aims to improve the robustness and safety of decision-making in both simulation and real-world deployments under realistic communication and sensing imperfections. The MARL policy execution flow is illustrated in Figure 2 and elaborated in Section 3.2.

We formulate a MARL problem based on the state, action, and information sharing through communication among agents as a tuple $G = (S, \mathcal{A}, P, r, \gamma)$, where S is the joint state space of all agents. Each agent i 's state s_i includes local observation $o_i := \{l_i, v_i, \alpha_i, d_i, c_i\}$. l_i, v_i, α_i are the ego agent's position, velocity, and acceleration; d_i is processed vision-based features including lane detection results from Ultra Fast Lane Detection (UFLD) [28] and estimated local trajectories; c_i is LIDAR-based collision indicators. \mathcal{A} denotes the discrete action space. When communication is enabled and agent j can share its local observation, $s_{j,i}$ denotes the information sharing between agent j to agent i . The action space a_i of each agent i includes: $a_{i,-1}$: Emergency Stop; $a_{i,0}$: Main-

tain Lane and Speed; $a_{i,1}$: Change Lane Left; $a_{i,2}$: Change Lane Right; $a_{i,3+k}$: Discrete Braking and Acceleration Actions. The unknown stochastic transition dynamics of the entire system with N agents is defined as $P : S \times \mathcal{A} \times S \rightarrow [0, 1]$.

The reward function is defined as:

$$r(s, a) = w_1 \|v_i\|_2 - w_2 \|c_i\| + w_3 \|l_i\| + r_i^{safe}, \quad (1)$$

where w_1 , w_2 , and w_3 are weight coefficients for high speed, collision avoidance, and proximity to the goal, respectively. The safety penalty term $r_i^{safe} = -c$ is applied whenever the Safety Shield intervenes or an unsafe action is detected, discouraging policies from over-relying on corrective filtering and reinforcing proactive safe behavior.

During both training and execution, agents operate with delayed observations $s_i^{del} = \{o_i, o_j^{del}, j \in \mathcal{N}_i\}$ to explicitly capture the effects of V2V communication latency from a neighboring agent j . In practice, this means that when agent i receives information from a neighboring agent j , it does not observe the most recent state $s_{j,t}$ but rather a delayed version $s_{j,t-1}$ that reflects one-step transmission latency. This formulation models realistic network conditions in which packet delays and reliability constraints prevent instantaneous sharing of state information. By embedding these delays directly into the MARL training loop, we encourage the learned decentralized policies $\pi_\theta(a_i | s_i^{del})$ to be robust to temporal misalignment of shared observations, thereby improving coordination performance under real-world communication constraints.

3.2 Safety Shield and Control Integration

A central feature of the framework is the **Safety Shield**, which ensures that only feasible and safe actions are executed. It leverages Control Barrier Functions (CBFs) with quadratic program (QP) constraints to filter unsafe control commands before they are applied to the vehicle. This integration of the Safety Shield with the controller backends is shown in Figure 2, which depicts how high-level policy actions are filtered and then executed through either PID or MPC.

To bridge the gap between high-level policy outputs and low-level actuation, we consider two complementary control backends:

- **PID Controller.** A lightweight option that maps discrete actions to reference controls and regulates them through proportional-integral-derivative feedback. PID control is computationally efficient and straightforward for real-time hardware use [30].
- **MPC Controller.** A model predictive controller that solves an optimization problem at each step, incorporating CBF and CLF constraints. MPC offers greater foresight and smoother trajectories at the expense of increased computational cost [31].

By supporting both PID and MPC, RSR-RSMARL allows for a direct comparison between efficiency and robustness, and demonstrates the flexibility of the framework.

3.3 Algorithm and Training in Simulation

Our complete algorithm design is as Algorithm 1. We adopt the framework of centralized training with decentralized execution (CTDE); we train robust PPO agents where each agent is equipped with an extra worst-case Q network [15, 5] estimating the expected return when agent’s action-selection is potentially affected by inaccurate or perturbed observations. During training, agents experience delayed states to emulate network latency and improve robustness to real-world conditions. The pluggable low-level controller (PID or MPC) ensures that the same MARL training loop can be used across different control architectures.

