

# Deep Reinforcement Learning for Autonomous Vehicle Intersection Navigation

Badr Ben Elallid  
Moulay Ismail University  
Meknes, Morocco  
badr.benelallid@edu.umi.ac.ma

Hamza El Alaoui  
AL Akhawayn University  
Ifrane, Morocco  
h.elalaoui@aiui.ma

Nabil Benamar  
AL Akhawayn University  
Ifrane, Morocco  
n.benamar@umi.ac.ma

**Abstract**—In this paper, we explore the challenges associated with navigating complex T-intersections in dense traffic scenarios for autonomous vehicles (AVs). Reinforcement learning algorithms have emerged as a promising approach to address these challenges by enabling AVs to make safe and efficient decisions in real-time. Here, we address the problem of efficiently and safely navigating T-intersections using a lower-cost, single-agent approach based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) reinforcement learning algorithm. We show that our TD3-based method, when trained and tested in the CARLA simulation platform, demonstrates stable convergence and improved safety performance in various traffic densities. Our results reveal that the proposed approach enables the AV to effectively navigate T-intersections, outperforming previous methods in terms of travel delays, collision minimization, and overall cost. This study contributes to the growing body of knowledge on reinforcement learning applications in autonomous driving and highlights the potential of single-agent, cost-effective methods for addressing more complex driving scenarios and advancing reinforcement learning algorithms in the future.

**Index Terms**—Autonomous Vehicles, Reinforcement Learning, Twin Delayed Deep Deterministic Policy Gradient, Intersection Navigation, CARLA Simulator

## I. INTRODUCTION

Intersections present a considerable challenge to road safety due to their intricate traffic conditions, accounting for 36% of road collisions [1]. Conventional methods of controlling vehicle flow, such as traffic lights and stop signs, often hinder traffic progression and restrict intersection capacity. Autonomous driving strategies hold the potential to enhance intersection navigation by minimizing collisions caused by human error and optimizing driving behavior to reduce travel times [2], [3].

Existing intersection navigation solutions in autonomous driving predominantly rely on vehicle-to-vehicle (V2V) communication and centralized systems. In this paper, we investigate the prospects of a single-agent approach employing Reinforcement Learning (RL) techniques to develop advanced, adaptive, and cost-efficient solutions for traversing complex intersections [4], [5].

In this study, we apply the Twin-Delayed DDPG algorithm (TD3) to facilitate autonomous vehicles' (AVs) intersection navigation without relying on V2V communication or centralized systems. Our model trains AVs to arrive at their destinations without collisions by processing features extracted

from images produced by the vehicle's front camera sensor and employing TD3 to predict the optimal action for each state. We simulate and train the proposed method using the CARLA simulator. Simulation results demonstrate the capacity of our model to learn over episodes in terms of reducing travel delay and collision rate.

The remainder of the paper is structured as follows: Section II reviews related works; Section III presents the problem formulation, including the state space, actions, and reward functions; Section IV details the simulation setup and experimental design; Section V presents our simulation results. Finally, Section VI concludes the paper and highlights future work and potential research directions.

## II. RELATED WORKS

In recent years, autonomous driving research has advanced significantly, particularly in the area of intersection management. Two primary solutions have emerged: vehicle-to-vehicle (V2V) communication and centralized solutions. Concurrently, reinforcement learning (RL) algorithms have gained traction within autonomous driving, especially for navigation tasks. However, their application in intersection navigation remains under investigated. Thus, while V2V and centralized strategies dominate the literature, RL-based approaches present a promising yet unexplored research direction.

### A. Vehicle-to-Vehicle Solutions

V2V communication has been proposed as a way to improve traffic throughput and safety at intersections for autonomous vehicles. In [6], the authors investigate V2V communication for cooperative driving, particularly in intersection management. They propose V2V intersection protocols that not only enhance traffic throughput but also prevent deadlock situations. Simulation results demonstrate considerable improvements in both performance and safety.

