Multi-Agent Reinforcement Learning for Autonomous Vehicle Coordination

Edoardo Conca

February 2025

Abstract

This study explores the application of multi-agent reinforcement learning (MARL) for optimizing autonomous vehicle (AV) behavior in unsignalized intersections shared with human-driven vehicles (HDVs). The main challenge is achieving real-time decision-making that balances safety, efficiency, and fairness in mixed autonomy scenarios. For this purpose, three RL models have been used: Proximal Policy Optimization (PPO), Advantage Actor-Critic (A3C), and Deep Deterministic Policy Gradient (DDPG) in conjunction with the Simulation of Urban Mobility (SUMO) framework, leveraging the Flow library for seamless RL-SUMO integration. The study investigates different control strategies and their impact on traffic coordination, demonstrating how MARL can lead to safer and more efficient intersection management.

1 Introduction

The advancement of autonomous driving technology aims to enhance the driving experience by improving safety, efficiency, and overall convenience. To achieve a high level of automation, various cutting-edge technologies must be integrated. With the rapid progress in artificial intelligence (AI), several automotive companies have been developing autonomous vehicles capable of making complex driving decisions. Among AI-driven approaches, deep reinforcement learning (RL) has gained significant attention as a viable method to optimize decision-making driving systems. However, designing a controller that enables real-time decision-making remains a critical challenge.

Decision-making systems in autonomous vehicles play a key role in determining driving behaviors to navigate complex environments successfully.s While highway and urban driving present unique challenges, intersections are particularly difficult due to the presence of multiple road users, including human-driven vehicles, pedestrians, and cyclists. Managing traffic flow efficiently at unsignalized intersections is especially challenging, as it requires dynamic coordination between autonomous and human-driven vehicles. Addressing this issue, **multi-agent reinforcement learning (MARL)** has emerged as a promising approach to enable cooperative and adaptive driving strategies in mixed traffic scenarios. Various studies have explored solutions to intersection management, but ensuring safe and efficient navigation without traffic signals remains an open research problem.

This project investigates the use of multi-agent reinforcement learning to control autonomous vehicles within a mixed unsignalized intersection scenario, where both self-driving and human-driven vehicles interact. To tackle this challenge, three reinforcement learning algorithms are employed in conjunction with Simulation of Urban Mobility (SUMO) (10) to simulate realistic traffic dynamics. Additionally, the Flow library (8), a RL framework for traffic control and optimization, is utilized as an interface between reinforcement learning and SUMO, providing a structured framework for defining traffic scenarios, managing vehicle behaviors, and optimizing agent control policies. By leveraging Flow, the implementation and evaluation of learning-based traffic control strategies become more modular and scalable. Through this approach, the study aims to enhance coordination strategies, ensuring safer and more efficient traffic management in mixed autonomy settings.

2 Related Work

The coordination of Connected Autonomous Vehicles (CAVs) at unsignalized intersections has been extensively studied using various control and learning-based approaches. Traditional methods such as optimization-based and rule-based techniques have been widely explored but often struggle with real-time adaptability and scalability. More recently, deep reinforcement learning (DRL) and multi-agent reinforcement learning (MARL) have emerged as promising solutions to address these limitations.

Several studies have investigated multi-agent reinforcement learning (MARL) frameworks for autonomous driving at intersections. One approach formulates the problem as a Partially Observable Stochastic Game (POSG) and introduces the Cooperative Multi-Agent Proximal Policy Optimization (CMAPPO) algorithm, demonstrating superior efficiency and safety in coordinating CAVs compared to Independent PPO (IPPO) (6). Similarly, another study proposes a multi-agent TD3-based framework for handling highway merging scenarios, effectively minimizing stop-and-go traffic while ensuring smooth vehicle coordination (4).

Other researchers have explored heuristic-guided MARL approaches. A heuristic-based QMIX (H-QMIX) framework was developed to integrate predefined rules with DRL, ensuring safer and more efficient intersection traversal. This approach outperforms standard MARL techniques such as Independent PPO and QMIX, highlighting the benefits of incorporating domain knowledge into reinforcement learning models (5).

A notable alternative to vehicle-centric learning is the route-based MARL approach, where traffic routes are treated as individual agents instead of vehicles. This reduces computational complexity and improves scalability, demonstrating effective coordination in large-scale urban intersections (3).

In addition to MARL methods, single-agent reinforcement learning approaches have also been explored for intersection navigation. A TD3-based model has been applied to T-intersection scenarios in dense traffic, achieving robust decision-making without reliance on vehicle-to-vehicle (V2V) communication (1). Another study investigates the use of PPO with Generalized Advantage Estimation (GAE) to guide CAVs in mixed traffic envi-

ronments, showing that a small number of strategically controlled CAVs can significantly enhance overall traffic efficiency (7).

While these approaches have demonstrated promising results, challenges remain in scalability, real-world deployment, and coordination under dynamic traffic conditions. Future work should focus on integrating game-theoretic models, hybrid learning approaches, and real-world testing to further enhance autonomous intersection navigation.

3 Mixed Autonomy Traffic Model

Reinforcement Learning (RL) is a computational approach to learning from interaction, where an agent learns to optimize its behavior in an environment by receiving rewards for performing actions that lead to desirable outcomes. Unlike supervised learning, which relies on labeled data, RL enables agents to explore and exploit their environment to learn optimal decision-making policies autonomously.
Mathematically, RL is formulated as a Markov Decision Process (MDP), defined by the tuple:

$$MDP = (S, A, T, R, \gamma)$$

where:

- S is the set of possible states the agent can occupy.
- A is the set of possible actions the agent can take.
- $T(s'|s, a)$ is the transition probability function that defines the likelihood of moving from state s to s' after taking action a.
- $R(s, a)$ is the reward function that assigns a scalar reward to an agent for performing action a in state s.
- $\gamma \in [0, 1]$ is the discount factor that balances the importance of immediate versus future rewards.

The goal of RL is to learn an optimal policy π^*, which maps states to actions in a way that maximizes the expected cumulative reward over time:

$$\pi^* = argmax_\pi E[\sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t)]$$

where the expectation is taken over all possible trajectories of the agent's interactions with the environment.

The mixed autonomy traffic environment in a simulation can be naturally modeled as a Markov Decision Process (MDP). At each discrete time step t, the state s_t, encapsulates a comprehensive representation of the traffic system, including the positions, velocities, accelerations, and other metadata of all vehicles in the road network. This state representation provides the agent(s) (i.e., autonomous vehicles, AVs) with the necessary information to make optimal driving decisions. The action a_t is defined as a tuple of acceleration values for all controlled AVs in the network at time step t. The non-AV

vehicles follow a human driving behavior model, which in this work is implemented using the Intelligent Driver Model (IDM) (9), a widely used car-following model. The reward function $r(s_t, a_t)$ is typically defined to optimize traffic performance, incorporating metrics such as throughput, fuel consuption and safety measures.
The stochastic transition function $T(s_{t+1}|s_t, a_t)$ is not explicitly defined but can be empirically sampled from a traffic simulator. Given a state s_t and an action a_t, the simulator updates all vehicles dynamics according to their respective control policies and physical constraints, generating the next state s_{t+1}. In this work, the Simulation of Urban Mobility (SUMO), integrated in Flow, is used to model realistic urban and highway traffic dynamics.

