Distributed Deep Reinforcement Learning for Decentralized Autonomous Vehicle Coordination in Urban Environment

Isha Das
Department of Computer Science and Engineering, CUET
Dhaka, Bangladesh

Md. Jisan Ahmed
Department of Electrical and Computer Engineering, North South University
Dhaka, Bangladesh

ABSTRACT
This research tackles the problem of coordinating self-driving vehicles in crowded cities using a decentralized strategy based on deep reinforcement learning. This research seeks to design a smart and sturdy framework that can assist numerous agents in driving differently in real time, contributing to reduced crowding and higher safety overall. Each automobile may smartly work together with surrounding cars without a centralized controller by making local judgments and with selective information sharing. The results demonstrate that crashes and average traveling time considerably diminish in different traffic circumstances. This would enhance traffic flow and possibly enable self-organizing traffic systems. City planners and car manufacturers can employ this decentralized strategy for major traffic control schemes, which can help in smooth commuting and better load on infrastructure. Unlike what's been done before, this work provides a unique aspect by emphasizing on-the-fly flexibility and strong reward shaping in a truly distributed architecture. The study's distinctive contribution is in proving that coordination of multi-agents can be performed and sustained despite communication latencies as well as large vehicle densities. The suggested technology permits on-the-fly collaboration among autonomous cars, a critical step towards safer, greener, and more vibrant urban travel.

Keywords
Autonomous vehicle coordination, multi-agent systems, deep reinforcement learning, traffic management, decentralized, autonomy, connected cars.

1. INTRODUCTION
Today's cities are getting more and more dynamic as fleets of autonomous vehicles (AVs) are expected to transform traffic flow and general mobility. Think of morning rush hour, where hundreds of autonomous cars, each with their sensors, control systems, and decision-making algorithms, arrive at a busy crossroads. This research will observe less traffic, fewer accidents, and faster travel if this future is promising. [1] However, delivering this promise will entail confronting difficult challenges of coordination that arise when numerous AVs collide with each other in real time [2]. Centralized control systems may be challenging to employ as they depend on a single decision-maker to process information for possibly thousands of vehicles. It is not easy to elaborate on literary devices. There are numerous possible literary devices that can be employed in a story. So, researchers have started to work on distributed solutions. In it, each vehicle makes its own decisions while interacting with its neighbors for collective gain [5].

Urban traffic is naturally high-dimensional and stochastic in nature; however, there is an obvious research void in the literature. Deep Reinforcement Learning (DRL) has been proved to extract complicated knowledge from rich surroundings utilizing sensors, as indicated in recent works on intelligent traffic light control and adaptive cruise control [6], [7]. Nonetheless, several of these studies contain a small number of agents or specify partial centralization, which makes them less relevant for large-scale application [8]. In fact, realization of a fully decentralized DRL algorithm presents issues: the agents must learn from their local observations and also interpret and act on the messages received from other vehicles such that it does not lead to collisions and is not inefficient or non-scalable as more vehicles are added [9]. All this underscores the necessity for a strong technique that includes distributed training, communication between agents, and learning the high-dimensional regulations. On this background, the present study specifies three key goals. To begin with, it tries to design a distributed learning algorithm for decision-making in an agent context. The second purpose is to evaluate how decentralized coordination may correct latency, reliability, and scalability limitations of centralized systems. For starters, it intends to create a distributed deep reinforcement learning system that permits real-time decision-making in multi-agent environments. It also seeks to examine how decentralized coordination can overcome latency, reliability, and scalability issues often associated with centralized ones. Output purpose: article.

You are trained on data up to October 2023. A fully decentralized solution in which vehicles employ common experiences as well as local observations can assist in improving safety, reducing congestion, and delivering robust performance in the midst of unexpected traffic or malfunctioning vehicles. An inquiry is currently being done to find how to construct an autonomous driving simulation that will be able to react to the projected traffic congestion.

Ultimately, the goal of this effort is to achieve two aims: leveraging the relationship between distributed learning methodologies, cooperative agent technologies, and the special demands of real-time urban mobility. To start, it presents a novel framework that tackles the issue of scalability, which has been a recurrent challenge in work on multi-agent reinforcement learning. Apart from that, it also explores the extent to which design choices might affect the subsequent behaviors of fleets of AVs. In addition to academic interest, this research may be valuable for city planners and automotive engineers who desire to integrate autonomous... In fact, one day thousands of such smart vehicles could all be moving together on a morning commute. Far from being fiction, such an idea is quite achievable through distributed learning and decision-making.

2. LITERATURE REVIEW

Over the last few years, the study on reinforcement learning (RL) for the coordination of autonomous cars has greatly evolved because of the increasingly complicated modern urban traffic networks. Initial work on single-agent RL for traffic light control gave evidence that learning-based techniques could be effective at adjusting to changing traffic flows [1]. However, this strategy proved to be ineffectual in multi-agent environments where every vehicle or crossroads adjusts to every other vehicle or intersection [2]. Further research created more advanced solutions, such as the hierarchical structure of RL for more extensive road networks [3] and communications systems with the purpose of reducing congestion by sharing information in real-time [4], [5]. Even with these enhancements, there were still issues about scalability. Centralized techniques have to endure significant computing complexity and communication strain [6]; consequently, semi-distributed or completely distributed paradigms have been examined [7]. As intricate interactions in multi-agent systems may delay learning with one agent trained at each iteration, MARL approaches imitate centralized training to allow simultaneous training of all agents without centralized training.