Another study [7] explores cooperative driving at blind intersections, which lack traffic lights, utilizing intervehicle communication. The authors introduce safety driving patterns representing collision-free movements and develop trajectory planning algorithms aiming to minimize execution time. Simulated results underscore the potential and utility of the proposed algorithms.

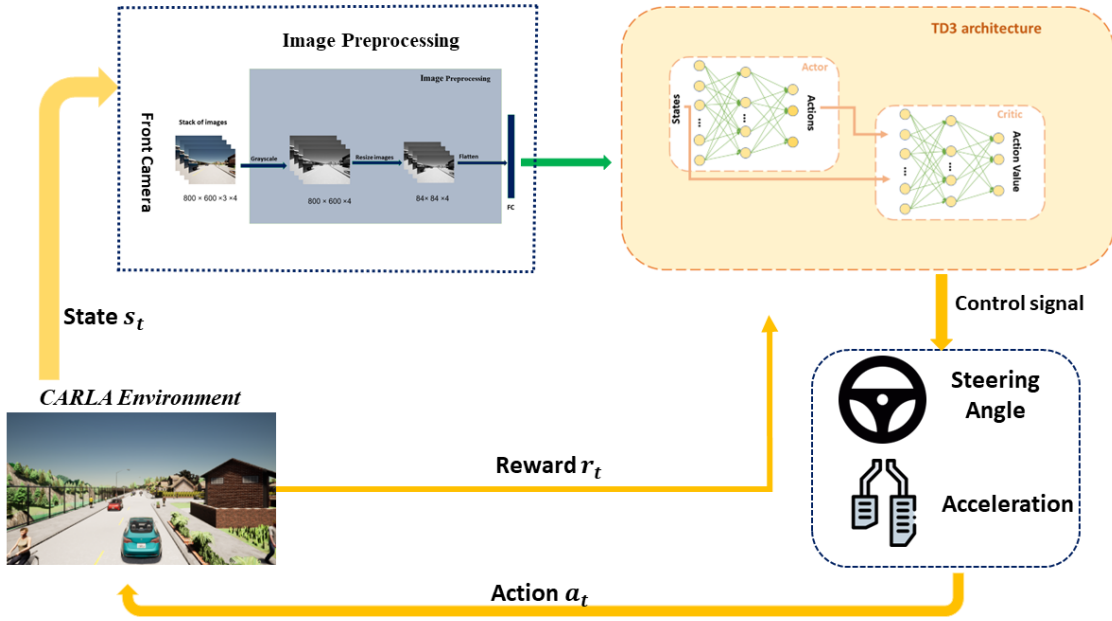


Fig. 1: TD3-based Deep Reinforcement Learning Architecture for Controlling an AV in a T-intersection Scenario with Dense Traffic

### B. Centralized Solutions

Centralized approaches to intersection management typically rely on a Roadside Unit (RSU) to coordinate the crossing sequence. However, this can be costly, as each intersection would require an RSU. For example, in [8], the authors present a centralized Model Predictive Control (MPC) approach for the optimal control of autonomous vehicles within an intersection control area. The problem is formulated as a convex quadratic program, enabling efficient solutions.

In [9], researchers propose an autonomous T-intersection strategy that combines motion-planning and path-following control, considering oncoming vehicles. Through CarSim simulations [10] and scaled car experiments, the effectiveness of the motion planner in generating collision-free trajectories and the path-following controller in ensuring safe and swift intersection navigation is demonstrated.

In [11], a Cooperative Intersection Control (CIC) methodology is developed to improve T-intersection navigation for autonomous vehicles. By employing virtual platoons and simulating six vehicles crossing the intersection, the authors demonstrate increased traffic efficiency without stopping. Although considerable progress has been made in intersection management through V2V communication and centralized solutions, exploring alternative approaches remains crucial. Employing reinforcement learning (RL) techniques within a single-agent framework for intersection navigation has the potential to produce adaptive and advanced solutions. These solutions could not only improve the efficiency and safety of autonomous vehicles in intricate intersections but also potentially offer cost advantages over multi-agent and

centralized methods.