3.1 Applied Reinforcement Learning Algorithms

In this study, three well-established reinforcement learning algorithms were employed: **Proximal Policy Optimization (PPO)**, **Asynchronous Advantage Actor-Critic (A3C)**, and **Deep Deterministic Policy Gradient (DDPG)**. These algorithms fall within the category of model-free reinforcement learning, meaning they do not explicitly model the environment's transition dynamics but instead learn from direct interaction with the simulator (SUMO in our case). However, they differ significantly in terms of policy representation, learning paradigm, and sample efficiency.

Proximal Policy Optimization (PPO) PPO is a state-of-the-art on-policy, policy-based reinforcement learning algorithm that improves upon traditional policy gradient methods by introducing a clipped objective function to prevent large policy updates, thereby enhancing stability. PPO is known for good sample efficiency and stable training, making it widely used in continuous control tasks.

Asynchronous Advantage Actor-Critic (A3C) A3C is an on-policy actor-critic algorithm that utilizes multiple workers asynchronously to collect experience, which helps improve the stability of training and prevents convergence to local optima. It learns both a policy (actor) and a value function (critic) in parallel, leveraging the advantage function to reduce variance in policy updates. However, due to its on-policy nature, it is less sample efficient than off-policy methods.

Deep Deterministic Policy Gradient (DDPG) is an off-policy, actor-critic reinforcement learning algorithm designed for continuous action spaces. Unlike PPO and A3C, DDPG uses a deterministic policy, meaning it directly outputs actions instead of sampling them from a distribution. It utilizes a replay buffer to improve sample efficiency and target networks to stabilize training. Since it is off-policy, it can re-use past experiences, making it more data-efficient than PPO and A3C.

Algorithm	**Model-Free/Model-Based**	**Value-Based/Policy-Based**	**On-Policy/Off-Policy**
PPO	Model-Free	Policy-Based	On-Policy
A3C	Model-Free	Policy-Based (Actor-Critic)	On-Policy
DDPG	Model-Free	Actor-Critic	Off-Policy

Table 1: Comparison of Reinforcement Learning Algorithms

4 Intersection Scenario

The intersection scenario consists of a four-way junction with incoming traffic from multiple lanes. The environment models both human-driven vehicles (HDVs) and autonomous vehicles (AVs), where the AVs must learn an efficient coordination policy through reinforcement learning. The control zone extends 200 meters before the intersection to allow AVs to make informed acceleration decisions based on observed traffic conditions. The inflow rate is set at 200 vehicles per hour, ensuring a continuous stream of vehicles entering the simulation. Unlike traditional traffic control mechanisms, traffic signals are disabled, creating a fully unsignalized intersection where AVs must learn to navigate efficiently and resolve conflicts dynamically.

4.1 Network Configuration in Flow

The road network topology is explicitly defined using the Flow Network API, which facilitates structured interactions with the SUMO simulator by enabling the creation of custom traffic networks. This is achieved by extending the `Network` class, allowing for precise specification of key elements such as nodes, edges, routes, and traffic inflows.
As mentioned, to ensure realistic traffic behavior, human drivers follow the Intelligent Driver Model (IDM) with Gaussian noise, capturing real-world variability in acceleration and deceleration decisions. AVs, on the other hand, employ different reinforcement learning strategies, such as Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), and Asynchronous Advantage Actor-Critic (A3C). Routing decisions are handled by the `MinicityRouter` module from the Flow library, which ensures that vehicles follow realistic trajectories through the intersection.

In this work, a customized intersection network, `Intersection`, was developed using the Flow library, integrating key traffic simulation parameters and reinforcement learning requirements. The main contributions include defining the network topology, configuring realistic vehicle behaviors, and structuring the reinforcement learning environment to ensure effective policy training. These components were implemented by specifying key network attributes defined above.

4.1.1 Network Topology and Structural Definitions

The network topology, illustrated in Figure **??**, consists of a four-way intersection with an explicit representation of entry and exit points, traffic flows, and vehicle behavior models. The core elements of the network include:

- **Nodes Definition** (`specify_nodes`): The intersection comprises five nodes—four corresponding to entry/exit points and one central junction ($J1$). Each node is positioned at a distance of 200 meters from the center, ensuring a balanced intersection layout. The SCALING factor allows dynamic modifications to the network dimensions, facilitating scenario adjustments for different traffic conditions.

- **Edges Definition** (`specify_edges`): The road network includes eight directed edges, each 200 meters in length, connecting entry and exit nodes. Every edge is designed with two lanes to accommodate multiple vehicles and lane-changing maneuvers. The speed limit is set at 30 m/s, enforcing realistic driving constraints.

Edge lengths are automatically computed based on node distances, ensuring geometric consistency.

- **Traffic Flow Management** (`specify_routes` and `InFlows`): Vehicles enter the network dynamically using the `InFlows` object, which enables the specification of incoming traffic volume, departure conditions, and stochastic variations in vehicle placement. The `InFlows.add()` function ensures vehicles are inserted in a stochastic manner, with a randomized departure lane selection, contributing to diverse traffic conditions across different training episodes.

4.1.2 Vehicle Behavior and Control Strategies

To simulate mixed-autonomy traffic, two distinct vehicle types were introduced with different control models:

Human-Driven Vehicles (HDVs):

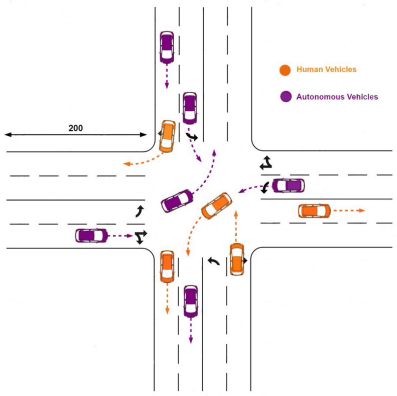
- Controlled by the Intelligent Driver Model (IDM), ensuring realistic longitudinal motion.
- Lane changes follow SUMO's default lane-changing model, replicating human behavior.
- Speed mode is set to *obey safe speed*, enforcing adherence to safe following distances and braking constraints.

Autonomous Vehicles (AVs):

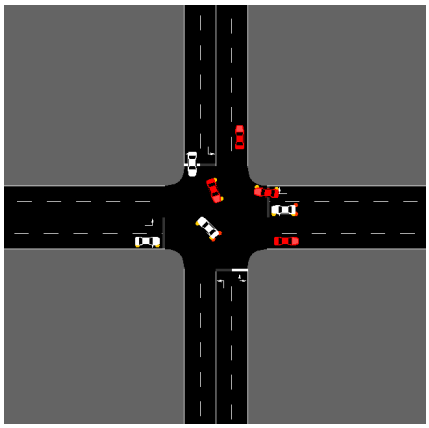
- Controlled by a Reinforcement Learning (RL) policy, which determines acceleration and deceleration actions.
- Speed mode is set to *obey safe speed*, ensuring AVs adhere to the network's speed limits.
- Maximum acceleration and deceleration are constrained at $3m/s^2$, enforcing safe maneuverability and preventing unrealistic behaviors.