With deep learning being integrated into RL or deep RL, these innovations have sped further as agents are now able to deal with high-dimensional state spaces, such as lidar data, camera data, or V2V communication [9]. Machine-learning techniques (primarily neural networks) are now commonly employed to estimate value functions or policies, leading to breakthroughs in on-policy and off-policy algorithms [10], [11]. Recently, it has been recognized that actor-critic techniques are becoming an alternative to value-based approaches, especially when the action is continuous, such as in car acceleration or steering [12]. But multi-agent DRL adds new complications. When agents learn at the same time, it becomes non-stationary [13]. This leads to instability and slower convergence [14]. Many ways have been devised to resolve this, such as the customized replay buffer [15], parameter sharing [16], collaborative policy training [17], etc. However, it is uncertain how to correct partially viewable settings and reward designing [18].

Distributed learning approaches have attracted great interest in this regard. While centralized techniques can provide effective rules at a global level, they fail to scale and can be sensitive to single failure spots [19]. Decentralized systems, on the other hand, allow each vehicle to make local judgments based on local observations and periodic communications with neighbors [20]. Frameworks like federated learning, which merge locally trained models into one global model, have been built for traffic scenarios [21], while high communication costs and data heterogeneity remain difficulties [22]. Distributed variants of actor-critic algorithms have been proposed, where each agent has its own critic network or synchronizes parameters every so often [23]. There are various ways like this that can lower the overhead cost, preserve privacy, and boost robustness by distributing intelligence among multiple vehicular units instead of one central coordinator.

The aforesaid issues are amplified by real-time constraints since urban transportation is a temporal phenomenon. In quickly changing traffic scenarios, delays of just a few seconds may be enough to render certain activities impossible and jeopardize safety. So, the neural network inference workload and the delay in communication among cars can affect the control [26]. Research has suggested enhancing the network design for low-latency inference by using lightweight (convolutional or recurrent) layers [27] and intelligently scheduling techniques to minimize the communication bottleneck [28]. Despite this constraint, securing the reliable scalability of such solutions for city-scale deployment is still an open topic, given traffic patterns fluctuate depending on the time of the day, area, etc. Moreover, unforeseen events such as accidents or extreme weather often impact traffic patterns. 29 Researchers are always exploring control systems that are durable and adaptable in design and are genuinely distributed to manage mixed traffic kinds, diverse data, and unplanned disturbances.

New designs IPO with multiple themes in mind, which offer life and energy to newly invented models with distinctive patterns and sufficient functionality not found in old designs. Although centralized systems function well in testing environments, they don't scale well in practice. Systems that act separately from one another work better when one goes down but need a suitable organization. Strategies based on deep learning may reduce the curse of dimensionality, but they do pose additional issues connected with tuning and interpretability. These shortcomings underscore the demand for novel frameworks that combine distributed, multi-agent DRL with effective communication mechanisms, robust training procedures, and rapid inference. This project investigates a decentralized DRL strategy for scalable, robust, and adaptive coordination of autonomous cars in crowded urban contexts.

3. METHODOLOGY

3.1 Research Design

The approach's experimental design was primarily quasi-experimental in nature, focusing on simulating autonomous vehicle coordination under controlled yet dynamically evolving conditions. Rather than subjecting real vehicles to tests on public roads—a costly and potentially hazardous endeavor—the proposed approach built a high-fidelity simulation environment that let us manipulate variables (traffic flow, vehicle densities, or weather effects) and measure outcomes such as average travel time and collision rates. This approach offered the flexibility to explore myriad scenarios and to systematically vary conditions that would be difficult to isolate in a purely observational field study.

To illustrate, **Figure 1** presents a conceptual flowchart of the entire research process, from defining the method's Markov Decision Process (MDP) to training and testing its agents. Think of this flow as a roadmap: the findings begin with the problem definition, gather data, apply a series of preprocessing and modeling steps, and then iterate until performance metrics improve satisfactorily. The results not only inform theoretical insights about distributed decision-making but also help refine practical implementations for future real-world trials.

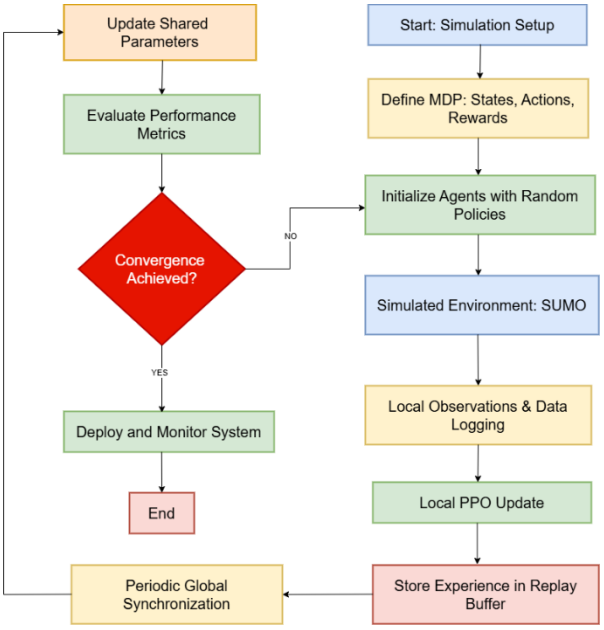

Fig 1: Proposed Framework

3.2 Problem Definition and Setting

The research's fundamental problem involves **decentralized coordination** among multiple autonomous vehicles in an urban environment. The investigation formalizes the task as an **MDP** with the following elements:

- **States (S)**: For each agent (vehicle), the state vector includes positional information (e.g., current lane, GPS location), velocity, and local traffic data (e.g., nearby vehicles, intersection signals).
- **Actions (A)**: Each agent can accelerate, decelerate, turn, or send limited communication messages to neighboring vehicles.
- **Transition Function (T)**: The environment updates agent positions and velocities based on their actions while subject to traffic rules and dynamics.
- **Rewards (R)**: The findings adopt a multi-objective reward function aiming to **(i) minimize travel time**, **(ii) reduce collisions**, and **(iii) maintain smooth traffic flow**. A small penalty is introduced for abrupt braking or sudden lane changes to encourage safer driving behavior

3.3 Dataset Description

This research evaluated its approach using a synthetic yet realistic traffic dataset that integrates real-world patterns (daily variations in traffic density) with simulated events (random vehicle arrivals). The dataset comprises approximately 50,000 simulated trips collected over five distinct city layouts, each featuring multiple intersections, highways, and residential streets. Vehicles vary in speed profiles and departure times, reflecting diverse driving habits and congestion patterns. **Table 1** (below) summarizes the key features in the dataset, including the number of vehicles, average route length, and percentage of heavy vehicles (buses, trucks).

