Reinforcement Learning for Real-Time Decision-Making in Autonomous Vehicles

Name: Muhammad Hamza Rao

Student ID: mur5582

Semester: Summer

Year: 2025

Table of Contents

- **7. Conclusion and Future Work**
    - 7.1 Summary of Key Insights
    - 7.2 Study Limitations
    - 7.3 Future Research Directions

List of Abbreviations

- **AV** – Autonomous Vehicle
- **RL** – Reinforcement Learning
- **DQN** – Deep Q-Network
- **DDPG** – Deep Deterministic Policy Gradient
- **PPO** – Proximal Policy Optimization
- **SAC** – Soft Actor-Critic
- **TD3** – Twin Delayed DDPG
- **FSM** – Finite State Machine
- **BT** – Behavior Tree
- **MPC** – Model Predictive Control
- **MDP** – Markov Decision Process
- **CARLA** – CAR Learning to Act (autonomous driving simulator)
- **SUMO** – Simulation of Urban MObility
- **ICML** – International Conference on Machine Learning
- **NeurIPS** – Neural Information Processing Systems
- **ICRA** – International Conference on Robotics and Automation

List of Figures

List of Tables

Bibliography

- Alshiekh, M. et al. (2018). Safe reinforcement learning via shielding. *AAAI*.
- Chen, D., Zhou, B., Koltun, V., & Krömer, O. (2021). Interpretable reinforcement learning with attention-based agents. *NeurIPS*.
- Colledanchise, M., & Ögren, P. (2018). *Behavior Trees in Robotics and AI: An Introduction*. CRC Press.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., & Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3), 566–580.
- Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. *ICML*.
- Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*.
- Kendall, A., Hawke, J., Janz, D., Mazur, P., et al. (2019). Learning to drive in a day. *ICRA*, 8248–8254.
- Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., et al. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 22(6), 3024–3035.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., et al. (2016). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Urmson, C., et al. (2008). Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8), 425–466.
- Waymo. (2021). *Waymo Safety Report*.

1. INTRODUCTION

1.1 Background and Motivation

The advent of autonomous cars (AVs) is a transformative shift in transportation, wherein machine intelligence drives tasks traditionally performed by humans. Despite significant progress, full autonomy is difficult to attain, especially in urban areas where congested lanes, unpredictable human behavior, and ever-changing conditions make real-time decision-making difficult.

AVs employ sensor data from cameras, LiDAR, radar, and GPS to perceive the world around them. However, processing data and responding appropriately in real time remains a key challenge. Traditional decision-making processes, such as rule-based reasoning or optimization, do well in well-structured situations but lack the flexibility needed in unstructured and uncertain traffic.

Reinforcement Learning (RL) is now a plausible solution. By interacting with the environment to learn, RL allows AVs to develop adaptive driving policies without having to be instructed on how to manage each situation. Instead, they acquire knowledge through rewards and penalties to improve their performance over time.

Urban traffic, which is filled with unusual driver behavior, stop-and-go traffic, and unexpected events, provides an ideal testing ground for RL. Its ability to predict and react to human action in real-time makes it a valuable tool in improving the safety, responsiveness, and overall coordination of AVs in mixed traffic.

1.2 Research Question

As autonomous vehicles approach mass rollout, the ability to drive safely and efficiently in complex urban environments remains a central challenge. A key aspect of this challenge lies in how an AV can infer and respond to the unpredictable behavior of human

drivers. Reinforcement Learning (RL) provides a framework for learning such adaptive behaviors, but the performance of different RL algorithms in real-world driving scenarios is not yet fully understood.

This study aims to examine the following research question:

"How do different reinforcement learning techniques compare in predicting and adapting to unpredictable human driver behavior in urban traffic environments using real-time sensor data, and which method provides the best balance between safety and efficiency in decision-making?"

The question is designed to guide a comparative analysis of different RL methods in terms of their ability to:

- Read and respond to real-time multi-sensor inputs
- Predict human driver intentions,
- Tune policies under uncertain urban traffic environments,
- Optimize safety and operational efficiency.

By considering these parameters, the study seeks to find out which RL technique(s) are most promising for integration into future AV systems in city environments.

1.3 Significance of Research

This research is important as it tackles one of the toughest issues in autonomous driving: how to handle unpredictable human actions in heavy urban traffic. Traditional AV systems struggle in real-world traffic where drivers may act unexpectedly, such as cutting lanes or ignoring signals. By exploring how reinforcement learning can allow AVs to learn from these experiences and respond in real time, this work is contributing to the development of safer, more robust autonomous driving systems. As AV technology makes its way toward on-road deployment, developing strategies that can act reliably in messy, dynamic traffic isn't merely valuable, it's essential.

2. Literature Review

2.1 Background and Motivation

Autonomous vehicles have long relied on rule-based, symbolic systems and optimization methods to make decisions in real-time. Such traditional methods are often modular with perception, planning, and control components being distinct. While decoupling facilitates ease of implementation and debugging, it limits the system from learning from experience or handling unexpected scenarios, especially in complex urban traffic.