### C. Reinforcement Learning Algorithms

Reinforcement learning (RL) is a branch of machine learning focused on training agents to make decisions by interacting with their environment. In RL, an agent learns an optimal policy, which maps states to actions, by maximizing the cumulative reward it receives from the environment. This learning process typically involves exploring the environment to gather information and exploiting the knowledge acquired to optimize actions [12]

Popular RL algorithms, such as Deep Q-Network [13], Deep Deterministic Policy Gradient [14], Proximal Policy Optimization [15], and Soft Actor-Critic [16], have been employed to address various challenges in autonomous driving, including intersection navigation. However, their performance and suitability for different driving scenarios can vary significantly.

Twin Delayed DDPG [17] has emerged as a relevant and promising algorithm for autonomous vehicles because of its stability, reduced overestimation bias, and improved exploration capabilities. TD3 is an off-policy actor-critic algorithm that extends DDPG by incorporating three key enhancements:

- 1) *Twin Q-networks*: which are used to mitigate overestimation bias by maintaining two separate Q-function approximators and taking the minimum value of the two.

- 2) *Delayed policy updates*: wherein the actor and target networks are updated less frequently than the Q-networks to improve stability.

3) *Target policy smoothing*: which adds noise to the target actions during the learning process to encourage exploration and prevent overfitting to deterministic policies.

These improvements have enabled TD3 to achieve superior performance in a variety of tasks, including intersection management, compared to its predecessor DDPG [17]. Additionally, TD3 offers an efficient approach for learning complex decision-making policies required for navigating intersections safely, making it well-suited for autonomous driving applications.

#### D. Simulation Environments for Autonomous Driving Research

Various simulation platforms have been developed to facilitate the evaluation of reinforcement learning (RL) algorithms for autonomous driving applications. Widely used platforms include CARLA, SUMO, Gazebo, CarSim, and LGSVL. Each platform offers unique advantages and capabilities tailored to different aspects of autonomous driving research.

CARLA (Car Learning to Act) is an open-source simulator specifically designed for autonomous driving research, providing high-fidelity urban environments, diverse traffic scenarios, and a range of weather conditions. It enables comprehensive testing and validation of autonomous driving algorithms, including RL approaches [18].

SUMO (Simulation of Urban Mobility) is a microscopic traffic simulator that models individual vehicles and their behavior in various traffic situations. It is particularly beneficial for large-scale simulations and can be integrated with other platforms or tools, allowing researchers to investigate the impact of RL algorithms on overall traffic flow and intersection management [19].

Gazebo is a versatile and extensible 3D robotics simulator that supports multiple physics engines, sensor models, and control interfaces. It facilitates the simulation of intricate autonomous driving scenarios, making it well-suited for the evaluation of RL algorithms in dynamic and challenging environments [20].

CarSim is a commercial vehicle dynamics simulation software that offers high-fidelity vehicle models and realistic driving environments. It allows for the integration of control systems and sensors, enabling researchers to evaluate the performance of RL algorithms in terms of vehicle control and dynamics [10].

LGSVL is a multi-robot AV simulator developed by LG Electronics America R&D Center, providing an out-of-the-box solution for testing autonomous vehicle algorithms. It offers photo-realistic virtual environments, sensor simulation, and vehicle dynamics, and supports integration with popular AD stacks such as Autoware and Baidu Apollo [21].

In our research, we opted for the CARLA simulator platform, given its focus on autonomous driving and comprehensive feature set. CARLA offers high-fidelity urban environments, diverse traffic scenarios, and a range of weather conditions, enabling rigorous testing and validation of our

reinforcement learning (RL) algorithms [18]. Additionally, CARLA's open-source nature and active community support make it a highly accessible and extensible tool for our research purposes. The platform's compatibility with Python allows seamless integration with machine learning frameworks, such as PyTorch and TensorFlow, streamlining the development process. Consequently, the combination of CARLA and Python provides a powerful and accessible foundation for our autonomous driving research.