3.4 Real-World Deployment

For deployment, the trained policies are executed on a connected vehicle testbed using 1/10th-scale autonomous vehicles. Each vehicle estimates its state from onboard LiDAR, cameras, and IMU

Algorithm 1: RSR-RSMARL with Pluggable Low-Level Controller (PID or MPC)

```

1 Initialize policy and critic networks; select controller  $\in \{\text{PID}, \text{MPC}\}$ ;
2 for each episode  $E$  do
3   Initialize state  $s = \prod_i s_i$ , memory  $M = \emptyset$ ;
4   Rollout: for each step, agent  $i$  do
5     Sample delayed state  $s_i^{\text{del}}$  to emulate communication latency;
6     Choose action  $a_i$  from policy  $\pi_{\theta_i}(\cdot | s_i^{\text{del}})$ ;
7     if controller = PID then
8       Compute reference  $u_i^{\text{ref}} \leftarrow \text{PID}(s_i^{\text{del}}, a_i)$ ;
9       Filter with Safety Shield:  $u_i^{\text{safe}} \leftarrow \text{CBF-QP}(u_i^{\text{ref}}, s_i^{\text{del}})$ ;
10    else
11      Solve MPC optimization:  $u_i^{\text{safe}} \leftarrow \text{MPC}(s_i^{\text{del}}, a_i)$ ;
12    end
13    Execute  $u_i^{\text{safe}}$ , observe  $s'$ , compute reward  $r_i$ ;
14    Store  $(s_i^{\text{del}}, a_i, r_i, s'_i)$  in  $M$ ; update  $s \leftarrow s'$ ;
15  end
16  Training: Update critics and policies using PPO and worst-case Q-learning objectives;
17 end

```

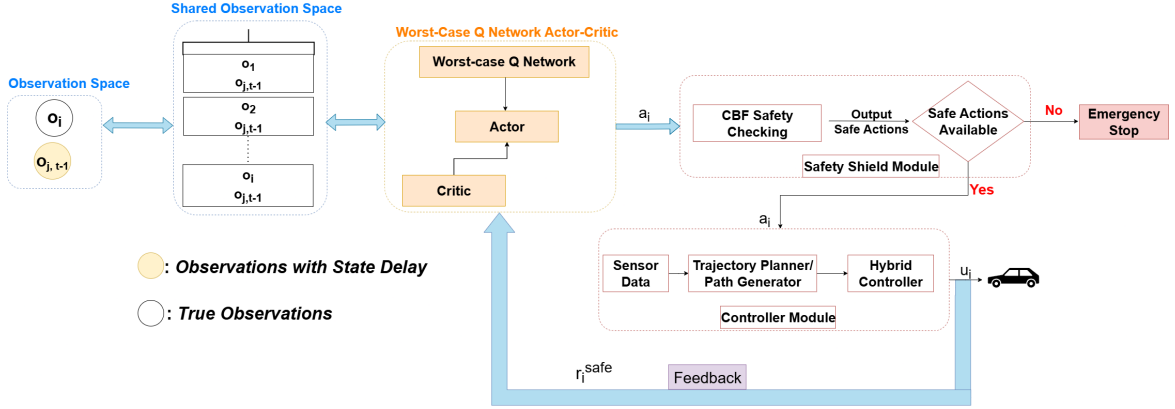


Figure 2: Policy execution pipeline for agent i , with all other agents following the same procedure. During training, both the critic and the worst-case Q network update the actor’s policy under observation delays. At test time, the actor samples a high-level action a_i from uncertain observations, which is first filtered by the CBF-based Safety Shield to ensure safety. If a safe action is available, it is passed to the Controller Module, which integrates sensor data, trajectory planning, and a hybrid low-level controller (PID or MPC) to generate safe control commands u_i . If no safe action exists, the system triggers an emergency stop. This layered architecture ensures robust decision-making, safety enforcement, and reliable real-world execution.

sensors, with additional information exchanged through V2V communication. The MARL policy produces high-level actions that are passed through the CBF-based Safety Shield, which filters unsafe commands before they reach the low-level controller. Depending on the configuration, safe actions are tracked by either a PID controller for lightweight execution or an MPC controller for smoother, optimization-based trajectories. This layered design ensures that real-world execution remains robust to sensing noise, communication delays, and actuation limits, while preserving the safety guarantees established in simulation. Further details on the hardware setup are provided in Section 4.1.

4 Experiments and Results

In our experiments, we seek to address the following research questions:

1. **Policy transfer.** Can RSR-RSMARL effectively transfer policies from simulation to the real world by leveraging real-world system states and actions as design constraints, while accounting for uncertainties such as communication delays, sensor noise, and inter-vehicle dynamics?
2. **Hybrid control performance.** How does RSR-RSMARL’s hybrid control architecture—combining RL policies with traditional controllers and Control Barrier Functions (CBFs)—compare to alternative baselines in terms of safety and coordination?
3. **Representation robustness.** How critical are robust state and action representations in enabling scalable and generalizable Sim2Real transfer for multi-agent autonomous vehicle systems?