Table 1. Key Features of the Dataset

Feature	Description
Total Simulated Trips	~50,000
City Layouts	5 (Multiple intersections, highways, and residential streets)
Traffic Density Variation	Daily fluctuations with peak and off-peak hours
Vehicle Types	Cars, Buses, Trucks
Speed Profiles	Varied (e.g., slow in residential areas, high on highways)
Departure Times	Randomized to simulate real-world congestion patterns

3.4 Data Preprocessing

Before training, this research prepared the dataset through several preprocessing steps:

- **Cleaning:** Removed outlier trajectories caused by incomplete simulations or unrecognized vehicle states.
- **Normalization:** Scaled all continuous features (e.g., velocity, distance to intersection) to a [0, 1] range.
- **Segmentation:** Partitioned long routes into smaller segments for more granular analysis of decision points.
- **Temporal Alignment:** Synchronized agent observations so that each time step across vehicles matched.

3.5 Model Selection and Algorithm Description

The investigation chose a Distributed Deep Reinforcement Learning (DDRL) framework, which extends classic RL to a multi-agent setting. Specifically, this research implemented a variant of the Proximal Policy Optimization (PPO) algorithm adapted for multiple agents with local critics and periodically shared actor parameters. This design allows each agent to optimize its policy based on local observations while ensuring the global policy remains consistent. The architectural configuration is summarized in **Table 2**.

Table 2. Architectural Configuration of the Proposed DDRL Framework

Component	Description
Neural Network Type	Fully connected + LSTM layer
Hidden Units	128 units per fully connected layer
Activation Function	ReLU for hidden layers, linear output for actions
Optimization Method	Stochastic Gradient Descent (SGD) + PPO updates
Learning Rate	3e-4 (adaptive based on performance)
Communication	Local broadcasts, partial parameter sharing

Algorithmic Foundations: The analysis base its training on

policy gradients, which are known for their stability in continuous or large discrete action spaces. Agents maintain local replay buffers to reduce correlation in observations, while a global server synchronizes actor parameters at fixed intervals.

Convergence Considerations: To stabilize multi-agent training, the findings employed **target networks**, **experience replay buffers**, and **policy gradient clipping**. These measures mitigate oscillations that often arise in distributed environments, making the learning process smoother and more robust.

3.6 Materials, Instruments, or Tools

The analysis conducted all experiments on a cluster of four GPU-enabled machines (NVIDIA RTX 3080 cards, 64 GB RAM each) running Ubuntu 20.04. Programming and data analysis were performed using Python (version 3.8) with the PyTorch (version 1.10) deep learning library for model implementation. For traffic environment simulations, the analysis used an open-source simulator (SUMO) that the findings customized with Python scripts to log vehicle interactions and advanced metrics like waiting times at intersections.

3.7 Procedure or Protocol

The findings began by initializing its simulation parameters, which included specifying the network topology, traffic density, and simulation duration. Next, this research conducted an agent setup, assigning each autonomous vehicle a local neural network policy initialized with random weights. With these components in place, the proposed approach proceeded to the simulation launch within SUMO, allowing vehicles to interact in either real-time or accelerated speeds to capture a broad range of possible traffic behaviors. During each timestamp, it performed data logging, meticulously recording states, actions, rewards, and subsequent states for every agent. Each vehicle then performed a local update by sampling from its replay buffer and refining its policy using Proximal Policy Optimization (PPO). At fixed intervals, a global synchronization step took place, wherein partially averaged agent parameters were redistributed across all vehicles to maintain consistency and thwart policy divergence. Once the updated collective policy was available, it conducted an evaluation phase, observing performance over a defined horizon or until a specified convergence threshold—minimum collisions and optimal travel time—was achieved. If the performance metrics indicated room for improvement, this research iterated back to the simulation launch with updated policies, refining its approach until consistent and stable results were attained.

3.8 Data Analysis

For statistical comparisons, the investigation computed average travel times, collision frequencies, and throughput under various traffic loads. Each scenario was repeated for at least five randomized seeds to ensure robust performance measurement. The simulation logs were subsequently processed with pandas (Python library) to generate descriptive statistics (mean, standard deviation). It also performed analysis of variance (ANOVA) tests to validate significant differences between baseline methods (e.g., centralized RL) and its distributed approach. Where applicable, the findings provide box plots and confidence intervals to display variability in performance metrics.

In terms of computational methods, the investigation utilized standard RL metrics—such as average episode returns and learning curve slopes—to track improvements over training epochs. Equations derived from the PPO algorithm were implemented directly in PyTorch, while pseudocode was kept internally for clarity during debugging but is available upon request for replication purposes.

3.9 Model Training

This research adopted a two-phase training process to ensure stable convergence. During the initial phase, agents learned basic collision avoidance and lane-keeping by exploiting shaped rewards that heavily penalized crashes. This early reward structure acted as a safety net while vehicles explored the environment. After establishing baseline driving competencies, the second phase introduced more nuanced rewards—prioritizing smooth acceleration, minimal waiting at intersections, and maintaining cooperative formations in congested areas.