### III. PROBLEM FORMULATION

- **State space**: Human drivers rely on more than just their visual perception to comprehend their surroundings; they also take into account the motion of other road users. Analogously, an autonomous vehicle (AV) must utilize a sequence of images to grasp the movement of objects within its environment. In this particular instance, our model processes a series of four consecutive RGB images acquired by the AV's front camera. These images have dimensions of  $800 \times 600 \times 3 \times 4$  pixels, which we subsequently resize to  $84 \times 84 \times 3 \times 4$  pixels and convert into grayscale. The resulting state  $S_t$  possesses dimensions of  $84 \times 84 \times 4$ , which are then input into our Actor and Critic architecture.
- **Action space**: The AV in the CARLA simulator receives three control commands from the environment: acceleration, steering, and braking. These commands are represented by float values ranging  $[0, 1]$  for acceleration,  $[-1, 1]$  for steering, and  $[0, 1]$  for braking. Since TD3 is a continuous DRL algorithm, the agent must select actions continuously. Therefore, at each step  $t$ , the agent needs to choose an action represented by (acceleration, steering, brake) while ensuring that each command falls within its respective range.
- **Reward function** : In our scenario, we formulate a reward function that takes into account potential situations in urban traffic. The AV must not collide with other participants on the road, such as vehicles, cyclists, motorcycles, and pedestrians, while also successfully reaching its intended destination. For this purpose, the reward function includes four components: a reward for maintaining speed, a penalty for collisions, driving in other or off the road, and the distance to the goal. We can represent the current distance to the goal as  $D_{cu}$ , the previous distance to the goal as  $D_{pre}$ , the velocity of the AV as  $V_{speed}$ , and the measure of the AV being off or in another lane as  $M_{offroad}$  and  $M_{otherlane}$ , respectively.

$$reward = \begin{cases} R_{t1} = -C_{collision} \\ R_{t2} = D_{pre} - D_{cu} \\ R_{t3} = \max(0, \min(V_{speed}, V_{limit})) \\ R_{t4} = -M_{offroad} - M_{otherlane} \\ R_{t5} = 100 \\ R_t = R_{t1} + R_{t2} + R_{t4} + R_{t5} \end{cases}$$

Where  $V_{limit}$  speed limit and  $C_{collision}$  is the penalty for colliding with road users such as vehicles, pedestrians, cyclists, and motorcycles.