5 Multi-Agent Reinforcement Learning Approach

The intersection coordination problem is naturally modeled as a multi-agent reinforcement learning (MARL) problem, where multiple autonomous vehicles (AVs) must learn to navigate the intersection efficiently while interacting with human-driven vehicles (HDVs). In this setup, each AV is treated as an independent learning agent, receiving local observations and selecting acceleration actions based on a shared policy. This approach allows AVs to coordinate their behavior dynamically without relying on explicit traffic signals.



(a) Implemented unsignalized intersection network with HVs and AVS involved



(b) Implemented unsignalized intersection network within SUMO

5.1 Multi-Agent Policy Decomposition

The problem is formulated as a decentralized partially observable Markov decision process (Dec-POMDP), where each AV makes decisions independently based on partial observations of the environment. Specifically, the global policy $\pi_\theta : S \to \Delta(A)$ is decomposed into per-agent policies $\pi_\theta^i : O^i \to \Delta(A^i)$, such that:

$$\pi_\theta(s) = \prod_i \pi_\theta^i(o^i)$$

where:

- A^i is the acceleration action space for each AV_i.
- O^i is the local observation space for AV_i, containing only a subset of the global state S.
- π_θ^i represents the shared policy function that maps AV_i's local observation o^i to a probability distribution over actions.

This decomposition enables the system to handle an arbitrary number of AVs while maintaining efficient coordination. The shared policy ensures that all AVs operate under the same learned strategy, adapting dynamically to changing traffic conditions.

5.2 Observation State

The observation function in this multi-agent reinforcement learning framework is designed to provide each autonomous vehicle (AV) with a local representation of the traffic state at the intersection. Since the scenario is modeled as a decentralized partially observable Markov decision process, each AV can only access information within its local perception range rather than the full state of the environment.
The function `get_state(self)` defines the observation for each RL-controlled AV by extracting relevant traffic features and normalizing them for stable learning. Below, each key component of the observation function are explained.

Structure of the Observation Space Each AV's observation vector consists of six key features that capture relevant information about its own state and its interactions with nearby vehicles. The observation vector is represented as:

$$o^i = [\frac{x^i}{L}, \frac{v^i}{V_{max}}, \frac{v_{lead} - v^i}{V_{max}}, \frac{d_{lead}}{L}, \frac{v^i - v_{follow}}{L}, \frac{d_{follow}}{L}]$$

where:

- x^i is the position of the AV along its current edge.
- v^i is the current velocity of the AV.
- v_{lead} is the velocity of the leading vehicle, if present.
- d_{lead} is the headway (distance to the leading vehicle).
- v_{follow} is the velocity of the following vehicle, if present.
- d_{follow} is the headway to the following vehicle.

To ensure numerical stability, position and distances are normalized dividing by the total network length L while velocities and relative speed differences are divided by the maximum speed. By normalizing all observations within a standard range, we improve the stability of policy learning and reduce the variance of gradient updates.

The observation function is structured to provide each autonomous vehicle (AV) with only the essential local information needed for effective acceleration decisions. The ego vehicle's position and speed allow it to assess progress through the intersection, while the leading vehicle's velocity and headway help anticipate slowdowns and avoid rear-end collisions. Additionally, considering the following vehicle's speed and headway enables smoother acceleration regulation, preventing abrupt stops that could propagate congestion.
This design ensures decentralization, as each AV operates solely on local observations, making the approach scalable across different traffic densities without relying on global state information. At the same time, coordination emerges through the inclusion of relative distances and velocities, allowing AVs to learn cooperative behaviors that enhance traffic flow through multi-agent reinforcement learning (MARL).

5.3 Shared Policy Learning

In multi-agent reinforcement learning (MARL), a shared policy is a strategy where multiple agents utilize the same learned policy rather than having individual policies for each agent. This approach is particularly beneficial in large-scale environments such as autonomous vehicle coordination, where multiple vehicles need to operate under a unified strategy while adapting to dynamic traffic conditions. Therefore, the shared policy mechanism plays a crucial role in improving learning efficiency and ensuring cooperative behavior among RL-controlled vehicles.
In Flow, a shared policy, implemented using **Ray RLlib**'s multi-agent API, means that all AVs are controlled by the same function approximator rather than having separate, independent policies for each agent. This setup is beneficial because:

- Experience is shared among all AVs, leading to faster learning.
- The model generalizes across different numbers of AVs, making it applicable to varying traffic conditions.
- The computational cost is reduced, as we train a single policy instead of multiple independent ones.

5.4 Reward Function

The reward function is formulated to encourage efficient traffic flow, safety, and coordination among autonomous vehicles (AVs) in the intersection scenario. It integrates multiple components that promote desirable driving behaviors while penalizing inefficiencies and risky actions. The total reward function r_t at time step t is defined as:

$$r_{\text{total}} = (\lambda \cdot r_{\text{combined}}) + r_{\text{vel}} + (\beta \cdot r_{\text{headway}}) \tag{1}$$

$$r_{\text{combined}} = r_{\text{vel}} + r_{\text{standstill}} + r_{\text{progress}} + r_{\text{delay}} + r_{\text{lane\_change}} + r_{\text{throughput}} \tag{2}$$

$$r_{\text{headway}} = \sum_{i \in \mathcal{V}_{\text{RL}}} \min\left(\frac{t_i - t_{\min}}{t_{\min}}, 0\right) \tag{3}$$

where the factor λ ensures a balanced contribution of different reward terms, and β scales the penalty for small time headways. The reward is shared among all AVs, reinforcing cooperative behavior in the system. The experimentally determined reward function has shown better performance compared to a simple summation approach. This function integrates multiple reward components defined in Flow, ensuring a more balanced and efficient reinforcement learning process. Specifically, it includes the following reward terms:

- **Desired Velocity Reward**: Encourages vehicles to maintain a target speed.
- **Penalize Standstill**: Penalizes vehicles that remain stationary.
- **RL Forward Progress**: Rewards RL vehicles for forward movement.
- **Punish RL Lane Changes**: Penalizes lane changes by RL-controlled vehicles.
- **Min Delay**: Encourages the minimization of travel delays.
- **Penalize Small Time Headways**: A newly introduced component that penalizes vehicles if their time headway is too low.
- **Throughput Reward**: A newly introduced component in this work that incentivizes vehicles to maximize intersection throughput, encouraging efficient traffic flow and reducing congestion.

The throughput-based reward has been amplified to further encourage vehicles to optimize intersection usage efficiently. Additionally, the penalties for standstill and lane changes have been reduced slightly to allow greater adaptability in decision-making. The combination of these modifications has shown improved stability and performance in traffic coordination.
Further details and an empirical evaluation of these reward components can be found in Appendix A.

5.5 Simulation and Training Framework

This section outlines the simulation parameters, environment configuration, and reinforcement learning training methodology used in this study. The objective is to train multi-agent reinforcement learning policies for autonomous vehicles in an unsignalized intersection using different reinforcement learning algorithms.