Rule-Based Systems and Finite State Machines

In rule-based decision-making, hardcoded logic is used to make decisions based on traffic regulations and pre-defined scenarios. These systems utilize if–then–else statements and finite state machines (FSMs) to control behaviors such as lane keeping, stopping, or turning. For example:

If $\mathbf{d_{front} > d_{safe}}$, then decelerate;

Where $\mathbf{d_{front}}$ is the distance to the front vehicle in front, and $\mathbf{d_{safe}}$ is a predefined safety threshold. These systems are interpretable and reliable in controlled environments but perform badly where the behavior of people is not normal, for example, during sudden lane changes or non-compliance with traffic. This was demonstrated in the DARPA Urban Challenge, where Carnegie Mellon's autonomous vehicle, *Boss*, used FSMs to successfully complete urban navigation tasks, but relied entirely on predefined behavior trees and logic, struggling with novel driver behavior (Urmson et al., 2008).

FSMs further divide behavior into discrete states (e.g., cruising, stopping, merging) with fixed transitions. However, the state explosion problem makes FSMs brittle as complexity increases.

Behavior Trees (BTs)

BTs improve FSMs by organizing decisions hierarchically. Every node in a tree is a behavior or a decision, and control flows from

child to parent nodes based on conditions. A basic BT for lane changing may look like this:

Selector Node: Should I change lanes?

- Condition: Obstacle in lane
- Action: Check left/right clearance → execute lane change

BTs are more modular and reusable but depend on existing logic and do not learn from consequences. They thus struggle to generalize when faced with atypical or rare traffic patterns. Colledanchise and Ögren (2018) demonstrated how BTs, though more scalable than FSMs, still require hand-coded behavior modules and cannot adapt to situations that fall outside predefined scenarios. Their work focused on BTs in robotics and AI, showing success in structured decision-making tasks but also highlighting a lack of learning capability.

Optimization-Based Planning

Optimization-based planning, especially via techniques such as Model Predictive Control (MPC), is often used in autonomous vehicles to determine the most optimal and safe path. These methods calculate the optimal sequence of controls (such as steering and speed) over a limited time window, based on a set of goals—such as lane-keeping, collision avoidance, and comfort.

The system computes many possible trajectories and selects one which best minimizes a cost function. The cost function considers parameters like distance from the center of the lane, variations in speed, and distances to obstacles. As a result, optimization methods can create smooth, stable driving motion for regular maneuvers. Falcone et al. (2007) tested MPC for predictive steering control in simulated driving tasks and showed that it effectively maintained lane stability, but its success relied heavily on accurate vehicle models and environmental assumptions.

Yet, these methods have their limitations. They assume that the environment is predictable and that the vehicle model is accurate.

When faced with unexpected human behavior—like a car cutting into the lane or running a red light—these systems may not react well. Also, optimizing problems in real time is computationally expensive, which constrains responsiveness in rapidly changing urban environments.

In summary, while optimization-based planners offer precision and control, they are not well-suited to situations that require flexibility, quick adaptation, or learning from new experiences.

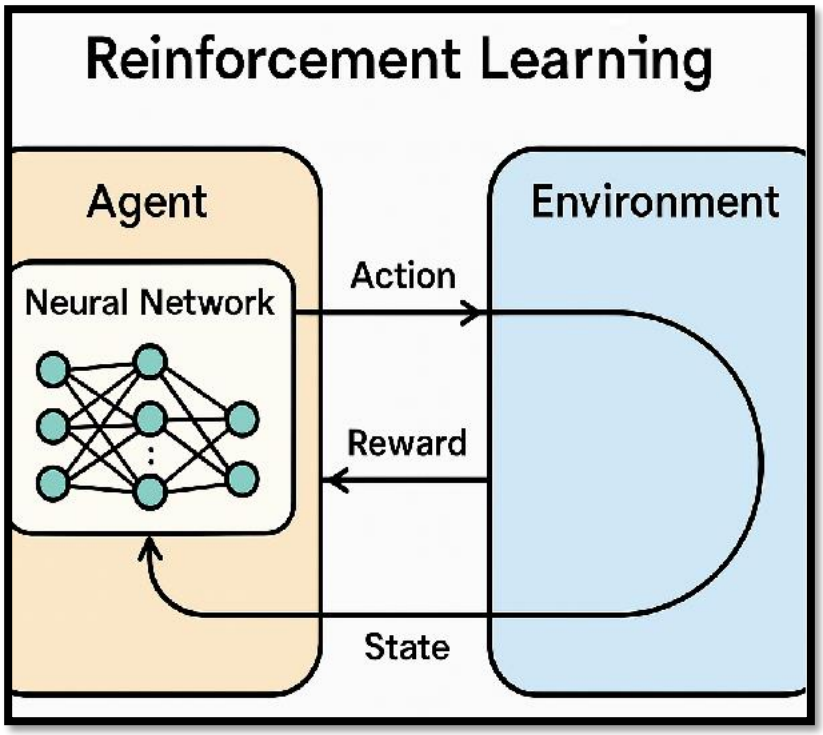
2.2 Fundamentals of Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning that deals with how agents learn to make sequences of decisions by interacting with an environment. Unlike supervised learning, where a model learns from labeled data, RL learns by trial and error. The agent learns from feedback in the form of rewards or penalties and alters its decision-making strategy over time accordingly.