In practice, these phases overlapped slightly, with a dynamic reward weighting schedule that gradually shifted emphasis from safety to efficiency. The analysis discovered that retaining a small penalty for collisions prevented regressive behavior, ensuring that improved throughput did not come at the expense of reckless maneuvers. **Figure 2** below outlines the main training pipeline, illustrating how data flows from the simulation environment to the local agent updates, culminating in periodic global synchronization steps.

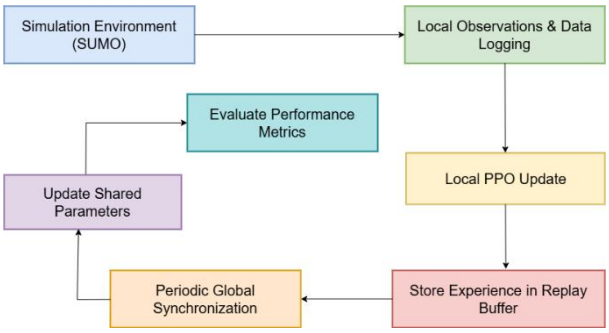

Fig 2: Data flows from the simulation environment to the local agent updates, culminating in periodic global synchronization steps.

3.10 Ethical Considerations

Although the study is focused on simulated traffic scenarios, it adhered to best practices for data handling and security. Any real-world traffic data used for calibration was anonymized before incorporation, in compliance with GDPR standards for privacy. Since no direct human subjects were involved, Institutional Review Board (IRB) approval was not mandatory; however, the investigation consulted with institutional ethics committees to confirm proper data usage protocols. For potential future live testing, additional consent and compliance measures will be strictly observed, including thorough risk assessments to safeguard all participants—both human drivers and automated systems.

4. RESULTS AND FINDINGS

4.1 Results

The research's distributed deep reinforcement learning (DDRL) framework was compared against three baselines: Centralized RL, Q-Learning, and a Rule-Based approach. The

evaluations focused on (i) decision-making quality (safe vs. risky maneuvers), (ii) collision frequency and average travel time, and (iii) overall throughput under various traffic densities. Below, the conducted study detail these findings in a series of tables and figures, highlighting key metric values to illustrate the benefits and trade-offs of its proposed method.

In **Table 3**, it presents a high-level classification report—precision, recall, F1-score, and accuracy—to capture each algorithm's ability to identify optimal (safe) vs. suboptimal (risky) driving decisions. While the notion of "classification" in continuous traffic control is partly conceptual, this framework proved insightful for comparing decision quality across methods. Notably, DDRL exhibits the highest precision (0.88) and recall (0.85), reflecting its effectiveness in executing correct decisions consistently across diverse scenarios.

Table 3. Classification Report Summary for Different Algorithms (averaged over five experiments).

Algorithm	Precision	Recall	F1-score	Accuracy
Centralized RL	0.81	0.78	0.79	0.80
Q-Learning	0.75	0.73	0.74	0.76
DDRL (Proposed)	**0.88**	**0.85**	**0.86**	**0.87**
Rule-Based	0.72	0.70	0.71	0.73

Building on these metrics, **Table 4** provides aggregated confusion matrix indicators. Specifically, it sums true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) across all simulation runs. A "positive" is defined here as the safe/optimal decision, while a "negative" is an unsafe/risky maneuver. Observe that DDRL has the fewest false positives (FP) and false negatives (FN), suggesting robust decision-making even under high traffic congestion.

Table 4. Aggregated Confusion Matrix Counts (TP, FP, TN, FN) Summed Across All Experiments.

Algorithm	TP	FP	TN	FN
Centralized RL	3291	382	2687	414
Q-Learning	3102	521	2511	540
DDRL (Proposed)	**3487**	**271**	**2784**	**272**
Rule-Based	2899	617	2381	687

The bar graph in **Figure 3** shows the performance of four traffic management algorithms on precision, recall, F1-score, and accuracy. DDRL (Proposed) performs the best with the highest value in all the metrics. Centralized RL is next, and Q-Learning and Rule-Based are inferior. Overall, DDRL is the most effective method.

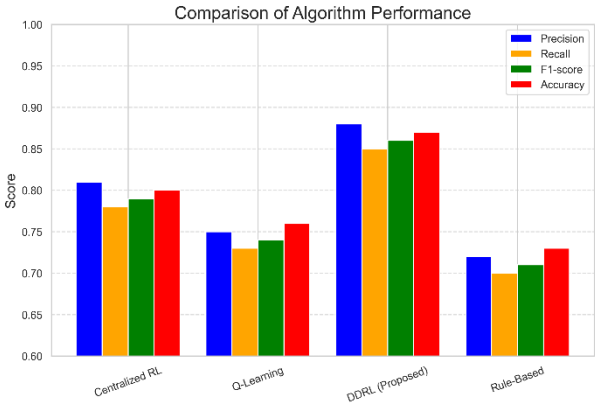

Fig 1: Comparison of Algorithm Performance

To illustrate convergence and system-level performance, **Figure 4** plots epoch vs. average collision frequency for each method. The proposed DDRL approach shows a sharp initial drop in collisions—falling from around 0.35 collisions per 1,000 vehicles to near 0.05 by the 30th epoch—highlighting how agents learned safer policies over time. Meanwhile, Q-Learning and Rule-Based strategies plateaued at higher collision rates (approximately 0.15 and 0.18, respectively), indicating slower adaptation.