5.5.1 Simulation Parameters

The intersection is modeled using the Simulation of Urban Mobility (SUMO) framework, where the road network consists of four bidirectional roads, each containing four lanes—two for incoming and two for outgoing traffic and at the beginning of each episode, vehicles are randomly spawned at the entry points. The main simulation parameters are summarized in table 2:

Parameter	**Value**
Road length	200 m per approach
Speed limit	30 m/s
Traffic flow rate	200 vehicles per hour per entry lane
Simulation step (δ)	0.1 s
Training horizon (H)	1500 steps per episode

Table 2: Main simulation parameters.

5.5.2 Environment Configuration

This section outlines the environmental parameters used to set up the implemented MARL environment, `VariantNormalizedMultiAgentAccelPOEnv`. More in details, it requires some parameters that are shown in the following table 3:

Table 3: Environmental Parameters for the Multi-Agent Simulation

Parameter	**Value**	**Description**
Maximum Acceleration	3 m/s^2	Upper bound on acceleration for AVs
Maximum Deceleration	3 m/s^2	Lower bound on deceleration for AVs
Target Velocity	20 m/s	Desired speed for AVs
Traffic Flow Rate	200 veh/h	Vehicles entering per lane per hour
Initial Speed (Human)	10 m/s	Starting speed of human-driven vehicles
Initial Speed (AVs)	5 m/s	Starting speed of autonomous vehicles
Speed Limit	30 m/s	Maximum speed allowed in the environment
Horizon	1500	Time steps per episode
N_Rollouts	40	Number of episodes collected per training iteration

5.5.3 Training and Evaluation Metrics

Training is carried out using A3C, PPO, and DDPG, each employing distinct learning paradigms to handle continuous control tasks. The training process spans 50 iterations with 40 rollouts, with the number of timesteps per iteration varying across algorithms due to differences in data collection and policy update mechanisms. While PPO collects experiences in fixed rollouts, A3C asynchronously gathers and updates policies, leading to fluctuations in timesteps per iteration. DDPG, being an off-policy algorithm, leverages

a replay buffer, further influencing the amount of experience used per iteration. The key hyperparameters used for each algorithm are detailed in Appendix B.

The performance of the trained policies is assessed using a set of key evaluation metrics, which provide insights into traffic efficiency, safety, and energy consumption.

- **Episode Reward**: Represents the cumulative reward collected by reinforcement learning agents, providing a measure of overall policy performance.
- **Policy Reward Mean**: Computes the average reward obtained per policy, facilitating a fine-grained comparison in multi-agent settings.
- **Throughput Efficiency**: The number of vehicles successfully exiting the network per unit time, reflecting the effectiveness of the trained policies in maintaining smooth traffic flow. This metric is crucial for understanding congestion mitigation and network utilization.
- **Average Speed**: The mean velocity of all vehicles in the environment, reflecting traffic throughput. Additionally, separate metrics are recorded for autonomous vehicles (AVs) and human-driven vehicles.
- **Number of Vehicles in the System**: Captures the total number of active vehicles at each timestep, which is useful for congestion analysis.
- **Acceleration Profiles**: The average acceleration for both AVs and human-driven vehicles, providing insights into driving smoothness and stability.
- **Fuel Consumption**: The total fuel consumption across all vehicles, serving as an indicator of energy efficiency.
- **Collision Count**: The total number of detected collisions during the episode, a crucial metric for safety evaluation.

During training, these metrics are updated at every timestep and aggregated over the entire episode.

6 Results and Analysis

This section presents the experimental results obtained from training and evaluating the three reinforcement learning algorithms. To assess the impact of reinforcement learning on autonomous vehicle control at unsignalized intersections, the results obtained of each model were compared against a baseline scenario consisting exclusively of human-driven vehicles (HVs). The baseline metrics were computed over 50 runs with a horizon of 1500 timesteps and 40 rollouts. Detailed graphs for the evaluation are further explained in Appendix C. These figures illustrate the performance trends of each algorithm, providing a detailed comparison of their learning dynamics and traffic efficiency.

6.1 Analysis of PPO Training Performance

The training process of the PPO algorithm showed notable trends across key performance metrics, highlighting both the strengths and areas for improvement in traffic coordination.

The mean episode reward followed a rising trajectory, peaking at 855,149, indicating consistent policy refinement. However, fluctuations in intermediate stages suggest that the agent required time to stabilize optimal decision-making. Similarly, the episode length increased up to 1,224.98 steps, reflecting a more intricate and prolonged interaction between agents as learning progressed.

Traffic flow analysis revealed that the average inflow rate peaked at 2,846 vehicles per hour, but the outflow rate lagged at 323 vehicles per hour. This huge disparity suggests bottlenecks emerging in the system, possibly due to imperfect traffic coordination strategies. Throughput efficiency, peaking at 0.141, demonstrated lack of success in optimizing road usage but indicates room for enhancement.

Safety metrics showed a gradual reduction in collision frequency, but the average collision rate remained at 0.47 per episode, higher than expected since the average speed is very low. The total collision count however, followed a declining pattern, suggesting that improvements were achieved over time, albeit at a slower rate than desired. Fuel consumption shows an increasing pattern, reaching a low of 0.00017 units per vehicle, and a maximum of 0.00052. A noticeable trend was the fluctuation in average vehicle speed, which peaked at 6.20 m/s. This suggests that the policy alternated between aggressive and conservative strategies before converging toward a more stable driving pattern. of cars in the system, averaging 24.09, remained relatively stable, supporting the model's ability to manage traffic density effectively.

From a training perspective, the policy loss remained low, with a minimum of -0.00040, indicating controlled updates. The value function loss, reaching 455,132.4, reflected significant variance in reward estimation, which may have contributed to the learning instability observed in early iterations. Overall, PPO demonstrated promising capabilities in traffic management but also exposed potential inefficiencies in throughput and collision reduction. The trends suggest that while the algorithm successfully optimized fuel consumption and decision-making over time, further refinements could enhance stability and coordination among autonomous agents.

6.2 Analysis of A3C Training Performance

The training process of the Asynchronous Advantage Actor-Critic (A3C) model exhibited several key trends across performance metrics, indicating the agent's learning behavior and convergence characteristics.

The episode reward mean reached a peak of 887,670, indicating effective policy learning while the average episode length increased up to 1,217.5 steps, suggesting a more complex decision-making process. The total timesteps processed were 93,289 across 77 episodes, with a total of 50 training iterations, ensuring sufficient learning stability.

In terms of traffic efficiency, the mean inflow rate reached 5,781 vehicles per hour, while

the outflow rate stabilized at 735 vehicles per hour. The throughput efficiency peaked at 0.327, demonstrating that autonomous vehicles contributed to optimized road usage. From a safety perspective, the collision rate was reduced to an average of 0.25 per episode. The total number of collisions also followed a downward trend, reinforcing the model's ability to minimize risks. Additionally, fuel consumption improved significantly, reaching a low of 0.00024 units per vehicle, indicating energy-efficient driving patterns learned by the agents.