4.1 Simulator and Hardware Testbed Setup

We use the CARLA Simulator environment [32] for robust policy training and evaluation; every vehicle is equipped with an onboard GPS, a collision sensor, and IMU sensors. The simulation was conducted on a server configured with an AMD Ryzen 3970X processor and an NVIDIA Quadro RTX 6000 GPU. Experiments were performed with CARLA 0.9.15, Python 3.10, PyTorch 2.6, and CUDA 12.2.

For real-world evaluation, we deploy trained policies on a fleet of 1/10th-scale F1TENTH autonomous vehicles. Each vehicle is equipped with a 2D LiDAR (Hokuyo UST-10LX), a Logitech C270 USB webcam, and an onboard IMU, providing state estimates including position, velocity, and yaw rate. Vehicle-to-vehicle (V2V) communication is established through Wi-Fi at 5 Hz, with communication latency empirically measured between 10–20 ms.

Communication delays are measured using synchronized timestamps with full network stack instrumentation. While our controlled testbed achieves 10-20ms latency, we acknowledge that production V2X networks exhibit significantly higher variability (50-200ms) due to congestion, handoffs, and security overhead. To account for this gap, our training incorporates stochastic delay sampling from realistic delay distributions observed in urban V2V deployments.

The onboard computation platform is an NVIDIA Jetson Orin Nano running ROS Noetic on Ubuntu 20.04, executing both the MARL policy and the CBF-based Safety Shield in real time. Policy inference and safety filtering run at 10 Hz, while low-level control commands are issued to the VESC motor controller at 50 Hz. This configuration enables closed-loop policy execution under realistic sensing, communication, and control constraints, closely mirroring the conditions modeled in simulation.

4.2 Real-World Evaluation

We evaluate the proposed framework in real-world environments using two distinct driving scenarios: a 3-lane miniature highway (Figure 3) and a 2-lane circular highway. Each environment is tested under three levels of complexity, corresponding to the number of obstacles introduced.

Table 1 presents the evaluation results across all scenarios. On the 3-lane highway, RSR-RSMARL achieves zero collisions across all obstacle variations while maintaining completion times between 34.2 and 36.1 seconds. In contrast, RSR-MARL, which excludes the Safety Shield module, exhibits a growing number of collisions as the number of obstacles increases. Although RSR-MARL completes each scenario faster, this speed comes at the cost of erratic and risk-prone driving behavior. These findings underscore that faster completion does not imply a safer or better driving strategy. The slightly slower pace of RSR-RSMARL is a deliberate trade-off that prioritizes collision avoidance and real-time safety guarantees—hallmarks of a robust and safe system.

The No-Comm RSR-RSMARL variant, which disables V2V communication, results in slightly longer completion times and occasional collisions under perturbations. This outcome underscores the additional robustness afforded by inter-agent communication in dynamic environments.

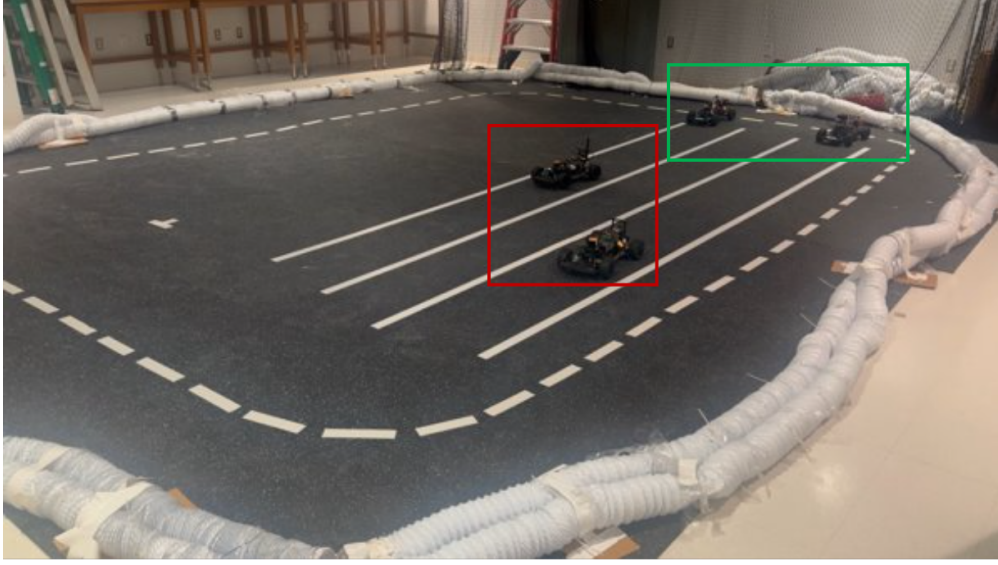


Figure 3: Hardware evaluation setting with 1/10th-scale F1TENTH vehicles. Green boxes denote ego vehicles running RSR-RSMARL policies, while red boxes indicate obstacle vehicles used to vary scenario complexity.