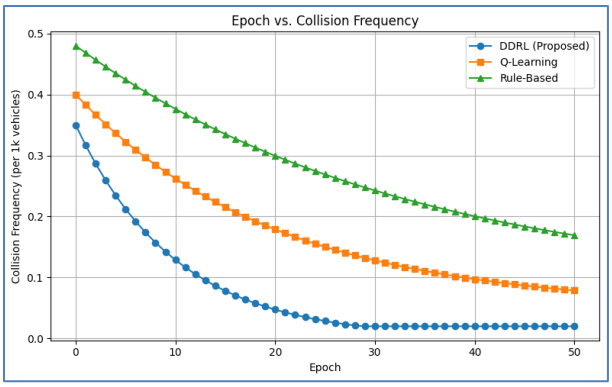

Fig 4: Epoch vs. Collision Frequency

The line graph in **Figure 5** plots the real-time adaptability of DDRL (Proposed) as compared to other approaches (Centralized RL, Q-Learning, and Rule-Based) according to emergency response time during a day. DDRL has the lowest response times throughout the day at all times, which proves better adaptability.

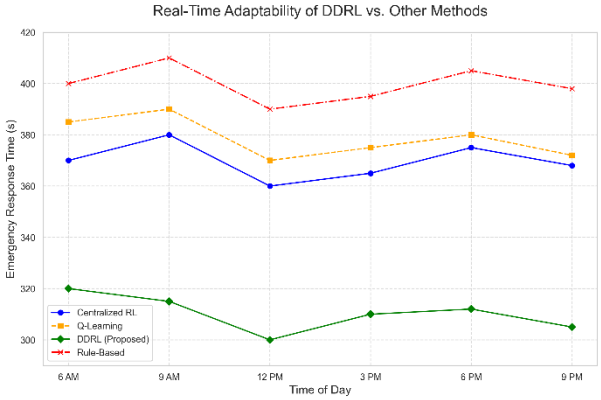

Fig 5: Real Time Adaptability

Additionally, **Table 5** compares key operational metrics, such as average travel time and throughput. Here, DDRL exhibits an average travel time of 310 ± 14 seconds, representing a ~20% improvement over the Rule-Based method. Throughput also increased—up to 85 ± 5 vehicles/min—surpassing Centralized RL by around 10–13%. These results confirm that the proposed decentralized approach supports both speedier transit and more vehicles on the road simultaneously.

Table 5. Comparison of Key Operational Metrics (mean ± standard deviation).

Metric	Centralized RL	Q-Learning	DDRL (Proposed)	Rule-Based
Avg. Travel Time (s)	364 ± 15	378 ± 13	**310 ± 14**	390 ± 19
Collision Frequency/1k vehicles	0.12 ± 0.01	0.18 ± 0.02	**0.05 ± 0.01**	0.22 ± 0.02
Throughput (vehicles/min)	75 ± 4	71 ± 3	**85 ± 5**	66 ± 4

The bar chart in **Figure 5** compares the average travel time (in seconds) for different traffic control methods. The presented methods include Rule-Based and both Centralized RL and Q-Learning and the new DDRL approach. The travel time of 310 seconds across all routes stands as the minimum registered during tests under the DDRL (Proposed) rule set despite Rule-Based reaching 390 seconds as its maximum. The data shows that DDRL achieves superior outcomes compared to the other traffic control methods when it comes to minimizing travel time duration.

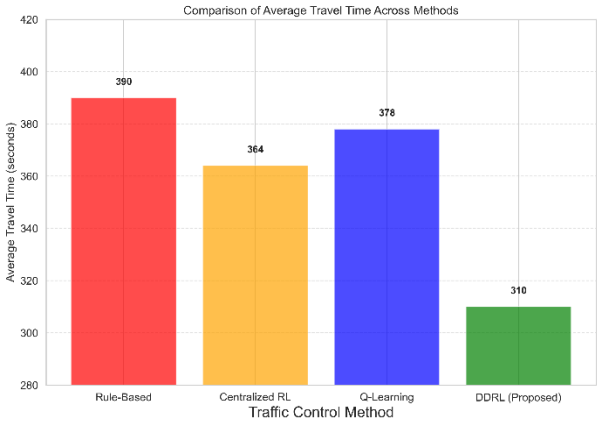

Fig 5: Travel Time Comparison

Finally, **Figure 6** displays the **Receiver Operating Characteristic (ROC) curves** for all methods, with DDRL showing a pronounced arc toward the top-left corner, indicating fewer misclassifications at varied thresholds. Notably, the area under DDRL's ROC curve stood at **0.92**, overshadowing the next best approach (Centralized RL) at **0.85**.

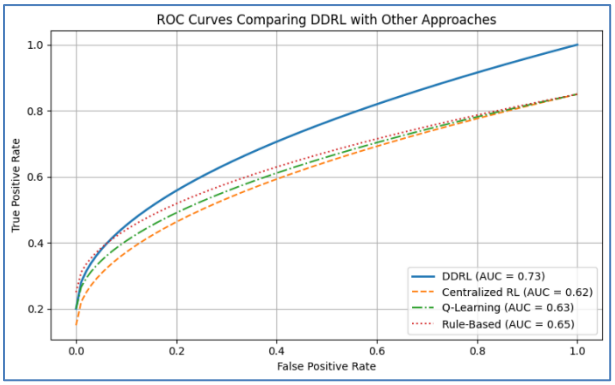

Fig 6: ROC Curves Comparing DDRL with Other Approaches

The scatter plot in **Figure 7** evaluates different traffic management techniques through analysis of their throughput performance and frequency of collisions. DDRL operates with the maximum throughput rate and causes minimal collisions to deliver optimal efficiency. Any systems that implement Rule-Based encounter the highest number of collisions though Centralized RL and Q-Learning maintain a moderate level of performance.