Vehicle behavior analysis highlighted a clear distinction between human-driven and autonomous vehicles. The average speed across all vehicles peaked at 13.2 m/s, ensuring smooth and controlled driving patterns. The mean number of cars in the simulation was 22.32, further validating traffic stability. Finally, the policy loss reached a minimum of -77,308.9, while the value function loss was stabilized at 4,690.0, demonstrating effective optimization of the learning process. The entropy values indicated sufficient exploration during training, preventing premature convergence to suboptimal policies.

Overall, the A3C algorithm successfully optimized traffic flow, reduced collisions, and improved energy efficiency while maintaining stable training dynamics. These results validate the applicability of multi-agent reinforcement learning in autonomous vehicle coordination.

6.3 Analysis of DDPG Training Performance

The training dynamics of the DDPG algorithm exhibited distinct patterns compared to PPO and A3C, with noticeable differences in stability and learning efficiency.

The episode reward mean reached a peak of 984,43, indicating strong policy performance. However, fluctuations throughout training suggest intermittent instability, possibly due to the deterministic nature of the policy updates. The episode length increased up to 1,500 steps, the highest among the tested algorithms, implying extended interactions between agents and a deeper decision-making process.

The total timesteps processed amounted to 50,001 over 40 episodes, with 50 training iterations, reflecting a structured learning progression. Unlike PPO, DDPG demonstrated a slower adaptation rate, likely influenced by its reliance on continuous action spaces without explicit policy entropy for exploration.

Traffic flow analysis was informed by a throughput efficiency mean of 0.3374, which suggests that DDPG facilitated reasonable traffic progression, albeit with room for improvement. Fuel consumption remained at an average of 0.000664, reflecting an energy-efficient policy. Additionally, the mean acceleration of autonomous vehicles was recorded at 1.9688, while the average speed reached 12.8978, further indicating controlled agent behavior. Notably, total collisions averaged 0.3768, implying that while safety mechanisms were in place, occasional incidents still occurred.

From a training perspective, the algorithm exhibited high variance in reward trends, indicating a greater sensitivity to hyperparameter tuning. The deterministic nature of DDPG might have led to more stable, albeit less exploratory, behaviors over time. Overall, while

DDPG achieved competitive reward levels and facilitated prolonged agent interactions, its performance in traffic flow optimization remains uncertain due to the absence of explicit throughput and collision-related metrics. Future refinements, such as improved exploration strategies and reward shaping, could further enhance its applicability to autonomous vehicle coordination.

6.4 Comparative Analysis

The performance comparison between A3C, PPO, and DDPG highlights key differences in learning efficiency, traffic coordination, and agent stability within an unsignalized intersection scenario.

Learning Efficiency and Stability A3C demonstrated the most stable convergence pattern, with continuous learning improvements across 93,289 timesteps. PPO exhibited strong but more fluctuating performance over 210,066 timesteps, indicating a longer training time with intermittent instability. DDPG outperformed both in reward maximization, reaching 984,433, but showed inconsistent learning patterns and required fine-tuned hyperparameters to prevent unstable behaviors.

Traffic Flow and Throughput DDPG outperformed in traffic optimization, achieving a throughput efficiency of 0.3374, with a mean outflow rate of 757.67 vehicles per hour, making it the most effective in optimizing traffic flow. Followed, with similar metrics by A3C with a throughput 0.327 and a mean outflow rate of 735. PPO, instead, lagged behind with an efficiency of 0.141 and an outflow rate of 323 vehicles per hour, suggesting suboptimal congestion handling.

Safety and Collision Reduction DDPG also led in collision mitigation, with an average collision rate of 0.3768 per episode, while A3C showed a declining trend in total collisions stabilizing at 0.38. PPO maintained a higher collision rate at 0.47 per episode.

Energy Efficiency and Vehicle Behavior Fuel consumption analysis showed that PPO achieved the best optimization, reducing fuel consumption to 0.00020 units per vehicle, followed by A3C at 0.00021. Finally, DDPG showed an average fuel consumption rate of 0.000664, the higer one, but this is inline with the fact that in terms of speed stability and aceleration it reached the best performance mantaining an average speed of 12.89 m/s and a an average AVs acceleration of 1.968 m/s^2, A3C's vehicles instead, maintained an average speed of 10.97 m/s, compared to PPO's 6.20 m/s, suggesting that A3C contributed to smoother and more predictable traffic patterns.

Policy Optimization and Training Challenges PPO exhibited controlled policy loss (-0.00040) but suffered from high value function loss (455,132.4), indicating large variance in reward estimation. A3C's policy loss was higher (-77,308.9), but its value function loss remained low (4,690.0), suggesting more stable reward prediction. DDPG, being a deterministic policy approach, exhibited instability in training, requiring careful hyperparameter tuning to prevent drastic fluctuations in performance.

Conclusion Each model presents trade-offs: A3C delivers stable learning, high traffic throughput, and strong safety metrics, making it the best suited for large-scale traffic coordination. PPO, while effective, requires longer training time and suffers from traffic congestion issues. Despite DDPG achieves the highest reward, the lowest collision rate and the highest throughput performance amonge the three models, it showed lacks robustness in training, requiring further refinements. Future work should explore hybrid approaches that combine the stability of A3C with the fine-grained control of DDPG to enhance reinforcement learning strategies for autonomous vehicle coordination.

7 Conclusion and Future Work

This project aimed to develop a multi-agent reinforcement learning (MARL) framework for autonomous vehicle coordination within a traffic simulation. The primary objective was to train multiple AVs as independent agents to optimize their movements while interacting with each other in real-time. The trained policies sought to balance individual objectives, such as minimizing travel time, with collective goals, such as ensuring smooth traffic flow and preventing accidents. Through the implementation of reinforcement learning-based control policies, the AVs demonstrated an ability to navigate the intersection while improving throughput and maintaining safety.

However, one key insight from this study is that coordination in MARL does not necessarily imply cooperation. The use of a shared policy allowed AVs to adopt similar behaviors, reducing variability and facilitating emergent coordination. However, true cooperative decision-making, where AVs actively adjust their strategies based on system-wide goals, would require explicit cooperation mechanisms such as global rewards, inter-agent communication, or adaptive multi-agent policies. While the current approach enables implicit coordination through shared learning, future extensions could explore these elements to achieve higher-level cooperative behaviors that enhance overall traffic efficiency.

7.1 Future Work

This work employs **Centralized Training with Decentralized Execution (CTDE)**, a common framework in MARL where policies are trained with access to global information but executed in a decentralized manner. While this approach enables implicit coordination through shared learning, future studies could explore additional cooperative mechanisms to further enhance system performance:

- **Scaling to More Complex Traffic Scenarios**: The current setup focuses on intersection management, but future work could extend the simulation to more complex scenarios such as highway merging, roundabouts, or dynamic traffic conditions. This would help evaluate the generalizability and robustness of the learned policies in diverse environments.
- **Incorporating Vehicle-to-Vehicle (V2V) Communication**: Introducing explicit message-passing mechanisms could enable AVs to share real-time traffic state information, improving coordination and reducing uncertainties in decision-making. This would be particularly useful in scenarios with limited observability or high traffic density.

