

Integrating a Large Language Model with Reinforcement Learning for Intelligent Traffic Signal Control

Gaurav Vuligonda
gauravvuli@gmail.com

ABSTRACT

Effective traffic light systems are essential to improve traffic flow, fuel efficiency, and public health. Traditional traffic light systems operate on fixed timers that give each direction equal green light time, making them inefficient under varying road environments. This study proposes and evaluates a hybrid adaptive traffic signal control system combining Proximal Policy Optimisation (PPO) reinforcement learning (RL) with a Large Language Model (LLM), providing an agentic extension for emergency scenario handling. A PPO agent is trained in the Simulation of Urban MObility (SUMO) simulator using synthetic, realistic traffic conditions. The standalone PPO agent reduced average halting vehicles per step by 31% and average waiting time per vehicle by 16% relative to the best fixed-timer configuration. The hybrid system achieved comparable regular-traffic performance, with average halting vehicles per step and average waiting time per vehicle both remaining within 3–4% of the standalone PPO agent, while completely eliminating emergency vehicle wait times. These findings demonstrate that a hybrid agent consisting of a PPO reinforcement learning agent and extended with an LLM for emergency response handling can significantly outperform fixed-timer configurations across traffic delay and congestion metrics, while effectively handling critical edge cases.

INTRODUCTION

Traffic congestion is the overcrowding of vehicles on roads, leading to increased travel times, greater fuel consumption and higher emissions. It has been a prevalent issue for decades, with it continuing to worsen as the urban population grows and vehicle usage rate increases. Beyond inconvenience, congestion weakens the economy of society, as time lost in traffic reduces productivity and increases travel costs for individuals (UCLA Institute of Transportation Studies, 2025). Additionally, traffic significantly impacts public health by increasing exposure to harmful pollutants released by vehicles. Prolonged exposure to fine particulate matter (PM_{2.5}) from traffic emissions has been linked to respiratory diseases and cardiovascular issues (Levy et al., 2010). Long-term air pollution has been associated with reduced life expectancy in urban populations. It is also extremely unsafe for society, as the large density of vehicles it

creates and erratic actions by frustrated drivers increase the risk of collisions. Not only can congestion lead to an increase in accidents, but it can also inhibit first-responders and emergency vehicles from reaching their destination, putting potential lives at risk.

A major contributing factor to traffic congestion in urban areas is poor signal timings, which can lead to a buildup of cars at intersections (Koonce, 2008). Traditional traffic light systems use a fixed signal timing system rather than adapting to real-time traffic conditions, making them inefficient during times when traffic flow fluctuates. These systems operate on predetermined schedules that do not account for unexpected cases such as accidents, natural disasters, or even uneven traffic across lanes. As the urban population continues to grow, these outdated traffic systems struggle to effectively manage the increasing number of vehicles on the road, further exacerbating congestion.

To address these issues, modern approaches explore integrating artificial intelligence (AI) into traffic light systems. AI-based Technologies, such as computer vision and reinforcement learning, have the potential to analyse traffic in real time using vehicle detection to dynamically adjust signal timings, improving overall traffic flow. This research paper proposes an adaptive hybrid traffic signal control system that combines reinforcement learning with an LLM. A Proximal Policy Optimisation agent is trained to adapt to the simulation environment and adjust signal timings accordingly to optimise traffic flow, replacing the traditionally fixed-timer systems most commonly used to control traffic. To handle emergency cases more robustly, a Claude Sonnet 4.6 (Anthropic, 2026) LLM intervenes when emergency vehicles are detected, ensuring minimal wait time for them. The hybrid system is evaluated under realistic traffic conditions, and it is compared to fixed-timer systems to measure the effectiveness of the hybrid system.

LITERATURE REVIEW

The rapid advancement of AI systems has attracted considerable attention in the field of traffic lights, driven by the limitations of fixed-timer systems and the development of better machine learning, computer vision, reinforcement learning, and LLMs. Existing literature primarily analyzes the effectiveness of some AI-driven systems: rule-based approaches to traffic light timings, reinforcement learning systems for adaptive signal timing, sensor-based traffic control, and integration of LLMs to dynamically control traffic lights.