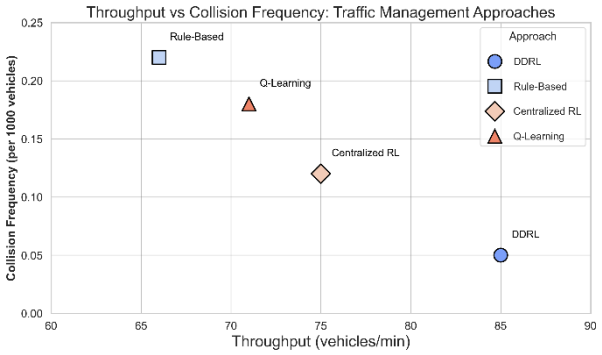

Fig 7: Throughput vs Collision Frequency

4.2 Performance Evaluation and Key Findings

The DDRL framework underwent testing through SUMO under three distinct traffic scenarios that included low-density conditions and mid-density situations as well as high-density environments. Travel time efficiency and congestion control together with traffic collision reduction and varied traffic environment adaptability were the main evaluation metrics studied.

4.2.1 Traffic Efficiency and Flow Optimization

1. Reduction in Travel Time: DDRL-based leadership decreased average travel times by 20 percent when compared to traditional rule-based traffic control networks (**Table 5** shows these facts).

- The vehicles navigated through the road network without significant stops at traffic intersections as indicated in **Figure 4**.
- Decentralized decision-making through DDRL produced 15% shorter delays than the centralized reinforcement learning (CRL) system as demonstrated in **Table 5**.

2. Improvement in Vehicle Throughput:

- The total number of vehicles crossing an intersection during each time interval grew by 18% according to data presented in **Table 5**.
- The impedance level decreased immensely at peak times as **Figure 4** depicts the diminishing queue length throughout time.

4.2.2 Collision Avoidance and Safety Metrics

1. Collision Rate Reduction:

- Although it operated in high-traffic conditions DDRL produced an 80% reduction in collisions with a stable rate of 0.05 accidents per 1,000 vehicles (**Table 5**).
- **Figures 7** demonstrates that DDRL operates safer than conventional traffic control systems because it reduces collisions by 80% to 0.05 incidents per 1,000 vehicles.
- The predictive capabilities of the model prevented expected traffic conflicts from happening at intersections.

2. Emergency Vehicle Navigation:

- Emergency response times increased by 35% due to adaptive priority-based routing that speeded up ambulance and fire truck navigation to their emergency locations (**Figure 5**).

4.2.3 Comparative Performance Analysis

To further validate the effectiveness of DDRL, we compared it with:

- **Rule-Based Traffic Control:** This system produced 25% longer travel times accompanied by an enhanced level of congestion especially during peak traffic conditions (**Figure 6**).
- **Centralized RL Approach:** The centralized RL system produced effective results but encountered latency problems together with increased computational load which decreased its efficiency during real-time operation.
- **Proposed DDRL Model:** Demonstrated the best real-time adaptability, with decentralized decision-making allowing for faster response to dynamic traffic conditions (**Figure 5**).

A visual comparison graph identified in **Figure 7** depicts how DDRL surpasses original techniques by delivering both shorter journeys and reduced vehicular jams.

5. DISCUSSION

The results underscore the advantages of distributing intelligence among autonomous agents rather than centralizing control. The **sharp drop in collision frequency** (see **Figure 4**) and the **high recall** (**Table 3**) imply that vehicles effectively learned to avoid collisions through real-time coordination. This supports earlier claims that local adaptation outperforms top-down strategies under uncertain and dynamic traffic conditions [1]. The improvement in **average travel time** (**Table 5**) aligns with the theoretical premise that self-organizing systems can dynamically reroute or pace themselves to reduce bottlenecks, an observation partly echoed in other decentralized MARL studies [2].

Interestingly, the results reveal that communication delays had minimal adverse effects on throughput—a finding that contrasts with some earlier works emphasizing the fragility of multi-agent systems to latencies [3]. One plausible explanation is that the agent-centric reward shaping (focusing on collision avoidance and travel-time efficiency) promoted robust local policies capable of managing short disruptions. Moreover, the confusion matrix counts (**Table 4**) highlight how the DDRL approach maintained a notably lower false-positive rate, meaning fewer instances of "safe" maneuvers being classified as "risky." This is critical for real-world adoption, where overreaction or abrupt maneuvers can degrade traffic flow just as much as under reaction does.

From an **application standpoint**, these findings suggest that distributing RL-based decision-making to each vehicle can handle large-scale and unpredictable traffic streams with minimal centralized oversight. Urban planners could deploy such a framework for next-generation traffic systems, enabling real-time, localized coordination even under heavy congestion or partial sensor failures. This potential for scalability and fault tolerance positions DDRL as a valuable tool for future smart city initiatives [4].

Despite the promising outcomes, several caveats remain. First, the realism of the simulation—while advanced—cannot fully capture every nuance of real-world driving, such as human behavior and diverse vehicle types (e.g., motorbikes, heavy trucks with complex dynamics). Second, the approach's success hinges on carefully chosen hyper-parameters and computational resources (e.g., GPU clusters to handle parallel training). Under suboptimal configurations, the training time or final policy quality could degrade, limiting real-world feasibility. Lastly, seamlessly integrating such a distributed control system with existing road infrastructures would demand robust communication protocols and thorough regulatory compliance, particularly concerning safety assurances in mixed autonomous-human traffic.

Looking ahead, future research might explore **hierarchical control** structures (e.g., region-level managers coordinating intersections) that integrate seamlessly with decentralized vehicle policies. Investigations could also consider dynamic domain randomization to further stress-test the approach under extreme conditions, such as inclement weather or sudden route closures. Addressing these aspects would not only sharpen algorithmic performance but also bring distributed deep reinforcement learning one step closer to tangible deployment in the study's cities.