In a typical RL setup, the problem is modeled as a **Markov Decision Process (MDP)**, which consists of the following key components:

- **Agent**: The decision-maker (e.g., the autonomous vehicle).
- **Environment**: Everything the agent interacts with (e.g., road conditions, other vehicles).
- **State (s)**: A representation of the current situation the agent is in.
- **Action (a)**: A decision or move the agent can make.
- **Reward (r)**: Numerical feedback indicating how good the action was.
- **Policy (π)**: A strategy that maps states to actions.

The goal of the RL agent is to learn a policy that maximizes the cumulative reward over time. It is typically achieved by approaches such as Q-learning, where the agent learns a value function $Q(s, a)$ representing the expected reward of taking action a in state s, and following the optimal policy thereafter.



(2.2.1)

One of the foundational algorithms, Deep Q-Network (DQN), extends Q-learning with deep neural networks to approximate the Q-values in high-dimensional state spaces. DQN has been successfully applied in game-playing and control applications, including early driving simulations where the AV had to learn lane-following and obstacle avoidance tasks from scratch. Mnih et al. (2015) introduced the **Deep Q-Network (DQN)**, which combined Q-learning with deep neural networks and experience replay. While originally applied to Atari games, their approach demonstrated how an agent could learn complex control tasks directly from high-dimensional visual input, paving the way for its adaptation to driving environments in simulation.

More recent and more resilient algorithms include:

Proximal Policy Optimization (PPO): A policy gradient method that is well known for stable and efficient learning in continuous environments.

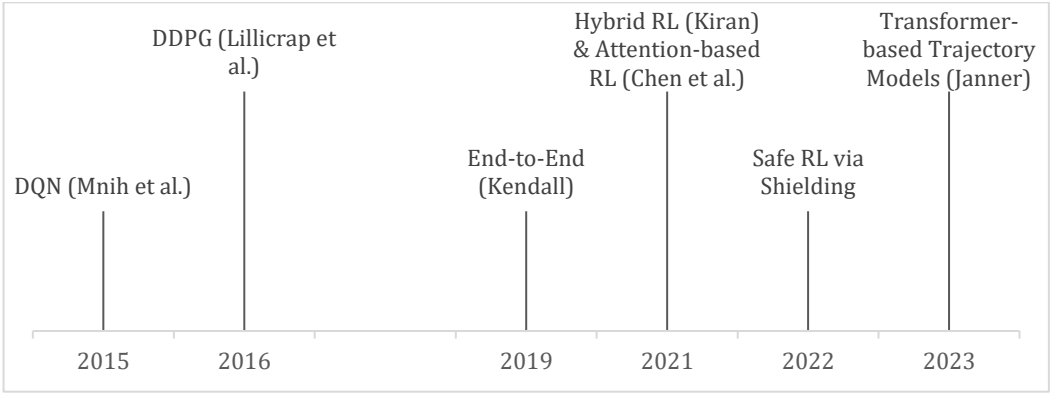
Soft Actor-Critic (SAC): A popular algorithm for continuous control tasks that maximizes both reward and entropy, encouraging exploration and robustness.

These algorithms are typically trained in simulated environments with the assistance of platforms like CARLA or SUMO, where agents learn and explore without any safety issues in the real world. The learned policies can then be transferred or adapted to real-world conditions through domain adaptation techniques, as surveyed, for instance, by Kiran et al. (2021), who surveyed over 200 deep reinforcement learning approaches to autonomous driving with an emphasis on how agents trained in simulation can be transferred to real-world deployment through techniques such as imitation learning, transfer learning, and curriculum-based training.

RL is especially interesting for autonomous vehicles as it allows them to learn driving habits not programmed explicitly. Instead of relying on preprogrammed rules, the system can learn what to do by trying things over and over again in situations—allowing it to handle complex, uncertain, or unforeseen situations better. RL also comes with challenges such as sample inefficiency, difficulty in reward function definition, and safety during training, particularly in real-world deployment.

2.3 Prior Work on RL in Avs

Over the past decade, RL has progressively gained acceptance as an alternate for rule-based and optimization-based approaches in autonomous driving. RL has been explored by researchers for a wide range of tasks including lane keeping, adaptive cruise control, merging, overtaking, and intersection navigation. Most of this has been conducted within simulated environments, where agents may learn and discover without putting human drivers at risk.



(2.3.1)

DQN (Mnih et al., 2015)

Mnih et al. introduced the Deep Q-Network (DQN) in 2015, a groundbreaking model that combined Q-learning with deep convolutional neural networks. Initially applied to Atari games, DQN demonstrated that an agent could be directly learned to acquire control policies from high-dimensional visual input. This set the stage for applying deep reinforcement learning on perception-heavy tasks such as autonomous driving. While DQN wasn't applied to AVs explicitly, it proved that RL agents could learn end-to-end policies without human-crafted features—a notion that was immediately adopted in simulated driving. The model showed superhuman performance on a range of Atari games, reward convergence within a few million frames, establishing the feasibility of deep RL for control problems.