A comparable trend appears in the 2-lane oval highway scenario. With PID control, RSR-RSMARL consistently achieves zero collisions, completing runs between 47.6 and 49.7 seconds. By contrast, the RSR-MARL baseline degrades as obstacle complexity increases, and while the No-Comm variant remains collision-free in the simplest setting, it struggles once the environment becomes more challenging.

The integration of **MPC** within RSR-RSMARL further enhances safety and optimality. Although **MPC introduces a modest increase in computational load per control step relative to PID**, the resulting policies generate smoother trajectories, exhibit greater resilience to uncertainty, and consistently select safer actions. This trade-off highlights a core advantage of optimization-based control: even if execution is marginally slower, the quality of decision-making improves substantially, reducing collision risk and strengthening multi-agent coordination.

Taken together, these results validate the real-world effectiveness of RSR-RSMARL in delivering safe and efficient autonomous driving under partially observable conditions. The combination of a Safety Shield, V2V communication, and either PID- or MPC-based low-level control significantly enhances robustness and reliability, positioning RSR-RSMARL as a practical and generalizable framework for scalable multi-agent autonomy.

4.3 Benchmarking in Simulation

We benchmarked our method in a simulated highway scenario against several baselines, analyzing the impact of communication delay and CBF removal on collision rate and efficiency return.

The efficiency return shown in Figure ?? captures only the reward from task performance (e.g., speed and goal-reaching). Penalties for safety violations such as collisions are applied separately and detailed in Table 2. This separation helps clarify why some methods may exhibit high efficiency returns despite high collision rates.

Figure 4 and 5 shows that RSR-RSMARL achieved the highest efficiency with zero collisions, demonstrating its strength in both safety and performance. Table 2 presents results averaged over 50 test episodes. As shown removing CBFs, even with communication, significantly increases collision risk. While CBF alone reduces collisions, the best results come from combining safety enforcement with reliable communication. This supports the need for both in robust multi-agent autonomy.

Table 1: Evaluation Results on 3-Lane Highway and 2-Lane Oval Highway

Method	Number of Obstacles		
	None	1	2
<i>3-Lane Highway</i>			
RSR-RSMARL (MPC)	0, 32.8	0, 33.9	0, 34.2
RSR-RSMARL (PID)	0, 34.2	0, 35.4	0, 36.1
Safe-RMM (MPC)	0, 34.5	1, 35.3	2, 36.1
RSR-MARL (PID)	1, 36.8	2, 37.5	3, 38.0
No-Comm RSR-RSMARL (PID)	0, 36.7	1, 37.9	1, 39.5
MARL-DR	2, 36.5	3, 37.8	5, 39.2
<i>2-Lane Oval Highway</i>			
RSR-RSMARL (MPC)	0, 48.1	0, 49.4	0, 51.6
RSR-RSMARL (PID)	0, 47.6	0, 48.9	0, 49.7
Safe-RMM (MPC)	0, 50.2	1, 51.5	2, 53.0
RSR-MARL (PID)	1, 51.3	2, 52.5	3, 53.1
No-Comm RSR-RSMARL (PID)	0, 54.8	1, 56.2	1, 57.1
MARL-DR	2, 52.0	3, 57.6	5, 59.0

Summary—RSR-RSMARL consistently achieves zero collisions and stable completion times, outperforming baseline methods that either lack safety guarantees or degrade under communication delays.

Each entry reports results as (number of collisions; completion time in seconds).

RSR-RSMARL (MPC): Robust Real-Sim-Real MARL policy with Safety Shield, V2V communication, and MPC-based low-level control.

RSR-RSMARL (PID): Robust Real-Sim-Real MARL policy with Safety Shield, V2V communication, and PID-based low-level control.

Safe-RMM (MPC): Baseline MARL with CBF-based safety but without communication delay modeling [33].

RSR-MARL (PID): Variant of the framework without the Safety Shield module.

No-Comm RSR-RSMARL (PID): RSR-RSMARL variant without V2V communication.

MARL-DR: Domain randomization variant adapted for multi-agent CAV settings [34, 35].

Finally, to further examine robustness under sim-to-real discrepancies, we benchmark our approach against a domain randomization variant.

4.4 Advantages of Robust MARL Compared with Domain Randomization

Domain Randomization (MARL-DR) is often used to improve policy robustness by injecting stochastic noise into state observations, thereby simulating discrepancies encountered during sim-to-real transfer. In our ablation study, we applied domain randomization across our vehicle states during evaluation to test whether this approach alone could achieve reliable performance in multi-agent driving tasks.