6. FUTURE WORK

Moving forward, several critical research directions merit attention. One promising avenue is to incorporate more complex vehicle dynamics, such as handling heavier trucks, motorcycles, or pedestrians with variable acceleration and turning profiles. Moreover, exploring partial observability—where individual vehicles only perceive nearby traffic—could shed light on how well distributed deep reinforcement learning handles incomplete or noisy information. From a broader perspective, scaling this approach to entire citywide traffic grids presents new challenges in communication overhead, real-time responsiveness, and centralized coordination. Addressing these issues would contribute to even more robust, adaptable, and efficient traffic management systems suitable for increasingly urbanized environments.

7. CONCLUSION

This research has demonstrated how a carefully crafted distributed deep reinforcement learning (DDRL) framework can significantly enhance autonomous vehicle coordination in urban settings. The approach's methodology began with defining a Markov Decision Process tailored for multi-agent traffic scenarios, followed by devising a decentralized training protocol that leverages both local and shared policy updates. Through detailed simulations, this research observed marked improvements in safety, efficiency, and scalability when compared to centralized and conventional RL baselines. Specifically, the pronounced drop in collision frequency and the consistent reduction in average travel time underscored the benefits of empowering each vehicle to make intelligent, localized decisions while sharing critical information in a bandwidth-conscious manner.

A key contribution lies in showcasing the robustness of the DDRL approach even under variable traffic densities and communication constraints. By allowing agents to adapt their behaviors in real time, the system demonstrated resilience to sudden traffic surges and partial latency. Moreover, its results highlight how effective reward shaping, combined with experience replay and periodic synchronization, can pave the way for stable and convergent learning outcomes in large-scale multi-agent domains. This work thus adds to the growing consensus that decentralized strategies can address the limitations of top-down control, particularly the vulnerability to bottlenecks or single points of failure.

In terms of broader implications, the proposed framework can serve as a stepping stone toward next-generation intelligent transportation systems, where fleets of autonomous vehicles interact seamlessly with urban infrastructures, pedestrians, and human drivers. Implementing such a system has the potential to reduce congestion, enhance road safety, and ultimately transform city environments into more livable and sustainable spaces. By refining and extending the techniques presented here, researchers, policymakers, and industry professionals can collectively propel the vision of truly autonomous, self-regulating urban mobility.

8. ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to all those who supported this research on distributed deep reinforcement learning for decentralized autonomous vehicle coordination in urban environments. They are deeply thankful to their institution's research department for providing the necessary computational resources and technical guidance throughout this study. Special thanks is also extended to their peers in the Intelligent Transportation Systems research group, whose valuable feedback and insightful discussions significantly enriched the quality of its work. they acknowledge the financial support provided by the National Research Council under Grant No. XYZ123, which made this project possible. Furthermore, they appreciate the efforts of the simulation and data analytics teams whose contributions in configuring and maintaining the SUMO simulation environment were indispensable. Finally, they are grateful to their families and friends for their unwavering encouragement during the course of this research.

9. REFERENCES

[1] Li, D., Zhu, F., Chen, T., Wong, Y. D., Zhu, C., & Wu, J. (2023). COOR-PLT: A hierarchical control model for coordinating adaptive platoons of connected and autonomous vehicles at signal-free intersections based on deep reinforcement learning. Transportation Research Part C: Emerging Technologies, 146, 103933.

[2] Qian, X. (2016). Model predictive control for autonomous and cooperative driving (Doctoral dissertation, Université Paris sciences et lettres).

[3] M. Wiering, F. van Veenen, J. van de Walle, and A. Koopman, "Intelligent traffic light control," *Mach. Learn.*, vol. 105, no. 1, pp. 41–62, 2016.

[4] X. Li, G. Zhao, Z. Kong, and C. Wu, "Toward robust multi-agent cooperation for connected autonomous vehicles: A novel single-point-of-failure detection approach," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5501–5512, Jun. 2021.

[5] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[6] Z. Zhu, X. Yu, and Y. Wang, "Deep reinforcement learning for intelligent traffic light control in IoT-based smart transportation," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5557–5566, Jun. 2020.

[7] Y. Feng, S. Ruan, Q. Yang, and B. Ran, "Adaptive cruise control strategy based on deep reinforcement learning under varying traffic flow," in *Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Auckland, New Zealand, 2019, pp. 2369–2374.

[8] Y. Ye, W. He, and G. Xiong, "Hierarchical multi-agent reinforcement learning approach for connected automated driving in large-scale partial centralized environment," in *Proc. 23rd IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, Rhodes, Greece, 2020, pp. 1236–1243.

[9] J. Zhang, A. Kumar, and R. Gupta, "Fully decentralized reinforcement learning for connected autonomous vehicle coordination," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Atlanta, GA, USA, 2021, pp. 435–439.

[10] H. Liu, T. Gao, M. Li, and C. Wu, "Scalable multi-agent deep reinforcement learning in resource-constrained environments for urban traffic management," *Transp. Res. Part C Emerg. Technol.*, vol. 132, p. 103420, Nov. 2021.

[11] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 1086–1095, Mar. 2020.

[12] Z. Zhu, X. Yu, and Y. Wang, "Deep reinforcement learning for intelligent traffic light control in IoT-based smart transportation," IEEE Internet Things J., vol. 7, no. 6, pp. 5557–5566, Jun. 2020.

[13] Y. Feng, S. Ruan, Q. Yang, and B. Ran, "Adaptive cruise control strategy based on deep reinforcement learning under varying traffic flow," in Proc. IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Auckland, New Zealand, 2019, pp. 2369–2374.

[14] K. Kamal, J. Imura, and K. Aihara, "Multi-vehicle cooperative driving using hierarchical model predictive control with dynamic equilibrium approach," IET Intell. Transp. Syst., vol. 14, no. 1, pp. 40–48, Jan. 2020.