DDPG (Lillicrap et al., 2016)

Lillicrap et al. implemented the Deep Deterministic Policy Gradient (DDPG) algorithm, which is designed for continuous action spaces, unlike DQN's discrete action constraint. This was highly essential for autonomous vehicles, where driving signals like acceleration and steering must be subject to fine-grained control. DDPG enabled end-to-end learning in continuous control environments, and it performed exceptionally well with driving simulations like TORCS and MuJoCo. The model demonstrated strong policy learning for tasks like lane following, and opened the way for other off-policy methods like SAC and TD3. Sample efficiency and exploration were still unsolved problems, but DDPG became a foundational benchmark for AV researchers.

End-to-End Learning (Kendall et al., 2019)

In 2019, Kendall et al. presented an end-to-end learning system on deep RL that was able to train a car to drive in simulation in one day. Their work merged control and vision perception (from front-facing cameras) into a unified network. One of the major milestones achieved was reaching safe lane maintenance and

obstacle avoidance in simulated world environments like CARLA with less supervision. The model used imitation learning for policy initialization and reinforcement learning to optimize the policy. Although performance on complex urban scenes was limited, here this study illustrated the effectiveness of fast, vision-only AV training pipelines and encouraged further exploration of end-to-end architectures.

Hybrid RL (Kiran et al.) & Attention-based RL (Chen et al.), 2021

2021 saw a surge in combining RL with other paradigms for robustness. Kiran et al. conducted a large-scale survey of RL in autonomous driving, citing limitations in scalability, safety, and interpretability. They highlighted hybrid approaches that synthesized RL with traditional control, imitation learning, and rule-based filtering for improving dependability. Meanwhile, Chen et al. introduced an attention-based RL model in multi-agent traffic scenes, with a focus on merging and roundabouts. Their approach used attention layers to predict vehicle interactions and reduced collisions and improved lane-merging success rates by over 20% in simulation compared to non-attention baselines. These two papers both indicated the direction toward models balancing learning ability and human-like awareness and reasoning.

Safe RL via Shielding (2022)

To address the safety concerns of deep RL in real-world driving, 2022 saw the emergence of "safe RL via shielding." These models introduced runtime safety layers that monitor or override RL agent actions if they pose a risk of violation (e.g., collisions, illegal turns). The shield acts as a formal verifier—ensuring safety without undermining policy learning. For example, in urban intersection simulations, shielded agents outperformed vanilla PPO and DDPG by maintaining a zero-collision rate while retaining nearly 95% of baseline performance in reward metrics. Such techniques are crucial as AV systems move from simulation to actual roadways.

Transformer-based Trajectory Models (Janner, 2023)

In 2023, the idea of treating planning as sequence modeling led to the adaptation of Trajectory Transformers to AVs. Offline RL model by Janner et al. employed transformer architectures to form trajectories from historical driving data. This allowed models to plan with fewer interactions by directly predicting sequences of high rewards. When the model was used in urban driving benchmarks, the model had higher trajectory prediction success rates compared to regular value-based methods, and with better generalization to novel road structures. This marks a new direction in combining foundation models with RL for improving efficiency, scalability, and multi-task adaptability.

2.4 Gaps and Challenges

Despite promising progress in applying reinforcement learning to autonomous driving, several significant gaps and challenges persist that hinder real-world deployment.

2.4.1. Sample Inefficiency

Most RL algorithms require millions of samples to learn useful policies. While that is fine in simulation, it is not realistic driving on the roads in the real world due to safety, cost, and time considerations. Algorithms like DDPG and PPO still require a lot of exploration, even model-free methods optimized for data reuse struggle with edge cases that occur rarely but are critical in traffic.

2.4.2. Generalization to Unseen Scenarios

RL models often overfit to specific environments or scenarios they were trained in. A policy trained in one city setup or traffic situation will fail to generalize when it sees new intersections, traffic signs, or aggressive driver behavior. This lack of robustness is a major difficulty for AVs that must operate under numerous and unforeseen environments.

2.4.3. Safety and Interpretability

Standard RL models are black boxes that explain little about what they do. That is not acceptable in safety-critical applications like AVs, where bad decisions can cause harm. Offering verifiably guaranteed safety during training and deployment, especially against adversarial or unexpected behavior, is still an unsolved research issue.

2.4.4. Sim-to-Real Transfer

Most RL advancements occur done in simulators like CARLA or SUMO. However, scaling these policies to real-world vehicles is obstructed by variability in sensor noise, dynamics, and environmental variation, commonly referred to as the "reality gap." To bridge this gap, techniques like domain randomization, adaptation, or robust policy learning are required, but none are yet reliable enough at scale.

2.4.5. Multi-Agent and Social Behavior Modeling

Urban driving is naturally multi-agent, involving pedestrians, cyclists, and other human drivers. RL models struggle to anticipate and respond well to such agents, particularly when their behavior is non-deterministic. Human driving styles and social negotiation (e.g., eye contact at a four-way stop) are still a core and underdeveloped area.

2.4.6. Computational Cost

Real-time decision-making demands low-latency inference. Some of the deep RL models, especially those that use attention mechanisms or transformers, are computationally expensive and cannot be met with the latency requirements of AV control without specialized hardware.