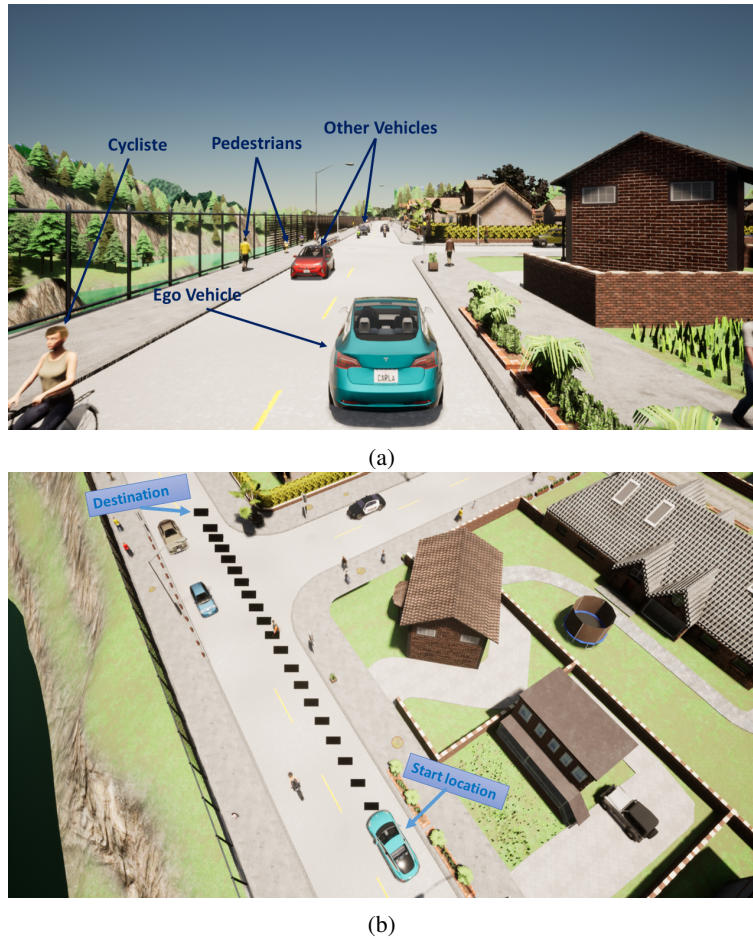


Fig. 2: Scenario Considered: (a) The Ego Vehicle Navigating Through a T-intersection with Heavy Traffic; (b) The Path the Autonomous Vehicle (AV) Follows to Reach its Desired Destination

- **Training:** Our TD3-based deep reinforcement learning architecture comprises of two networks: actor and critic. The actor-network takes the current state as input and generates the next action for the agent. On the other hand, the critic network predicts the action value based on both the state and the value obtained from the actor. To encourage exploration during training, the actor-network adds noise to the predicted action. We have set the number of hidden layers to two, with each layer containing 256 neurons. Other parameters are presented in Table I

#### IV. SIMULATION SETUP

We employed the CARLA simulator (version 0.9.10) and PyTorch for our autonomous driving experiments, focusing on a T-intersection scenario within the CARLA environment. This scenario presents a complex and challenging situation for autonomous driving systems due to its unique traffic dynamics.

Vehicles in the T-intersection scenario must navigate multiple lanes while interacting with other road users, such as managing oncoming traffic, merging with traffic flow, and responding to pedestrians at designated crossing locations. The

numerous real-time decisions and actions required for safe and efficient navigation contribute to the scenario's complexity.

To create a comprehensive and realistic simulation, we included various traffic participants like pedestrians, cars, bicycles, and motorcycles. This diverse set of road users enables a thorough assessment of the autonomous driving algorithm's performance across a wide range of traffic situations and challenges.

We adjusted parameters including vehicle speeds, distances between vehicles, and the frequency of traffic participants entering the simulation to emulate dense traffic conditions. This allowed us to subject our algorithms to demanding circumstances that closely resemble real-world driving conditions, essential for evaluating the effectiveness of our proposed solutions in complex driving environments.

Figure 2 depicts a dense traffic scenario where an autonomous vehicle safely navigates a T-intersection. The vehicle begins at a start point, follows a path to its destination, and avoids collisions. The scene includes 300 vehicles—motorcycles, cyclists, and cars—controlled by the CARLA traffic manager. All vehicles use autopilot, maintaining a 2.5-meter gap and a speed limit of 30 m/s.

We also placed 100 pedestrians, with 80% using crosswalks. Training episodes terminate if: 1) the AV collides with another entity; 2) the AV reaches its destination; or 3) the episode surpasses 500 training steps.

TABLE I: Parameters Used in the Simulation

Parameter	Value
Learning rate (Actor and Critic)	0.0003
Episodes	2000
Batch Size	64
$\gamma$	0.99
Exploration Noise	0.1
Exploration Step	10000
Policy Update Frequency	2
Replay Memory Size	5000