As summarized in Table 2, MARL-DR policies achieve moderate returns and outperform methods without robustness enhancements. However, the results also show that MARL-DR suffers from a higher number of collisions and reduced efficiency compared to our full RSR-RSMARL framework. This outcome highlights a key limitation: while domain randomization can provide some resilience to observation noise, it does not address the fundamental challenges of inter-agent coordination, communication delays, and real-time safety enforcement. By contrast, RSR-RSMARL integrates a CBF-based Safety Shield and communication-aware training, which together enable zero-collision execution and consistently higher efficiency.

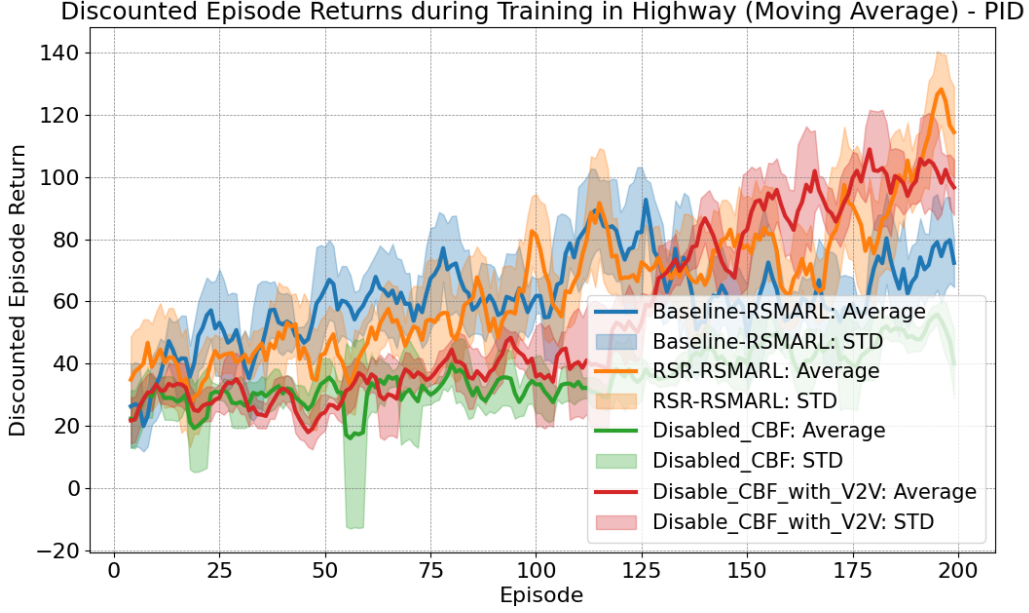


Figure 4: Discounted Efficiency Returns during Training with PID Controller. *The RSR-RSMARL method (orange curve) consistently achieves the highest discounted returns compared to all base-lines, highlighting its superior robustness and safety during training.*

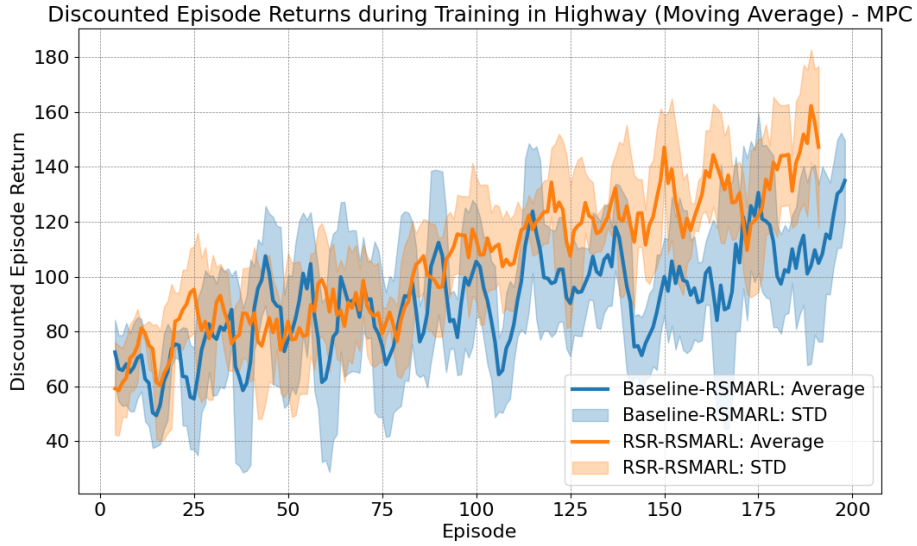


Figure 5: Discounted Efficiency Returns during Training with MPC Controller. *Compared to the baseline RSMARL policy (blue curve), our RSR-RSMARL method (orange curve) achieves consistently higher returns and demonstrates smoother convergence, confirming the benefit of integrating the CBF-based Safety Shield and real-sim-real training.*

4.5 Impact of Communication on Safety and Coordination

The presence of V2V communication significantly improved cooperative behavior among agents, especially in scenarios involving lane merging and intersection negotiation. Figure 6 compares the frequency of safety violations and CBF activations between communication-enabled and communication-disabled deployments.