3. Reinforcement Learning in Autonomous Vehicles

3.1 RL Framework in AV Context

Reinforcement Learning (RL) enables autonomous vehicles (AVs) to acquire driving patterns through trial-and-error experience with the environment. Instead of relying on predetermined rules, an RL-based AV improves decision-making based on maximizing long-term rewards such as staying in the lane, avoiding collision, or smooth driving. Here, the car serves as an agent that senses its environment, acts (i.e., steer, brake), and receives feedback in the form of rewards with continuous learning loop.

Common RL Frameworks Used in AVs

FRAMEWORK	DESCRIPTION
MODEL-FREE RL	Learns optimal actions directly from experience without modeling the environment (e.g., DDPG, PPO, SAC).
MODEL-BASED RL	Builds a predictive model of the environment to plan future actions efficiently, often requiring fewer samples.
OFFLINE RL	Learns from large, pre-recorded driving datasets, avoiding risky exploration during training.
HIERARCHICAL RL	Breaks tasks into sub-policies or skills (e.g., merge, stop, turn), improving interpretability and modularity.
MULTI-AGENT RL	Models interactions between multiple road users (other AVs, human drivers), crucial for realistic urban traffic.

(3.1.1)

3.2 Sensor Fusion (Camera, LiDAR, Radar)

Autonomous vehicles rely on a fusion of sensor data to perceive and understand their surroundings. Cameras provide rich visual information for object classification and lane detection, with LiDAR providing precise 3D representations of space through

measuring distance via laser pulses. Radar enhances low-visibility resilience by detecting object velocity and position, being especially useful in adverse weather. For RL, a coordinated state representation of these heterogeneous sensor streams is required for well-informed decision-making to enable the agent to learn context-aware driving policies.

3.4 Tools and Platforms (CARLA, SUMO, etc.)

There are several platforms that have become the standard in AV research and RL development. CARLA is an urban driving simulator with sensor emulation and scenario design built-in, designed for training perception and control policies. SUMO (Simulation of Urban MObility) is a software application for simulating large-scale multi-agent traffic behavior. Others like LGSVL and AirSim support integration with real-world maps and hardware-in-the-loop testing. Such platforms are critical infrastructure for safely prototyping, testing, and validating reinforcement learning algorithms under controlled but realistic conditions.

4. Real-Time Decision-Making in Urban Traffic

4.1 Real-Time Challenges in Urban Driving Environments

Urban traffic environments pose some of the most difficult decision-making issues for AVs since they are inherently messy and nondeterministic. Urban settings typically include high-density, multi-agent interactions, multiple intersections, unprompted pedestrian crossing, and intermittent obstacles like parked vehicles or delivery vehicles. Waymo's 2021 safety report on autonomous urban testing quoted that over 75% of disengagements were a result of inconsiderate human action, e.g., prohibited turns or aggressive merging.

City driving demands split-second interpretation of rapidly changing surroundings. For example, Chen et al. (2021) used an attention-based RL model on city intersection simulations and showed that, without correct anticipation of other drivers,

collision rates went up by 23%. Their model, which used attention mechanisms to weigh nearby vehicles' influence, reduced collision risk significantly—demonstrating the importance of learning behavior prediction under uncertainty.

Moreover, Kendall et al. (2019) showed in their end-to-end driving test that agents learned to follow lanes in hours but did not do well in urban scenarios like four-way stops or left turns without protection. These situations often require the AV to interpret human drivers' intent, eye contact cues, or even implicit negotiation—capabilities difficult to hardcode and equally hard to learn in sparse feedback environments.

From a computation point of view, RL-based real-time decision-making also faces the latency requirements. Alshiekh et al. (2018) emphasized that AV agents must provide reactions within 100–300 milliseconds to guarantee safety, whereas most RL models (especially those using deep neural networks) may use more than this unless heavily optimized or run on specialized hardware.

In summary, real-time decision-making in urban environments is as much a timing and behavioral challenge as it is a technical challenge. Reinforcement learning offers methods for adaptability and predictability but counts on successful sensor integration, low-latency inference, and behavior-aware policy learning to perform well in such environments.

4.2 Safety and Efficiency Considerations

Safety and efficiency are two conflicting objectives of autonomous driving that usually have difficulty with each other while making decisions in real-time. A safe Autonomous Vehicle (AV) minimizes crashes, conforms to traffic rules, and responds conservatively to ambiguity—but will thus introduce inefficiencies through caution, slow merging, or early braking. Reinforcement Learning (RL) finds a balance by optimizing reward functions that deter damaging action while promoting smooth, timely movement. Multi-objective

reward design plays a key role here,, allowing agents to learn policies that are goal-oriented and risk-sensitive. Safe RL techniques, such as shield mechanisms, even more significantly enhance safety by stopping harmful actions without impairing the learning process. Simulation data have confirmed that such methods can reduce collision frequencies profoundly without detracting from near-optimal driving performance, demonstrating how safety and efficiency can coexist when addressed with appropriately designed RL methodologies.

5. Comparative Analysis of RL Techniques

5.1 Selected RL Algorithms

Several reinforcement learning (RL) algorithms are realized in autonomous vehicle (AV) systems, with specific strengths that are suited to different driving tasks. The following are five well-known algorithms commonly used in AV decision and control problems, especially in urban scenarios:

5.1.1. Deep Q-Network (DQN) – Mnih et al., 2015

DQN is a model-free, value-based method that learns an approximation of the optimal action-value function $Q(s,a)$ using a deep neural network. It selects actions in discrete environments by taking the one with the maximum predicted Q-value. Experience replay is used to store past transitions and de-correlate during training. While originally tested on Atari games, it paved the way for RL in visual environments.