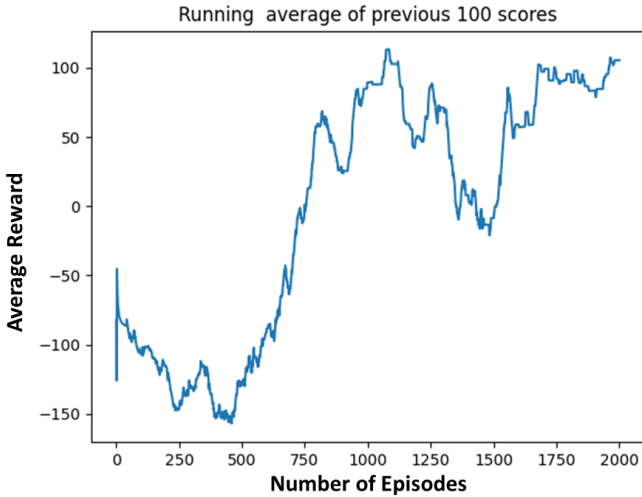


Fig. 3: Average Reward Over Episodes

## V. RESULTS & DISCUSSION

In this section, we present the results obtained using our proposed method. Figure 3 illustrates that the average reward progressively increases over episodes, ultimately attaining a high value at 2000 episodes. Furthermore, around the 2000th episode, the model stabilizes and converges, showcasing the efficacy of our approach.

During the testing phase, the T-intersection remains consistent with the training phase, though the traffic density varies. The density is determined by the random spawning of pedestrians (Ped) and other vehicles, such as cyclists and motorcycles (Veh), within the environment. We select five scenarios: 1)  $Ped = 100$  and  $Veh = 100$ ; 2)  $Ped = 200$  and  $Veh = 200$ ; 3)  $Ped = 300$  and  $Veh = 300$ ; 4)  $Ped = 400$  and  $Veh = 400$ ; 5)  $Ped = 450$  and  $Veh = 450$ .

The policy network trained by the model governs the vehicle as it navigates its environment and reaches its destination during the testing phase. To evaluate the model, we execute ten episodes and measure the travel delay and number of collisions in each episode. Since the model's objective is to minimize travel delay and accidents in dense traffic, we compute the average of both metrics. We repeat this process ten times and

establish the confidence interval. The test results, depicted in Figures 4 and 5, demonstrate that the vehicle quickly arrives at its destination and avoids collisions. Our model exhibits a distinct advantage in evading collisions with road participants, as the average number of collisions remains low.