- **Exploring Multi-Agent Cooperation Strategies**: While this work primarily employed independent learning with a shared policy, future studies could explore explicit cooperative strategies, such as adaptive policies that dynamically adjust their behavior based on the actions or states of other agents. This would enable more flexible and context-aware cooperation in complex traffic scenarios.
- **Comparing Shared vs. Independent Policies**: A deeper comparison between shared-policy learning and independent-policy learning could reveal trade-offs in terms of scalability, adaptability, and traffic throughput. For instance, independent policies might offer greater flexibility in heterogeneous environments, while shared policies could simplify training in homogeneous settings.
- **Real-World Deployment Considerations**: Testing the learned policies in high-fidelity autonomous driving simulators (e.g., CARLA, LGSVL) would be an essential step towards practical deployment in intelligent transportation systems. This would help validate the policies under more realistic conditions and identify potential challenges for real-world implementation.

In conclusion, this project provided valuable insights into multi-agent learning for traffic coordination, demonstrating that reinforcement learning can be a viable solution for optimizing AV behavior. However, achieving full cooperation among agents remains an open challenge, requiring further research into communication, global optimization, and policy adaptability.

A Detailed Reward Functions

In this appendix, a detailed description of the reward functions used in the reinforcement learning model is provided.

A.1 Reward Functions

The total reward function in Equation 4 is a weighted sum of different terms explained in the following session.

$$r_{\text{total}} = (\lambda \cdot r_{\text{combined}}) + r_{\text{vel}} + (\beta \cdot r_{\text{headway}}) \tag{4}$$

$$r_{\text{combined}} = r_{\text{vel}} + r_{\text{standstill}} + r_{\text{progress}} + r_{\text{delay}} + r_{\text{lane\_change}} + r_{\text{throughput}} \tag{5}$$

$$\tag{6}$$

- r_{vel} encourages AVs to maintain a target velocity v_{target} minimizing deviations from the optimal speed:

$$r(s,a) = \max\left(\frac{\|\mathbf{v}^*\| - \|\mathbf{v} - \mathbf{v}^*\|}{\|\mathbf{v}^*\| + \epsilon}, 0\right) \tag{7}$$

  Where:

  - $\mathbf{v}$ is the vector of the current velocities of the vehicles.
  - $\mathbf{v}^*$ is the vector of the desired target velocities for each vehicle.
  - $\|\cdot\|$ denotes the Euclidean norm.

  - ϵ is a small positive value to avoid division by zero.

- $r_{progress}$ rewards forward progress through the intersection, defined as the difference in position along the route.

$$r(s,a) = g \cdot \sum_{i \in \mathcal{V}_{\mathrm{RL}}} |v_i| \tag{8}$$

  Where:

  - g is the gain factor.
  - $\mathcal{V}_{\mathrm{RL}}$ is the set of RL vehicles.
  - v_i is the velocity of vehicle i.

- $r_{standstill}$ penalizes AVs that remain stationary, discouraging unnecessary stopping.

$$r(s,a) = -g \cdot N_0 \tag{9}$$

  Where:

  - g is a multiplicative factor (gain).
  - N_0 is the number of vehicles with speed $v = 0$.

- r_{lane_change} penalizes excessive lane changes to promote stability in traffic flow.

$$r(s,a) = -p \sum_{i \in \mathcal{V}_{\mathrm{RL}}} \mathbb{1}(\mathrm{LC}_i) \tag{10}$$

  Where:

  - p is the penalty applied for each change of lane.
  - $\mathbb{1}(\mathrm{LC}_i)$ is an indicator function that equals 1 if vehicle i has changed lanes.

- r_{delay} encourages minimal travel time by penalizing prolonged intersection traversals.

$$r(s,a) = \max\left(\frac{C_{\max} - C}{C_{\max} + \epsilon}, 0\right) \tag{11}$$

  Where:

  - $C_{\max} = \Delta t \cdot N$ is the maximum cost, where N is the number of vehicles, and Δt is the time step.
  - $C = \Delta t \sum_i \frac{v_{\mathrm{top}} - v_i}{v_{\mathrm{top}}}$ is the actual cost.
  - v_{top} is the maximum speed limit.

- $r_{headway}$ penalizes vehicles that follow too closely, enforcing safe time headways.

$$r(s,a) = \sum_{i \in \mathcal{V}_{\mathrm{RL}}} \min\left(\frac{t_i - t_{\min}}{t_{\min}}, 0\right) \tag{12}$$

  Where:

- $t_i = \frac{d_i}{v_i}$ is the time headway of vehicle i.
- $t_{\min}$ is the minimum acceptable time headway.

- $r_{throughput}$: encourages high throughput efficiency by maximizing the number of vehicles exiting the system:

$$r(s, a) = g \cdot \text{outflow} \tag{13}$$

For a complete overview of the parameters used in the reward functions, refer to Table 4.

Reward Function	**Parameter**	**Value**
Desired Velocity	fail	kwargs['fail']
Penalize Standstill	gain (g)	0.5
RL Forward Progress	gain (g)	0.5
Punish RL Lane Changes	penalty (p)	0.5
Minimum Delay	min_delay factor	0.5
Throughput Reward	gain (g)	0.5
Penalize Small Time Headways	t_min	1
Combined Reward	λ	0.2
Cost2	β	0.5

Table 4: Summary of reward function parameters.

B Hyperparameters

Hyperparameter	Value
Discount Factor (γ)	0.99
Generalized Advantage Estimation (λ)	0.95
Learning Rate (α)	9.88×10^{-5}
Gradient Clipping	40.0
Entropy Coefficient	0.01
Value Function Loss Coefficient	1.0
Training Batch Size	60,000 (horizon $\cdot$ num_rollouts)
Policy Network Architecture	(256, 256) fully connected layers
Activation Function	Tanh
Number of Workers	1
Sample Batch Size	10
Number of Evaluation Episodes	10
Training Batch Size	$60,000$

Table 5: Best hyperparameters used for A3C training.