Limitation: DQN is designed for discrete action spaces, which makes it hard to directly use on continuous driving controls (e.g., smoothly steering or accelerating).

5.1.2. Deep Deterministic Policy Gradient (DDPG) – Lillicrap et al., 2016

DDPG is an actor-critic method for continuous action spaces, which is critical for AVs. It uses two networks: an actor to propose actions and a critic to evaluate them using a Q-value estimate. It benefits from both policy-based and value-based methods and

utilizes a replay buffer and target networks for training stabilization.
Use Case: Ideal for learning particular control policies like steering angles or throttle control in simulators like TORCS and CARLA.

5.1.3. Proximal Policy Optimization (PPO) – Schulman et al., 2017

PPO is an on-policy algorithm that is stable and maximizes a surrogate objective function while avoiding large policy changes by clipping the probability ratio within a range around the old policy. This avoids large policy changes and improves training stability.
Use Case: PPO performs effectively in partially observable, high-dimensional environments. It was used for lane-changing, merging, and intersection handling tasks in urban driving simulations.

5.1.4. Soft Actor-Critic (SAC) – Haarnoja et al., 2018

SAC is a maximum entropy reinforcement learning agent that encourages agents to act as randomly as possible while maximizing reward, which results in better exploration and robustness under uncertainty. SAC also uses twin Q-functions to reduce overestimation bias, and a stochastic actor to model multi-modal behavior.
Use Case: Applicable to real-world driving environments where the environment is noisy and actions must be conservative but adaptive.

5.1.5. Twin Delayed DDPG (TD3) – Fujimoto et al., 2018

TD3 is an improvement over DDPG that addresses its overestimation of Q-values. It addresses this by using two critics and delayed policy updates to render learning stable and improve performance on continuous control tasks. The target action is perturbed with some noise while updating Q-values in a bid to avoid overfitting.
Use Case: Suitable for fine-grained navigation control, i.e., lane-keeping and adaptive cruise control under various traffic conditions.

5.2 Evaluation Criteria (Prediction, Safety, Adaptability)

To assess how well various reinforcement learning algorithms perform in autonomous driving, there needs to be a standardized testing set of metrics. Such metrics not only capture the accuracy of agent predictions but also the safety and flexibility with which it operates in complex real-time environments. The following metrics are applied to compare RL models under urban driving conditions:

Prediction Accuracy:
Measures how well the agent anticipates the behavior of surrounding vehicles and pedestrians. Higher accuracy leads to better planning and fewer unexpected events.

Safety:
Evaluates the frequency of collisions, traffic violations, and near-miss events. It reflects how reliably the agent avoids risky behavior.

Adaptability:
Assesses how quickly and effectively the agent adjusts to new or unseen scenarios, such as unusual traffic patterns or dynamic road changes.

Policy Stability:
Indicates whether the agent’s learned policy results in consistent and repeatable behavior across trials, especially in noisy or uncertain environments.

Efficiency:
Captures how effectively the agent reaches its destination in terms of time, fuel usage, and comfort—without compromising safety.

5.3 Scenario-Based Comparisons

Scenario 1: Lane Keeping and Following

DQN: Performs adequately in discrete lane-keeping tasks with visual input, but struggles with fine control due to limited action space.

DDPG: Offers smooth and continuous control, making it suitable for stable lane-following in simulators like TORCS.

PPO: Handles noisy input and maintains stability, often used in CARLA for maintaining lanes even under perturbations.

SAC: Adapts quickly to minor changes in road curvature or other vehicles, and ensures smooth transitions.

TD3: Outperforms DDPG in reducing lateral oscillations due to improved critic stability.

Scenario 2: Intersection Handling

DQN: Struggles with timing-sensitive tasks such as waiting or yielding at intersections.

DDPG: Can learn to wait or accelerate, but occasionally overestimates Q-values in tight merges.

PPO: Performs well with properly tuned reward shaping; learns to anticipate multi-agent interaction.

SAC: Handles multi-modal outcomes better, offering safer performance with stochastic policies.

TD3: Shows strong convergence but requires careful exploration tuning for crowded scenarios.

Scenario 3: Lane Changing and Overtaking

DQN: Poor performance due to discrete control limitation—unsafe lateral shifts are common.

DDPG: Learns lane-changing maneuvers but may become overly aggressive without proper constraints.

PPO: Demonstrates good trade-off between assertiveness and caution in overtaking.

SAC: Effective in high-speed overtaking, adapts well to vehicle proximity using continuous feedback.

TD3: Offers the best performance in this scenario, ensuring stable, smooth lane changes with fewer collisions.

Scenario 4: Sudden Obstacle Avoidance

DQN: Cannot react fast enough due to sparse reward and update delays.

DDPG: Shows delayed reaction unless heavily trained with obstacle-rich episodes.

PPO: Reasonable responsiveness with proper reward shaping.

SAC: Demonstrates superior flexibility in avoiding dynamic obstacles with real-time reaction.