Table 2: Evaluation Results

Method	Number of Collisions	Mean Discounted Efficiency Return
RSR-RSMARL (MPC)	0	142.80
RSR-RSMARL (PID)	0	129.52
MARL-DR	10	95.07
Safe-RMM (MPC)	15	75.65
RSR-MARL	42	28.84
Non-Robust RSR-MARL	45	25.26

Simulation benchmarking results across different methods. Each entry reports the total number of collisions (over 50 test episodes) and the mean discounted efficiency return.

RSR-RSMARL (MPC): Proposed robust and safe Real-Sim-Real MARL policy with Safety Shield, V2V communication, and MPC-based low-level control.

RSR-RSMARL (PID): Proposed robust and safe Real-Sim-Real MARL policy with Safety Shield, V2V communication, and PID-based low-level control.

MARL-DR: MARL policy evaluated with domain-randomized input noise across vehicle states.

Safe-RMM (MPC): Baseline MARL implementation with CBF-based safety but without communication delay modeling [33].

RSR-MARL: Framework variant without the Safety Shield module.

Non-Robust RSR-MARL: Variant without the Safety Shield and with communication delay.

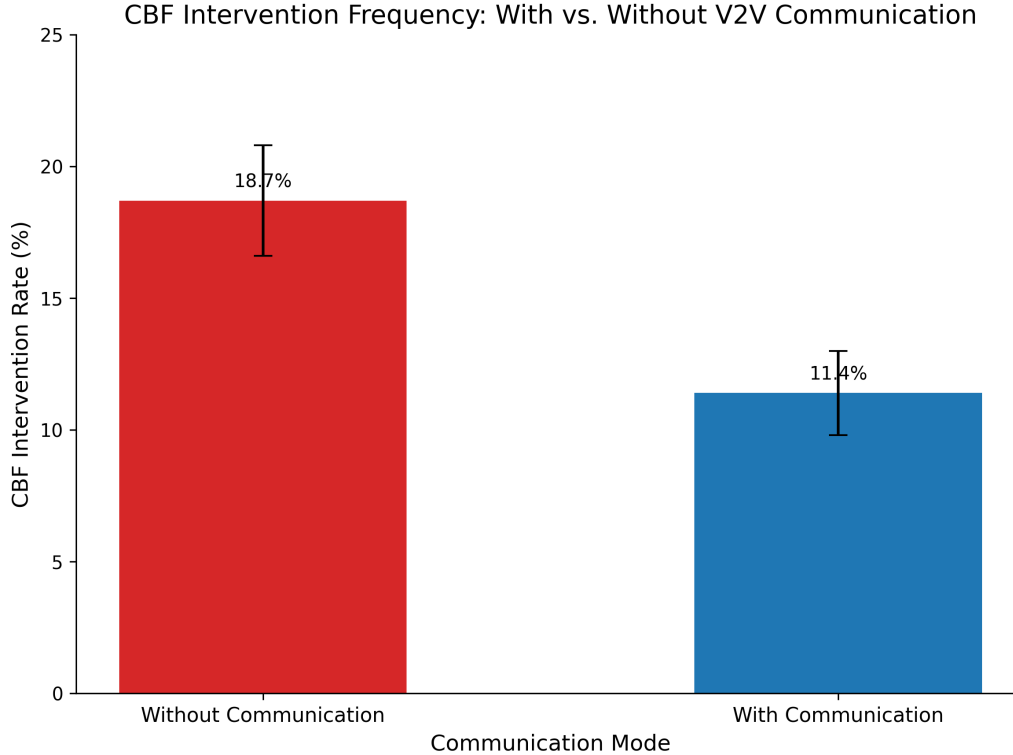


Figure 6: CBF Intervention Frequency: With vs. Without V2V Communication

The data indicate that shared information reduces both the need for reactive overrides and the likelihood of unsafe maneuvers. Furthermore, the coordination enabled by V2V helped stabilize policy

outputs during high-density interactions, reducing the average intervention rate from 18.7% (no communication) to 11.4% (with communication).