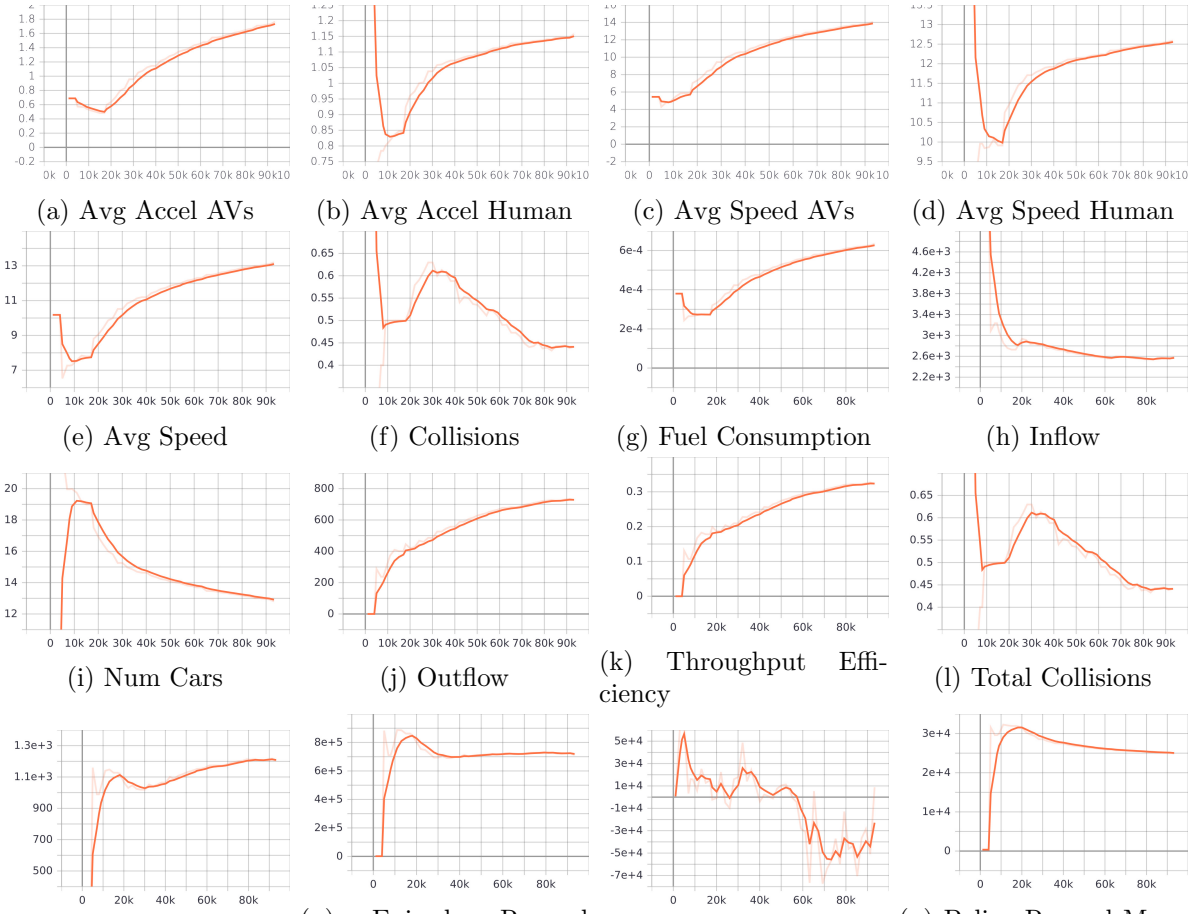
Hyperparameter	Value
Discount Factor (γ)	0.99
Generalized Advantage Estimation (λ)	0.97
Learning Rate (α)	1.76×10^{-4}
KL Coefficient	0.2
KL Target	0.01
Clipping Parameter (ϵ)	0.3
Entropy Coefficient	0.05
Number of SGD Iterations	10
Policy Network Architecture	(32, 32, 32) fully connected layers
Training Batch Size	4096
Sample Batch Size	200
SGD Minibatch Size	128
Use Generalized Advantage Estimation (GAE)	Yes
Value Function Loss Coefficient	1.0
Value Function Clipping Parameter	10.0
Number of Workers	1

Table 6: Best hyperparameters used for PPO training.

Hyperparameter	**Value**
Discount Factor (γ)	0.99
Actor Learning Rate (α)	2.34×10^{-5}
Critic Learning Rate	3.92×10^{-5}
Target Network Update Rate (τ)	0.01
Policy Network Architecture	(128, 128) fully connected layers
Critic Network Architecture	(128, 128) fully connected layers
Activation Function	ReLU
Training Batch Size	256
Replay Buffer Size	50,000
Exploration Noise Type	OU Noise
OU Noise Scale	0.115
OU Noise Theta	0.157
OU Noise Sigma	0.176
Exploration Fraction	0.1
Target Noise	0.2
Target Noise Clip	0.5

Table 7: Best hyperparameters used for DDPG training.

C Results



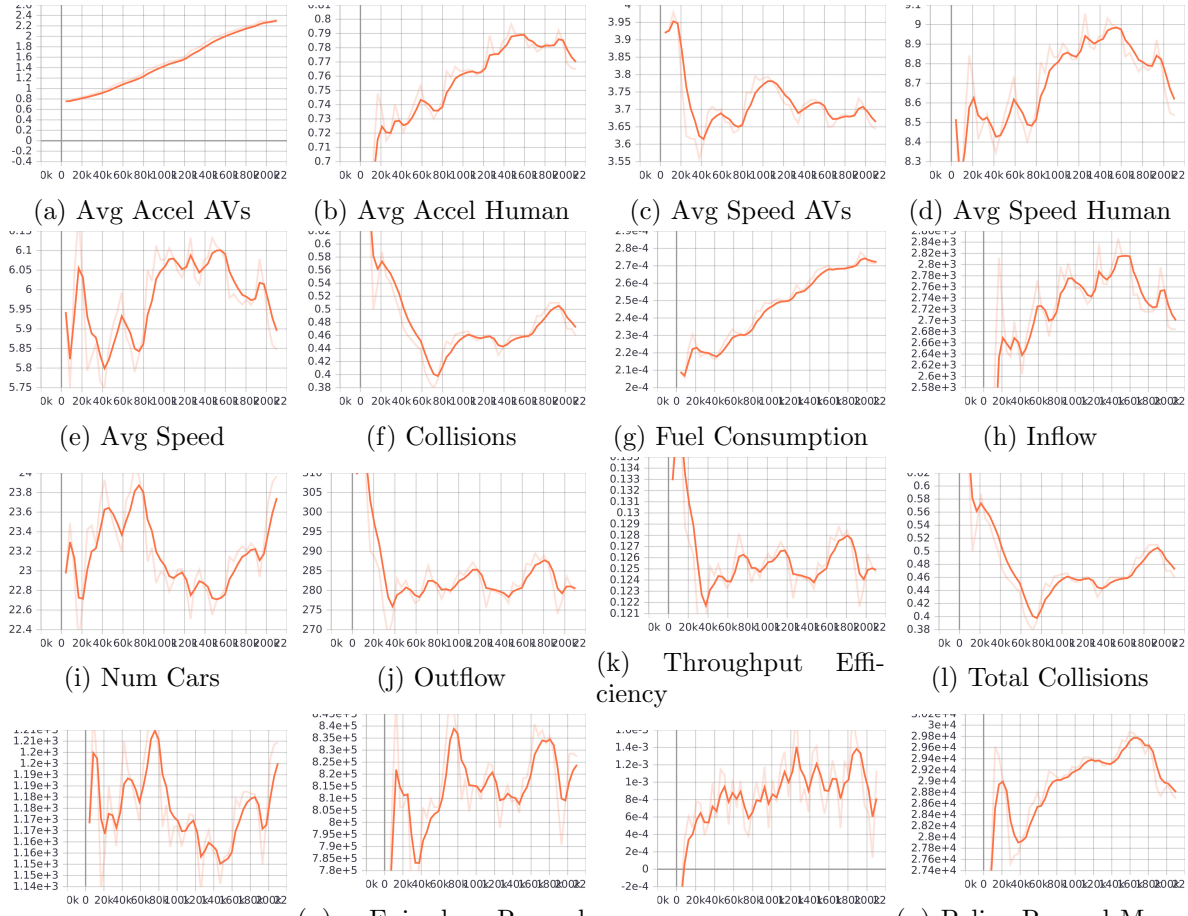
(a) Avg Accel AVs (b) Avg Accel Human (c) Avg Speed AVs (d) Avg Speed Human