Rule-Based Approach to Traffic Signal Control

Traditional traffic signals operate on fixed timing cycles that give equal green time to each direction, regardless of traffic volume. As an AI-driven alternative, Elbasha and Abdellatif (2025) proposed a rule-based system using YOLO, a real-time object detection model, for vehicle detection from CCTV footage and adjusting signal timings based on the number of vehicles in each direction. Their PyGame simulation demonstrated a 34% improvement in intersection flow compared to static fixed-timer traffic systems, concluding that utilizing computer vision to control signal timings can significantly reduce

congestion and vehicle waiting times in urban areas compared to traditional fixed-timer systems. However, the study does not report dataset metrics such as size or environmental diversity and uses YOLOv3, an outdated model which is superseded by more accurate models such as YOLOv8 and YOLOv11. The present study addresses the limitation conceptually by proposing the integration of YOLOv11 as a deployment component in the future, while focusing on the signal control logic rather than the detection of vehicles.

Reinforcement Learning for Adaptive Traffic Signal Control

Reinforcement learning in traffic light systems offers the ability to adaptively control traffic signal timings without the need for predetermined or fixed rules. Liu, Liu, and Chen (2021) utilized an intelligent traffic controller system applying AIoT technologies and multi-agent Q-learning across several interconnected intersections. This system accounted for real-world traffic constraints, such as incorporating pedestrian and bicycle flows, to ensure practical application. Simulation results demonstrated that the proposed solutions outperformed existing approaches in terms of vehicle and pedestrian queue lengths, vehicle wait time at intersections, and several other performance metrics. Nevertheless, the study relied on simulated pedestrian arrival rates rather than real-world movement data and assumed idealised traffic conditions without accounting for unexpected events such as accidents. The proposed study partially addresses these limitations by incorporating emergency vehicle prioritization into adaptive traffic signal control, while recognizing that additional factors, such as real-world traffic data and unexpected events, remain areas for future work.

Azfar et al. (2024) proposed a co-simulation framework integrating CARLA (Car Learning to Act) and SUMO to make an accurate, large-scale traffic flow simulation. Cameras were used alongside YOLO detection to send vehicle counts to a multi-agent reinforcement learning system that controlled signal timing in SUMO. Results showed that agents performed favourably even under nonoptimal detection conditions, demonstrating the scalability of the framework for real-world use. However, the framework requires CARLA, a computationally expensive simulator which demands high-end hardware, and the multi-agent system significantly increases training complexity without any mechanism to handle edge cases. The proposed study addresses both limitations by employing SUMO without CARLA to reduce computational requirements and integrating an LLM for edge case handling.

Wei et al. (2018) proposed IntelliLight, a deep Q-network approach for single-intersection signal control using real traffic data from Jinan, China. IntelliLight achieved a 38% reduction in average queue length compared to fixed-timer baselines, establishing an early benchmark for deep reinforcement learning in traffic signal control. The present study achieves a 31% reduction in the average number of halting vehicles per step compared to the optimal fixed timer. This slightly lower improvement may be due to the use of synthetic traffic data and a simpler single-intersection configuration, whereas IntelliLight was evaluated using real-world traffic patterns and data.

Genders and Razavi (2016) proposed a traffic control system applying deep Q-networks (DQN) to single-intersection traffic signal control in the SUMO simulation environment. Their results demonstrated that the reinforcement learning controller reduced congestion-related measures compared to traditional signal control methods such as fixed-timed systems. The present study similarly applies a reinforcement learning approach for adaptive signal control, using PPO rather than DQN due to PPO's training stability and performance in complex decision-making environments. Both studies demonstrate the ability of reinforcement learning to improve overall traffic efficiency compared to traditional signal timing methods.

Chen et al. (2020) proposed MPLight, a pressure-based reward formulation for multi-intersection reinforcement learning evaluated on real traffic networks from New York City. Their work demonstrated the effectiveness of pressure-based reinforcement learning for coordinating traffic signals across large urban networks. The pressure-based reward is different from this study's delta-delay reward formulation, and the multi-intersection evaluation scope highlights a limitation of the proposed single-intersection approach that should be addressed in the future.