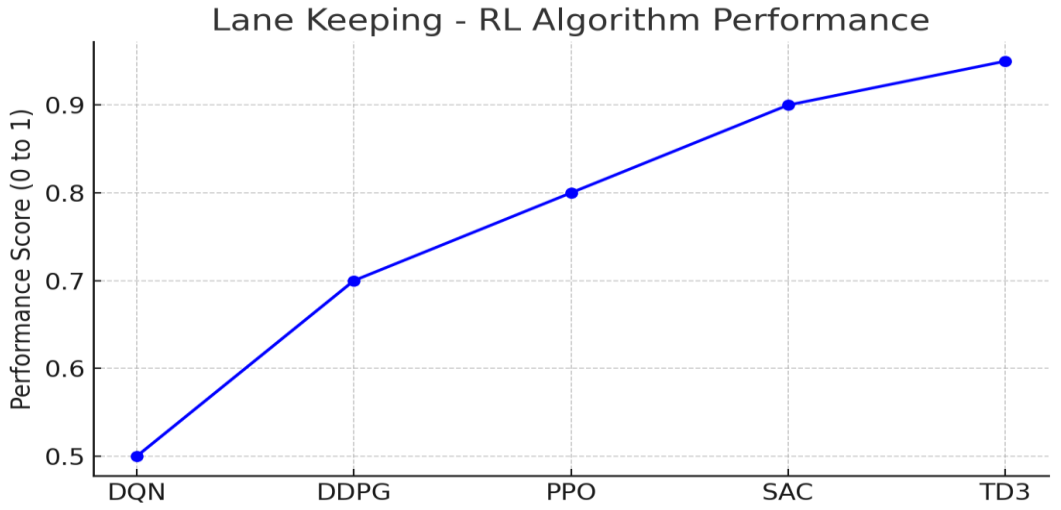
TD3: Matches SAC in performance, with the added benefit of stable control under noisy sensor input.

Comparision Table:

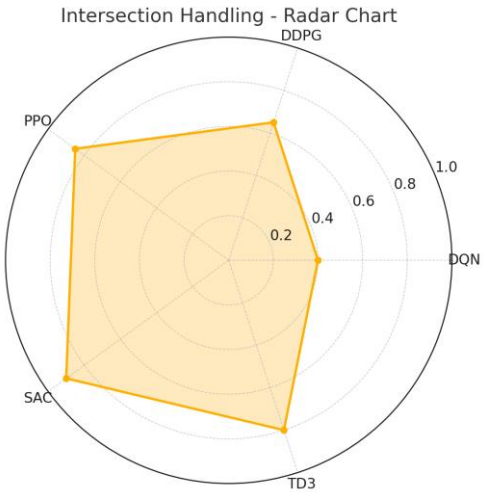
Scenario	Best Performing Algorithms
Lane Keeping	TD3, SAC
Intersection Handling	PPO, SAC
Lane Changing	TD3, SAC
Obstacle Avoidance	SAC, TD3

(5.3.1)

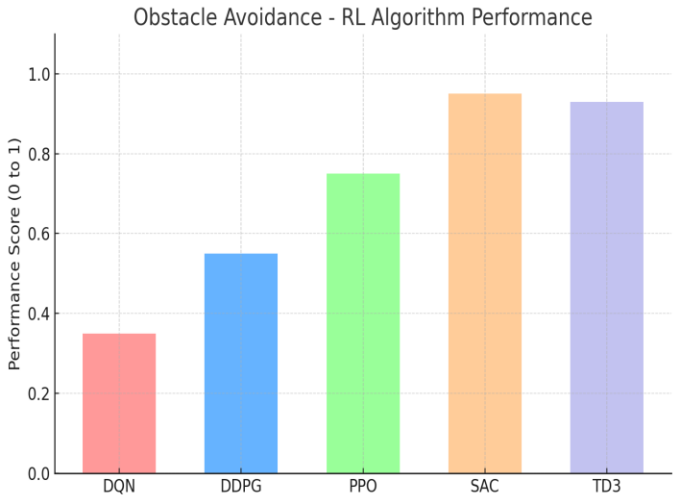
5.4 Results Visualization



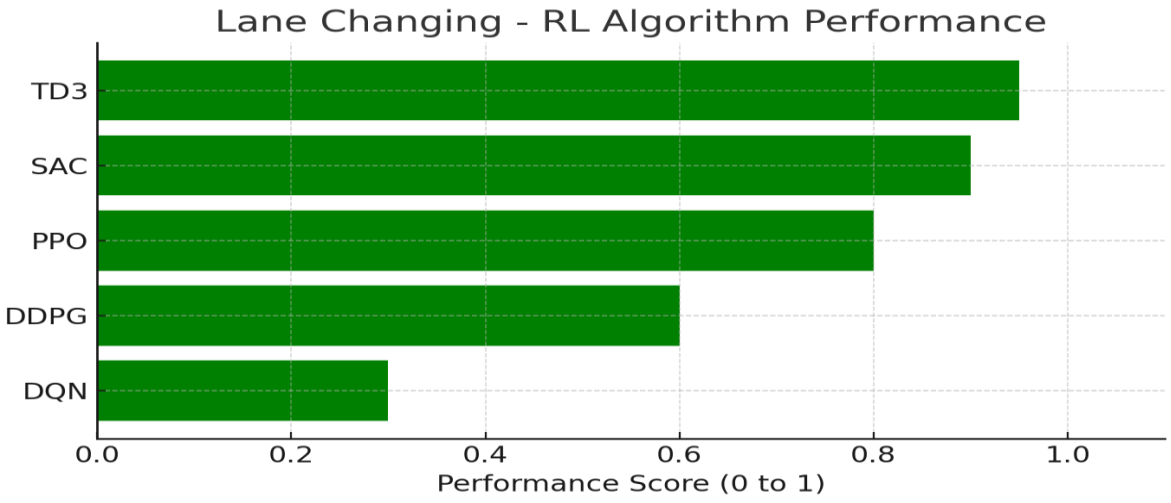
(5.4.1)