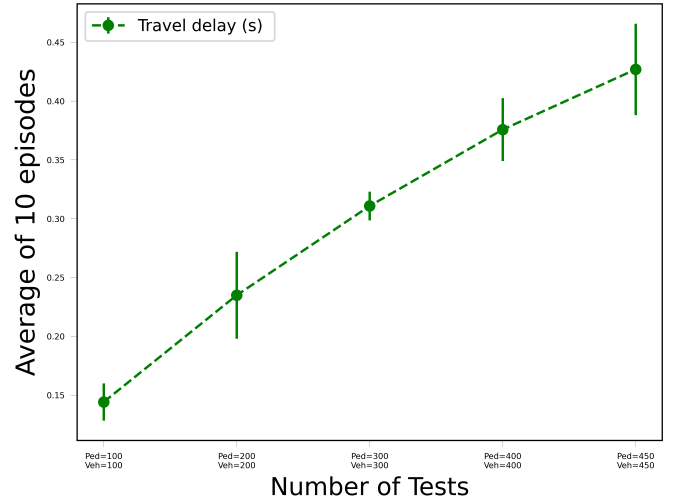


Fig. 4: Average Travel Delay for Each Test

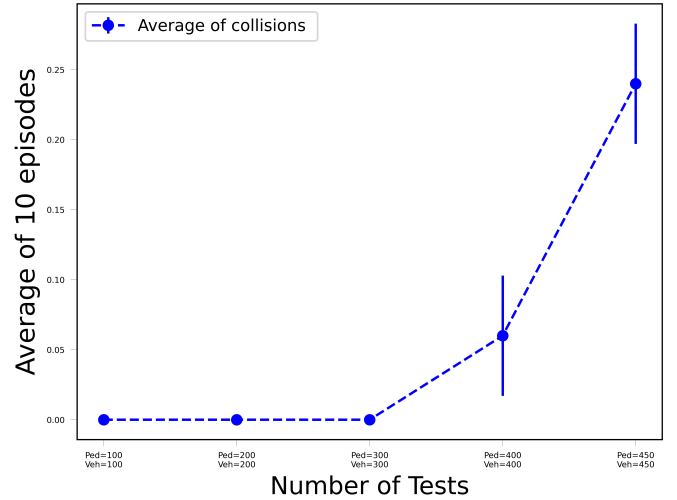


Fig. 5: Average Number of Collisions for Each Test

## VI. CONCLUSION

In this paper, we presented a single-agent approach for navigating T-intersections using Twin Delayed Deep Deterministic Policy Gradient (TD3) in dense traffic. Our method uses reinforcement learning to train an autonomous vehicle (AV) for real-time decisions. We used the CARLA platform to simulate a realistic urban environment with diverse traffic and challenging conditions. Our results indicate that the TD3 algorithm demonstrates stable convergence and improved exploration capabilities, enabling the AV to navigate T-intersections safely and effectively. Furthermore, the proposed

method exhibits a low number of collisions and reduced travel delays in various traffic density scenarios, highlighting its potential for real-world autonomous driving applications. In future work, we aim to integrate additional sensors, such as LIDAR and RADAR, to enhance the AV's perception. We plan to extend our research to complex driving scenarios, investigate other advanced reinforcement learning algorithms for improved performance, and explore the impact of different network architectures and hyperparameters on the proposed method's performance.

## REFERENCES

- [1] V. Milanés, J. Pérez, E. Onieva, and C. González, "Controller for urban intersections based on wireless communications and fuzzy logic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 243–248, 2009.
- [2] J. Huang and H.-S. Tan, "A low-order dgps-based vehicle positioning system under urban environment," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 5, pp. 567–575, 2006.
- [3] B. B. Elallid, N. Benamar, A. S. Hafid, T. Rachidi, and N. Mrani, "A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving," *Journal of King Saud University-Computer and Information Sciences*, 2022.
- [4] B. B. Elallid, S. E. Hamdani, N. Benamar, and N. Mrani, "Deep learning-based modeling of pedestrian perception and decision-making in refuge island for autonomous driving," in *Computational Intelligence in Recent Communication Networks*, pp. 135–146, Springer, 2022.
- [5] B. B. Elallid, N. Benamar, N. Mrani, and T. Rachidi, "Dqn-based reinforcement learning for vehicle control of autonomous vehicles interacting with pedestrians," in *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 489–493, IEEE, 2022.
- [6] S. Azimi, G. Bhatia, R. Rajkumar, and P. Mudalige, "Reliable intersection protocols using vehicular networks," in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, pp. 1–10, 2013.
- [7] L. Li and F.-Y. Wang, "Cooperative driving at blind crossings using intervehicle communication," *IEEE Transactions on Vehicular technology*, vol. 55, no. 6, pp. 1712–1724, 2006.
- [8] L. Riegger, M. Carlander, N. Lidander, N. Murgovski, and J. Sjöberg, "Centralized mpc for autonomous intersection crossing," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, pp. 1372–1377, IEEE, 2016.
- [9] Y. Chen, J. Zha, and J. Wang, "An autonomous t-intersection driving strategy considering oncoming vehicles based on connected vehicle technology," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2779–2790, 2019.
- [10] M. S. Corporation, "Carsim." <https://www.carsim.com>, 2023. Accessed: [March 15, 2023].
- [11] A. I. M. Medina, N. Van de Wouw, and H. Nijmeijer, "Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles," in *2015 IEEE 18th international conference on intelligent transportation systems*, pp. 1696–1701, IEEE, 2015.
- [12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
- [16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.
- [17] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.
- [18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.
- [19] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of sumo-simulation of urban mobility," *International journal on advances in systems and measurements*, vol. 5, no. 3&4, 2012.
- [20] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2149–2154, IEEE, 2004.
- [21] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, "SVL Simulator: A High Fidelity Simulator for Autonomous Driving," *arXiv e-prints*, p. arXiv:2005.03778, May 2020.