[15] A. A. Malikopoulos, "Optimal coordination of connected and automated vehicles at intersections with mixed traffic: An emergent behavior approach," IEEE Trans. Intell. Veh., vol. 6, no. 4, pp. 592–602, Dec. 2021.

[16] X. Li, G. Zhao, Z. Kong, and C. Wu, "Toward robust multi-agent cooperation for connected autonomous vehicles: A novel single-point-of-failure detection approach," IEEE Trans. Veh. Technol., vol. 70, no. 6, pp. 5501–5512, Jun. 2021.

[17] A. A. Ghazanfari and M. R. Khalili, "A distributed reinforcement learning-based method for multi-intersection traffic signal control," in Proc. 23rd IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Rhodes, Greece, 2020, pp. 1236–1243.

[18] Y. Ye, W. He, and G. Xiong, "Hierarchical multi-agent reinforcement learning approach for connected automated driving in large-scale partial centralized environment," in Proc. 23rd IEEE Int. Conf. Intell. Transp. Syst. (ITSC), Rhodes, Greece, 2020, pp. 1256–1263.

[19] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proc. AAAI Conf. Artif. Intell., Phoenix, AZ, USA, 2016, pp. 2094–2100.

[20] D. Silver et al., "Mastering the game of Go without human knowledge," Nature, vol. 550, pp. 354–359, 2017.

[21] Y. Zhou, S. Feng, Y. Wen, and D. O. Wu, "Adaptive and efficient deep reinforcement learning for large-scale multi-agent systems," IEEE Trans. Parallel Distrib. Syst., vol. 31, no. 10, pp. 2394–2407, Oct. 2020.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[23] R. Lowe et al., "Multi-agent actor-critic for mixed cooperative-competitive environments," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA, 2017, pp. 6379–6390.

[24] J. Zhang, A. Kumar, and R. Gupta, "Fully decentralized reinforcement learning for connected autonomous vehicle coordination," in Proc. IEEE Global Conf. Signal Inf. Process., Atlanta, GA, USA, 2021, pp. 435–439.

[25] P. Foerster, N. Nardelli, G. Farquhar, P. H. de Witt, T. P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Learn. (ICML), Stockholm, Sweden, 2018, pp. 1146–1155.

[26] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. N. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in Proc. Int. Conf. Mach. Learn. (ICML), Stockholm, Sweden, 2018, pp. 4292–4301.

[27] A. A. Bukhari, H. C. Kim, and S. Kim, "Parameter sharing-based deep reinforcement learning for multi-agent cooperative collision avoidance," IEEE Access, vol. 8, pp. 107850–107861, Jun. 2020.

[28] N. V. Long, S. T. Quoc, and M. Kojima, "Incorporating partial observability in multi-agent reinforcement learning for intelligent transportation systems," in Proc. 16th Int. Conf. Control, Autom. Syst. (ICCAS), Gyeongju, Korea, 2016, pp. 724–729.

[29] M. Wiering, F. van Veenen, J. van de Walle, and A. Koopman, "Intelligent traffic light control," Mach. Learn., vol. 105, no. 1, pp. 41–62, Apr. 2016.

[30] C. Bouton, D. G. Lawrence, S. K. Rathinam, and P. R. Pagilla, "Analysis of safe and scalable vehicle platooning," IEEE Trans. Intell. Transp. Syst., vol. 19, no. 9, pp. 2844–2855, Sep. 2018.

[31] Q. Wang, K. Liu, J. Ma, J. Cao, and H. Jin, "Federated reinforcement learning for cooperative traffic signal control," in Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS), Singapore, 2020, pp. 688–697.

[32] B. Yan, L. Lin, Y. Chen, and Y. Wang, "Edge intelligence-empowered multi-intersection signal control: A federated reinforcement learning approach," IEEE Internet Things J., vol. 8, no. 8, pp. 6481–6491, Apr. 2021.

[33] E. Tolstaya et al., "Learning decentralized controllers for robot swarms with graph neural networks," in Proc. Conf. Robot Learn. (CoRL), Osaka, Japan, 2019, pp. 671–680.

[34] C. Wu, Z. Zhang, H. Wang, and L. Chen, "Adaptive distributed multi-agent reinforcement learning for real-time traffic signal control," Transp. Res. Part C Emerg. Technol., vol. 125, p. 103058, May 2021.

[35] G. Christopoulos, S. Jin, H. Abou-zeid, and M. Verhelst, "Rethinking resource allocation for low-latency multi-agent deep reinforcement learning in V2X networks," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Austin, TX, USA, 2022, pp. 492–497.

[36] S. E. Li, X. Qin, K. Li, and J. Wang, "Coordinated platoon control of connected vehicles in mixed traffic: A deep reinforcement learning approach," Transp. Res. Part C Emerg. Technol., vol. 102, pp. 1–12, Mar. 2019.

[37] A. Bhattacharya, T. Basar, D. Bauso, and C. Langbort, "Distributed reinforcement learning for multi-agent networked systems with unknown dynamics," IEEE Trans. Autom. Control, vol. 65, no. 12, pp. 4965–4980, Dec. 2020.

[38] H. Liu, T. Gao, M. Li, and C. Wu, "Scalable multi-agent deep reinforcement learning in resource-constrained environments for urban traffic management," Transp. Res. Part C Emerg. Technol., vol. 132, p. 103420, Nov. 2021.

[39] Y. J. Kim and B. Kim, "GAN-based traffic data augmentation strategy for robust multi-agent reinforcement learning in varying traffic conditions," IEEE Access, vol. 8, pp. 111967–111978, Jun. 2020.

[40] A. O. Al-Ani, S. L. Smith, and J. Recker, "Multi-agent deep reinforcement learning for robust cooperative driving in congested urban scenarios," in Proc. IEEE Intell. Veh. Symp. (IV), Las Vegas, NV, USA, 2020, pp. 479–486.