5 Conclusion

In this work, we propose **RSR-RSMARL**, a robust and safe multi-agent reinforcement learning framework that enables Real-Sim-Real policy transfer for connected autonomous vehicles with vehicle-to-vehicle communication. By aligning the state and action spaces with real-world sensing and actuation capabilities, incorporating communication delays into the training loop, and enforcing safety through a CBF-based Safety Shield, our framework bridges the gap between simulation and real-world deployment.

Through extensive experiments in both CARLA simulation and on 1/10th-scale autonomous vehicles, we demonstrate that RSR-RSMARL achieves zero-shot policy transfer while maintaining safety guarantees and effective coordination across agents. Results consistently show that the integration of the Safety Shield and inter-agent communication significantly reduces collision risk, while the inclusion of MPC further improves trajectory smoothness and resilience to uncertainty compared to PID-based execution. Moreover, by explicitly modeling state-delays during training, RSR-RSMARL policies learn to anticipate and adapt to communication latencies and sensing imperfections, leading to more robust performance in real-world deployments. Taken together, these findings establish RSR-RSMARL as a practical and generalizable framework for safe, multi-agent autonomy.

References

- [1] U. I. J. P. Office. Saving lives with connectivity: A plan to accelerate v2x deployment non-binding contents, 2024.
- [2] Z. Zhang, S. Han, J. Wang, and F. Miao. Spatial-temporal-aware safe multi-agent reinforcement learning of connected autonomous vehicles in challenging scenarios. pages 5574–5580, 2023.
- [3] N. Hyldmar, Y. He, and A. Prorok. A fleet of miniature cars for experiments in cooperative driving. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:3238–3244, 5 2019. ISSN 10504729. doi:10.1109/ICRA.2019.8794445.
- [4] A. Miller, K. Rim, P. Chopra, P. Kelkar, and M. Likhachev. Cooperative perception and localization for cooperative driving. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1256–1262, 5 2020. ISSN 10504729. doi:10.1109/ICRA40945.2020.9197463.
- [5] Z. Zhang, H. M. S. Ahmad, E. Sabouni, Y. Sun, F. Huang, W. Li, and F. Miao. Safety guaranteed robust multi-agent reinforcement learning with hierarchical control for connected and automated vehicles. 9 2023. URL <https://arxiv.org/abs/2309.11057v2>.
- [6] 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. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 8765–8771, 3 2022. ISSN 10504729. doi:10.1109/ICRA46639.2022.9811626. URL <https://arxiv.org/abs/2203.06333v2>.
- [7] 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 Transactions on Intelligent Transportation Systems*, 18:1066–1077, 5 2017. ISSN 15249050. doi:10.1109/TITS.2016.2600504.
- [8] S. Han, S. Zhou, J. Wang, L. Pepin, C. Ding, J. Fu, and F. Miao. A multi-agent reinforcement learning approach for safe and efficient behavior planning of connected autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):3654–3670, 2024. doi:10.1109/TITS.2023.3336670.
- [9] A. Mokhtarian, P. Scheffe, M. Kloock, S. Schäfer, Heeseung Bang, Viet-Anh Le, Sangeet Ulhas, J. Betz, S. Wilson, S. Berman, A. Prorok, and B. Alrifaa. A survey on small-scale testbeds for connected and automated vehicles and robot swarms. 2024. doi:10.13140/RG.2.2.16176.74248/1. URL <https://arxiv.org/abs/2408.03539>.
- [10] Y. Shao, M. A. M. Zulkefli, Z. Sun, and P. Huang. Evaluating connected and autonomous vehicles using a hardware-in-the-loop testbed and a living lab. *Transportation Research Part C: Emerging Technologies*, 102:121–135, 5 2019. ISSN 0968-090X. doi:10.1016/J.TRC.2019.03.010.
- [11] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone. Deep reinforcement learning for robotics: A survey of real-world successes. 8 2024. doi:10.1146/((please)). URL <https://arxiv.org/abs/2408.03539v2>.
- [12] Y. Feng, C. Hong, Y. Niu, S. Liu, Y. Yang, W. Yu, T. Zhang, J. Tan, and D. Zhao. Learning multi-agent loco-manipulation for long-horizon quadrupedal pushing, accepted, ICRA2025. URL <https://arxiv.org/abs/2411.07104>.
- [13] P. Werner, T. Seyde, P. Drews, T. M. Balch, I. Gilitschenski, W. Schwarting, G. Rosman, S. Karaman, and D. Rus. Dynamic multi-team racing: Competitive driving on 1/10-th scale vehicles via learning in simulation. In *7th Annual Conference on Robot Learning*, 2023. URL <https://openreview.net/forum?id=fvXFBCHVGn>.