Oroojlooy et al. (2020) proposed AttendLight, an attention-based deep reinforcement learning architecture that can be applied to intersections with different road layouts. By incorporating attention mechanisms, the model effectively coordinated traffic signal control under various road configurations. AttendLight's adaptability across various intersection layouts is a capability the present study does not address, highlighting a possible future development for the proposed model.

Delavari, Khanzada, and Kwon (2025) conducted a survey of approximately 100 reinforcement learning studies evaluated in the CARLA simulator, identifying Proximal Policy Optimization (PPO) as one of the most widely used algorithms because of its stable policy optimization and broad applicability. This survey provides context for the algorithm selected in the proposed study and connects its methodology within current reinforcement learning research.

Donatus, Ter, and Udekwe (2025) surveyed multi-agent reinforcement learning applications in intelligent transportation systems, highlighting various challenges such as scalability, communication constraints, and the difficulty of transferring solutions in simulation environments to real-world environments. Their study underscores the need for adaptive decision-making methods in unexpected traffic scenarios. The proposed study contributes to this area by utilizing a hybrid PPO-LLM architecture that allows emergency vehicle prioritization within a simulated urban intersection.

Large Language Models in Traffic Control

LLMs have demonstrated exceptional capabilities in reasoning and decision-making across various domains. Recent research exploring the applications of LLMs aims to explore the potential of these models for enhancing the adaptability and intelligence of traffic control systems. Zhang et al. (2025) proposed LATS, a LLM-assisted teacher-student framework for multi-agent reinforcement learning in

traffic signal control. LATS uses an LLM as a teacher to capture semantic features during training and a student network to emulate these features, enabling the final model to operate independently from the LLM during real-world deployment. Experiments across various traffic datasets demonstrated that LATS outperforms traditional RL and LLM-only systems across several performance metrics. However, the LLM only being used for training inhibits the model's ability to dynamically reason about edge cases such as traffic accidents or emergency vehicles.

Lai et al. (2023) proposed LLMLight, a framework that uses GPT-4 to generate reward functions for reinforcement learning-based traffic signal controllers. LLMLight demonstrated that LLM-generated reward functions outperformed manually designed rewards on congestion-related metrics, particularly average queue length. This supports the value of integrating large language models into adaptive traffic control systems and provides motivation for the hybrid PPO-LLM architecture proposed in the present study.

Li et al. (2025) proposed REG-TSC, a retrieval-augmented distributed LLM agent framework designed for generalizable traffic signal control with consideration for emergency vehicles. The framework integrates emergency scenario reasoning and RL-based optimisation to improve LLM decision-making across various intersection environments. Evaluations across three real-world road networks containing 17 to 177 intersections demonstrated improvements over existing approaches, including a 83.16% reduction in emergency vehicle waiting time. However, the framework focuses on LLM agents operating across large-scale networks, whereas the proposed study evaluates a hybrid PPO-LLM architecture for emergency vehicle prioritization at a single urban intersection, highlighting multi-intersection control as an important future development for the model.

Wei et al. (2026) proposed Virtual Traffic Police, an LLM-augmented traffic signal control framework that was designed to adapt during unforeseen traffic incidents. Instead of replacing existing traffic controllers, the framework uses an LLM-based agent to adjust signal parameters in response to unexpected events such as accidents and road maintenance. Experimental results demonstrated that the framework improved traffic management performance and reliability compared with traditional adaptive traffic signal control methods, reducing average delay from 97.6 seconds to 0 seconds in an ambulance priority scenario with high traffic demand. However, the framework focuses on adjusting existing controllers during unexpected incidents rather than directly integrating LLM reasoning with an RL-based controller. The proposed study addresses this limitation by combining a PPO agent with an LLM component for emergency vehicle prioritization.

Guo et al. (2026) proposed CuraLight, an LLM-based traffic signal control framework that combines reinforcement learning with LLM decision-making. Evaluations across real-world traffic networks, including Jinan, Hangzhou, and Yizhuang, demonstrated a reduction of 7.02% in average waiting time compared to baseline approaches. The proposed study demonstrates a 16% reduction in average waiting time per vehicle compared with the optimal fixed timer, while prioritizing emergency vehicles through the integration of an LLM.