(e) Avg Speed (f) Collisions (g) Fuel Consumption (h) Inflow

(i) Num Cars (j) Outflow (k) Throughput Efficiency (l) Total Collisions

(m) Episode Len Mean (n) Episode Reward Mean (o) Policy Loss (p) Policy Reward Mean AV

Figure 2: A3C Results



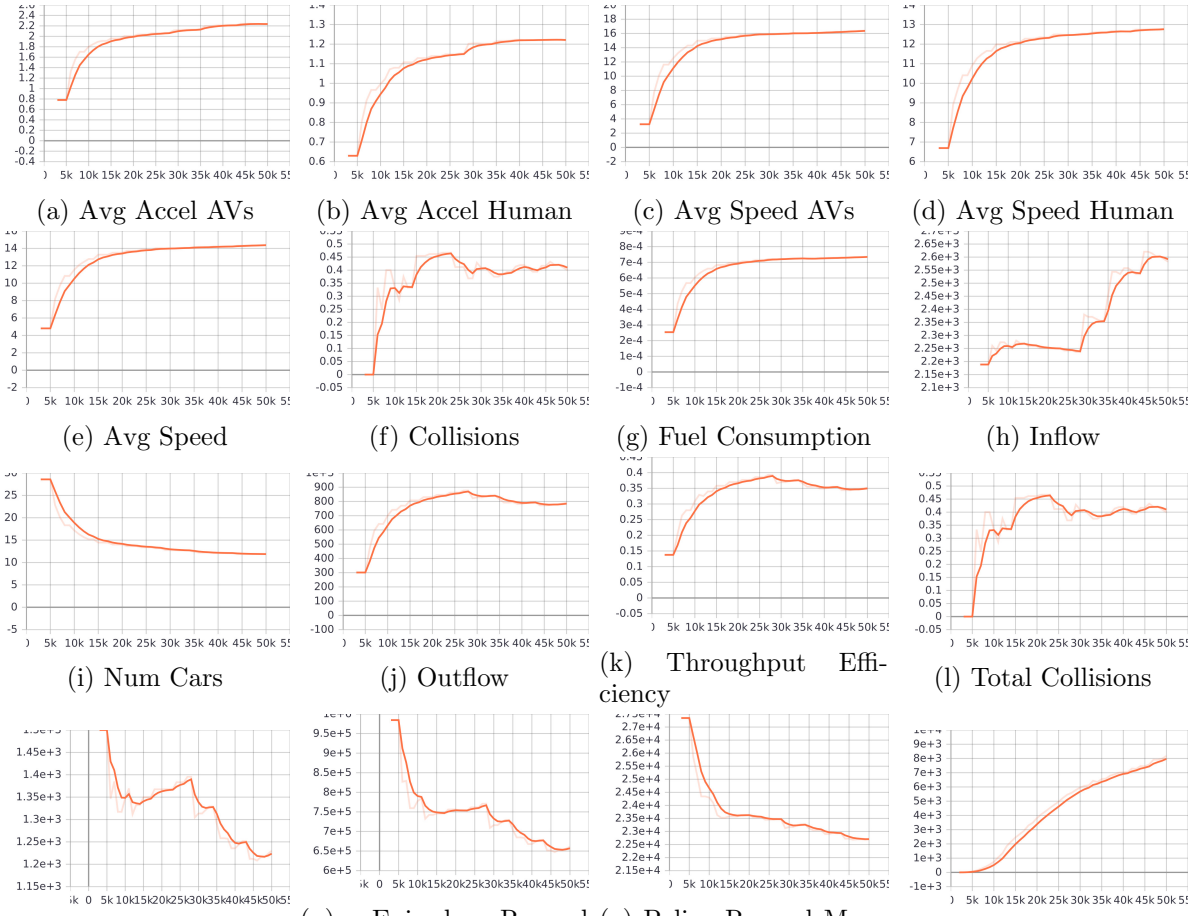
(a) Avg Accel AVs (b) Avg Accel Human (c) Avg Speed AVs (d) Avg Speed Human

(e) Avg Speed (f) Collisions (g) Fuel Consumption (h) Inflow

(i) Num Cars (j) Outflow (k) Throughput Efficiency (l) Total Collisions

(m) Episode Len Mean (n) Episode Reward Mean (o) Policy Loss (p) Policy Reward Mean AV

Figure 3: PPO Results



(a) Avg Accel AVs (b) Avg Accel Human (c) Avg Speed AVs (d) Avg Speed Human

(e) Avg Speed (f) Collisions (g) Fuel Consumption (h) Inflow

(i) Num Cars (j) Outflow (k) Throughput Efficiency (l) Total Collisions

(m) Episode Len Mean (n) Episode Reward Mean (o) Policy Reward Mean AV (p) Mean q

Figure 4: DDPG Results

References

[1] Badr Ben Elallid, Hamza El Alaoui, and Nabil Benamar, "Deep Reinforcement Learning for Autonomous Vehicle Intersection Navigation," , 2021.

[2] Christos Spatharis, Konstantinos Blekas , "Multiagent reinforcement learning for autonomous driving in traffic zones with unsignalized intersections," Journal of Intelligent Transportation Systems , 2022.

[3] Min Hua1,Xinda Qi,Dong Chen,Kun Jiang,Zemin Eitan Liu, Quan Zhou, Hongming Xu, "Multi-Agent Reinforcement Learning for Connected and Automated Vehicles Control: Recent Advancements and Future Prospects," Journal of Intelligent Transportation Systems , 2024.

[4] Sai Krishna Sumanth Nakka, Behdad Chalaki, Andreas A. Malikopoulos, "A Multi-Agent Deep Reinforcement Learning Coordination Framework for Connected and Automated Vehicles at Merging Roadways", 2022.

[5] Zihan Guo, Yan Wu, Lifang Wungm,"Heuristic-Based Multi-Agent Deep Reinforcement Learning Approach for Coordinating Connected and Automated Vehicles at Non-Signalized Intersections," IEEE Transactions on Intelligent Transportation Systems, vol. 25, 2024.

[6] Jian Zheng, Kun Zhu and Ran Wang, "Deep Reinforcement Learning for Autonomous Vehicles Collaboration at Unsignalized Intersections," IEEE Global Communications Conference: IoT and Sensor Networks, 2022

[7] Bile Peng, Musa Furkan Keskin, Balézs Kulcsar, Henk Wymeersch, "Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning ", 2021

[8] "Multi-Agent Reinforcement Learning for Autonomous Vehicle Coordination," 2021.

[9] M.Treiber, A.Hennecke, and D.Helbing, "Congested traffic states in empirical observations and microscopic simulations," Physical Review E, vol.62, no.2, pp.1805–1824, 2000.

[10] 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.