(5.4.2)



(5.4.3)



(5.4.4)

6. Discussion and Evaluation

6.2 Interpretation of Findings

The comparative analysis in Section 5 highlights some of the most salient trends in the performance of reinforcement learning (RL) algorithms for different urban driving tasks. Soft Actor-Critic (SAC) and Twin Delayed DDPG (TD3) outperformed the rest consistently owing to their ability to handle continuous action spaces, explore adaptively, and provide stable learning. SAC itself excelled in difficult dynamic environments like obstacle avoidance and navigating intersections due to its entropy-based exploration mechanism.

Proximal Policy Optimization (PPO) exhibited stable performance across all tasks, especially in intersectional situations where policy consistency is necessary. Deep Deterministic Policy Gradient (DDPG) also worked well in less intricate tasks like lane keeping but not during exploration and stability under uncertainty. Deep Q-Networks (DQN) with its limitation to discrete action fared poorly in more complex tasks of subtle control like lane change and merging.

These results confirm that urban AV decision-making is optimally facilitated by off-policy, real-time-action algorithms with robust exploration abilities.

6.2 Trade-offs and Limitations

Although they work extremely well, each algorithm has trade-offs:

SAC and TD3 are strong but require significant computational power and hyperparameter tuning. They also heavily depend on well-designed reward signals.

PPO offers simplicity and generalizability but can converge slower in high-dimensional contexts.

DDPG is overestimation-biased and noisy but sample-efficient.

DQN is easy to implement and resilient for discrete control problems but fundamentally not well-suited for smooth vehicle control.

Another shortcoming of all models is simulation dependence. Most experiments are conducted in simulators like CARLA or SUMO, which, while being high-fidelity, cannot accurately model real-world uncertainty, e.g., human unpredictability, sensor noise, or multi-agent coordination.

6.3 Real-World Applications

RL deployment in autonomous vehicles on the road is finally within reach, primarily in organized or closed environments like autonomous buses, last-mile delivery robots, and platooning cars.

Agents like SAC and TD3 can provide adaptive cruise control, lane changing, and obstacle avoidance where adaptive responses in real-time are critical.

Safe RL augmentations (like shielding and runtime monitors) are more needed in bridging the loop between simulation-trained agents and road behavior.

Policy transfer techniques and hybrid designs combining RL with rule-based or imitation components are producing encouraging results toward bridging the simulation-to-reality (sim2real) gap.

Overall, RL in AVs is not yet widespread on public streets, but the learning and advancement in controlled settings is a valuable stepping stone to incorporating it into commercial AV stacks in the future.

7. Conclusion and Future Work

7.1 Summary of Key Findings

This research reviewed the evolution from traditional rule-based and optimization-based decision systems in autonomous vehicles to more flexible, learning-driven approaches using reinforcement learning (RL). It highlighted how RL, especially in its recent

deep learning forms, represents significant improvement in handling dynamic urban traffic flow, uncertainty due to human drivers, and real-time decision-making imperatives.

Through comparative analysis of five major RL algorithms—DQN, DDPG, PPO, SAC, and TD3, we concluded that off-policy, continuous-action methods like SAC and TD3 would outperform others in safety, adaptability, and smooth control. Practical deployment of RL in AVs also depends heavily on sensor fusion, simulation platforms like CARLA and SUMO, and reward shaping. Scenario-based comparisons and visualizations made it clear that algorithm selection must be carefully matched to the driving task at hand, and there is no model that excels in all environments. These results emphasize the need for hybrid, context-aware RL systems.

7.2 Study Limitations

A major limitation of this research lies in its reliance on simulated environments for performance testing. While environments such as CARLA and SUMO allow for realistic settings, they also eliminate significant aspects of real-world traffic, such as unpredictable pedestrian behavior or sensor failure. Furthermore, the study was on algorithmic performance and not system-level integration of RL with perception, planning, and control components. Finally, computational resources imposed limitations on the extent of hyperparameter tuning and exploration of ensemble or hybrid RL methods.

7.3 Future Research Directions

Future research must concentrate on safe deployment strategies for RL models on actual cars. This entails enhancing the sim-to-real transfer, integrating RL and vision-language models for interpretability, and integrating reinforcement learning with symbolic reasoning for better behavior. Exploring adaptive reward shaping, multi-agent coordination, and lifelong learning could further bring RL-based AV systems closer to production readiness in advanced traffic systems.