- [14] S. Han, S. Su, S. He, S. Han, H. Yang, and F. Miao. What is the solution for state adversarial multi-agent reinforcement learning? *arXiv preprint arXiv:2212.02705*, 2022.
- [15] Y. Liang, Y. Sun, R. Zheng, and F. Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22547–22561, 2022.
- [16] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. 3 2024. URL <https://arxiv.org/abs/2403.03949v1>.
- [17] M. T. Villasevil, A. Jain, V. Macha, J. Yuan, L. L. Ankile, A. Simeonov, P. Agrawal, and A. Gupta. Scaling robot-learning by crowdsourcing simulation environments.
- [18] W. Zhao, J. P. Queralta, and T. Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020*, pages 737–744, 12 2020. doi:10.1109/SSCI47803.2020.9308468.
- [19] Y. Jiang, C. Wang, R. Zhang, J. Wu, and L. Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. In *Conference on Robot Learning*, 2024.
- [20] S. S. Sandha, L. Garcia, B. Balaji, F. Anwar, and M. Srivastava. Sim2real transfer for deep reinforcement learning with stochastic state transition delays. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1066–1083. PMLR, 16–18 Nov 2021. URL <https://proceedings.mlr.press/v155/sandha21a.html>.
- [21] J. Blumenkamp, A. Shankar, M. Bettini, J. Bird, and A. Prorok. The cambridge robomaster: An agile multi-robot research platform. 5 2024. URL <https://arxiv.org/abs/2405.02198v2>.
- [22] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.
- [23] I. ElSayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, and L. Feng. Safe multi-agent reinforcement learning via shielding. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS ’21*, page 483–491, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450383073.
- [24] Z. Cai, H. Cao, W. Lu, L. Zhang, and H. Xiong. Safe multi-agent reinforcement learning through decentralized multiple control barrier functions, 2021.
- [25] J. Wang, S. Yang, Z. An, S. Han, Z. Zhang, R. Mangharam, M. Ma, and F. Miao. Multi-agent reinforcement learning guided by signal temporal logic specifications. *arXiv preprint arXiv:2306.06808*, 2023.
- [26] S. He, S. Han, S. Su, S. Han, S. Zou, and F. Miao. Robust multi-agent reinforcement learning with state uncertainty. *Transactions on Machine Learning Research*, 2023.
- [27] A. Mokhtarian, P. Scheffe, M. Kloock, S. Schäfer, Heeseung Bang, Viet-Anh Le, Sangeet Ulhas, J. Betz, S. Wilson, S. Berman, A. Prorok, and B. Alrifae. A survey on small-scale testbeds for connected and automated vehicles and robot swarms. 2024. doi:10.13140/RG.2.2.16176.74248/1. URL <https://rgdoi.net/10.13140/RG.2.2.16176.74248/1>.
- [28] Z. Qin, H. Wang, and X. Li. Ultra fast structure-aware deep lane detection. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 276–291. Springer, 2020.

- [29] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7:10914–10921, 2 2022. ISSN 23773766. doi:10.1109/LRA.2022.3192802. URL <https://arxiv.org/abs/2202.08449v2>.
- [30] L. Berducci, S. Yang, R. Mangharam, and R. Grosu. Learning adaptive safety for multi-agent systems. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2859–2865. Institute of Electrical and Electronics Engineers Inc., 2024. ISBN 9798350384574. doi:10.1109/ICRA57147.2024.10611037.
- [31] E. Sabouni, H. M. S. Ahmad, V. Giammarino, C. G. Cassandras, I. C. Paschalidis, and W. Li. Reinforcement learning-based receding horizon control using adaptive control barrier functions for safety-critical systems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 401–406. Institute of Electrical and Electronics Engineers Inc., 2024. ISBN 9798350316339. doi:10.1109/CDC56724.2024.10886217.
- [32] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. pages 1–16, 2017.
- [33] Z. Zhang, H. M. S. Ahmad, E. Sabouni, Y. Sun, F. Huang, W. Li, and F. Miao. Safety guaranteed robust multi-agent reinforcement learning with hierarchical control for connected and automated vehicles. 9 2023. URL <https://arxiv.org/abs/2309.11057v2>.
- [34] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *IEEE International Conference on Intelligent Robots and Systems*, 2017-September:23–30, 3 2017. ISSN 21530866. doi:10.1109/IROS.2017.8202133. URL <https://arxiv.org/abs/1703.06907v1>.
- [35] B. Mehta, M. D. Mila, F. G. Mila, C. J. P. Mila, P. Montréal, and C. L. Paull. Active domain randomization, 5 2020. ISSN 2640-3498. URL <https://proceedings.mlr.press/v100/mehta20a.html>.