RESEARCH GAPS

- Existing rule-based traffic control systems rely on outdated detection models such as YOLOv3, resulting in lower accuracy and reduced efficiency under complex traffic conditions.
- Multi-agent reinforcement learning approaches introduce high computational complexity, making them difficult to scale and deploy in resource-constrained environments.
- Absence of efficient optimization techniques to reduce training time and computational cost in reinforcement learning-based systems.
- Existing models lack robust mechanisms to handle edge cases such as emergency vehicle prioritization.
- Large language model-based frameworks such as the LATS framework are primarily used during training and not in real-time deployment, restricting dynamic decision-making.
- Insufficient exploration of hybrid approaches combining rule-based methods, reinforcement learning, and LLMs into a unified traffic control system.
- Lack of explainability and interpretability in AI-driven traffic systems, making it difficult to justify decisions in critical situations.

RESEARCH OBJECTIVES

- To explore the effectiveness of implementing reinforcement learning to create an adaptive traffic control system and evaluate its ability to outperform fixed-timer systems in urban traffic conditions.
- To investigate the integration of Large Language Models to create a hybrid traffic control system and examine whether real-time LLM reasoning can improve edge case handling without compromising traffic efficiency.
- To provide a comparison between fixed-timer, reinforcement learning, and hybrid control systems using traffic performance metrics to evaluate the relative effectiveness of these systems to optimise traffic flow.

METHODS

Simulation Environment

To test the effectiveness of traffic control systems, a simulation environment capable of modelling realistic traffic flow is required. This study utilizes the Simulation of Urban MObility (SUMO), an open-source, microscopic, and continuous traffic simulator (Lopez et al., 2018). Unlike macroscopic simulators that focus on aggregate traffic flow and system-wide dynamics, SUMO provides a detailed analysis of traffic flow at intersections and simulates each vehicle using car-following models that reproduce realistic driving behaviours.

This study employs SUMO due to its computational efficiency compared to other traffic simulations, such as CARLA (Dosovitskiy et al., 2017). While CARLA offers a comprehensive 3D environment suitable for testing the effectiveness of various traffic light systems, it introduces unnecessary complexity for single-intersection traffic optimisation, where vehicle detection is handled programmatically rather than through computer vision. Additionally, CARLA relies on Unreal Engine to render a realistic three-dimensional environment and recommends a GPU with at least 8 GB of VRAM (CARLA Team, 2024), while SUMO is a lightweight microscopic simulator that primarily relies on CPU-based computation, making it substantially less computationally demanding than CARLA.

The simulation environment consists of a four-way signalised intersection with roads from four directions, each road having two lanes in each direction. The North-South road and East-West road are both 400 m long, with the intersection dividing both roads into two equal 200 m sections. Vehicles from all four directions are allowed to either go straight or turn right. The simulated roads operate under an urban speed limit of 50 km/h, representing a typical speed limit for urban roads with signalized intersections. The environment was designed to represent a generalized urban intersection rather than a specific city, allowing the proposed traffic control system to be evaluated under common urban traffic conditions.

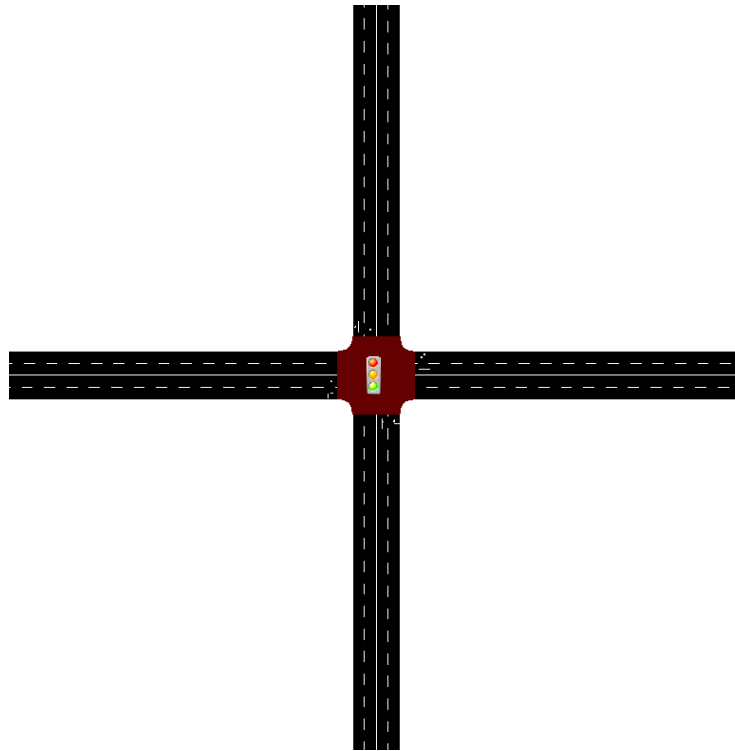


Fig. 1: SUMO Simulation Environment of a four-way intersection with traffic lights.

The traffic light that controls the intersection has two main signal phases with a five-second transition phase between the following main phases:

NS green - North and South roads have right of way
NS yellow - five-second transition period
EW green - East and West roads have right of way
EW yellow - five-second transition period

The two-phase system reflects a standard urban intersection configuration and provides the AI agent with a decision at each step: maintain the current phase or switch to the next phase.

The traffic flow was manually adjusted during specific intervals across a 3600-second simulation period to represent realistic urban traffic conditions. Three different traffic flow periods were implemented in the simulation. North-South and East-West refer to the two opposite approach roads: vehicles travelling in the North-South direction enter from either the North or South approach and exit through the opposite side, while East-West vehicles follow the same pattern but travel horizontally.

Morning rush (0-1200s)

This represents the period in a day when the majority of vehicles travel in one direction. North-South vehicle flow is set to 1200 vehicles per hour, while East-West traffic flow is set to 300 vehicles per hour, creating a four-to-one direction imbalance that tests the performance of the adaptive traffic system compared to fixed timers: This 80/20 directional split is slightly above the range typically documented in traffic engineering practice, where the directional distribution is frequently around 60-70% (Texas Department of Transportation, 2024). The split was chosen intentionally to test the system's capability to handle a more severe traffic load than the range typically documented in traffic engineering practice.

Midday balanced (1200-1400s)

This represents the period in a day when the number of cars travelling from North-South and East-West is relatively the same. Both directions are set to 600 vehicles per hour, testing the agent's ability to optimise traffic flow for both directions equally and efficiently.

Evening rush (2400-3600s)

This represents the period in a day when the majority of vehicles travel in the direction opposite to that of the Morning Rush. East-West vehicle flow is set to 1200 vehicles per hour, while North-South traffic flow is set to 300 vehicles per hour. This additional 80/20 directional split is implemented to make sure the agent is not biased towards a specific direction.

The simulation includes cars, trucks, buses, and emergency vehicles, each with varying acceleration rates and maximum speeds. Two ambulances and two police cars are used to measure the LLM's ability to handle edge cases. The emergency vehicles enter the intersection at $t=600s$, $t=1400s$, $t=2200s$, and $t=3000s$, respectively. These vehicles are introduced at different time periods throughout the simulation to test the LLM in the hybrid model's capability to detect and respond to emergencies during varying traffic density and at random times.

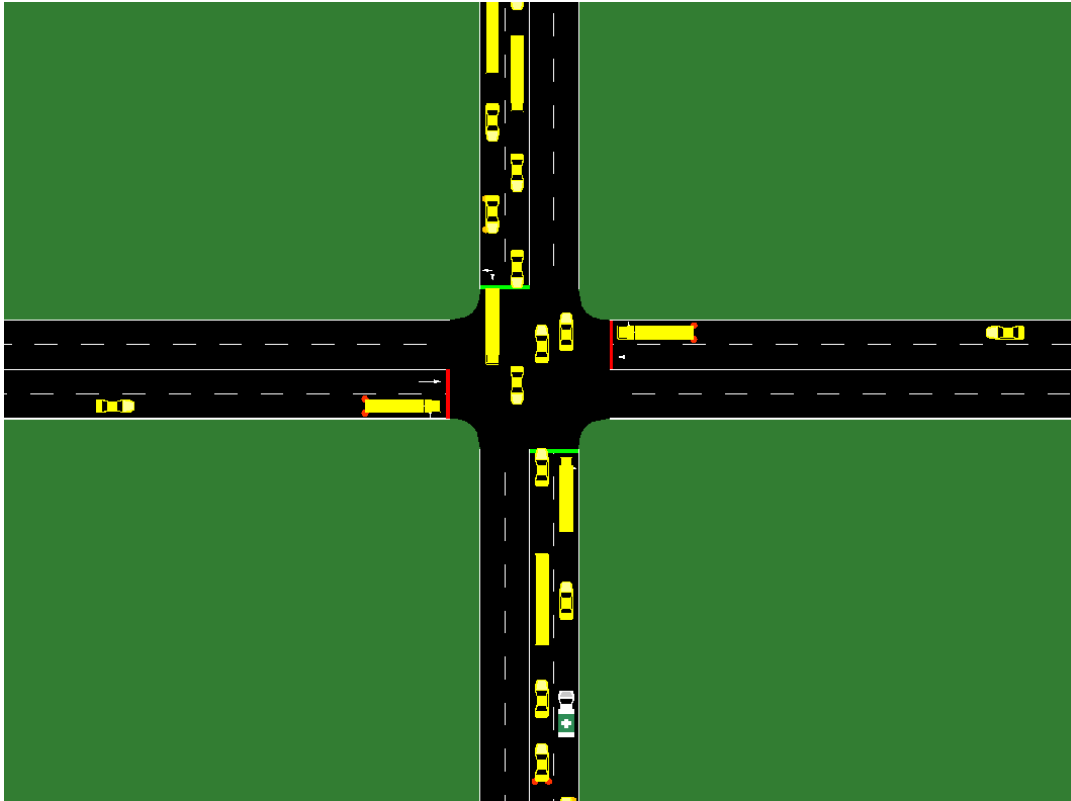


Fig. 2: White emergency vehicle entering the simulation from the South.

Baseline Fixed Timer

Traditional traffic signal systems use fixed timing cycles, providing each direction with equal green and red time regardless of the number of cars on the road. To ensure that the reinforcement learning model is being compared by an effective fixed-timer system, six fixed green time durations were evaluated: 10, 15, 20, 25, 30, and 42 seconds. These fixed green times were selected as they represent a broad range of short to moderate signal timings for comparative evaluation. The 42-second time configuration was included since it falls within the typical range of green times used in real-world urban intersections (Koonce, 2008). Testing multiple time configurations rather than a single configuration ensures that the reinforcement learning agent is being compared to the best fixed-timer performance.

Reinforcement Learning Agent

Reinforcement learning is a form of machine learning where an agent learns to make decisions by adapting to its learning environment. The agent observes the current state, chooses an action, receives a reward indicating how good or bad that action was, and updates its behaviour to maximise the reward it receives. Unlike supervised learning, which requires labelled data, an RL agent learns through trial-and-error. The unpredictable nature of traffic flow and conditions makes the RL model well-suited to optimise traffic light timings.

Proximal Policy Optimisation (PPO) was selected as the reinforcement learning algorithm for this study (Schulman et al., 2017). PPO is an on-policy actor-critic algorithm that comprises two neural networks: an actor that decides which action to take, and a critic that evaluates how good the action was. A notable feature of the PPO algorithm is its clipping mechanism, which limits the extent of policy updates. This prevents drastic changes that could destabilise the learning capability while training. Additionally, PPO addresses a serious problem that other reinforcement learning models face, called catastrophic forgetting: when the model's policy changes so drastically that previously learned behaviour is forgotten. PPO addresses this by limiting the ratio between the new and old policy to a 20% range, ensuring steady improvements rather than drastic and risky changes in policy.

PPO was chosen over alternatives, such as Q-learning, due to its relatively uncomplicated implementation compared to off-policy models and its stability during training. Trust Region Policy Optimisation (TRPO) is a closely related reinforcement learning model that has stable policy updates through trust region constraints and is considered the baseline for PPO. However, TRPO requires computationally expensive second-order optimization methods such as conjugate gradient descent, making it significantly more complex and less efficient to implement compared to PPO's first-order gradient descent and clipping mechanism.

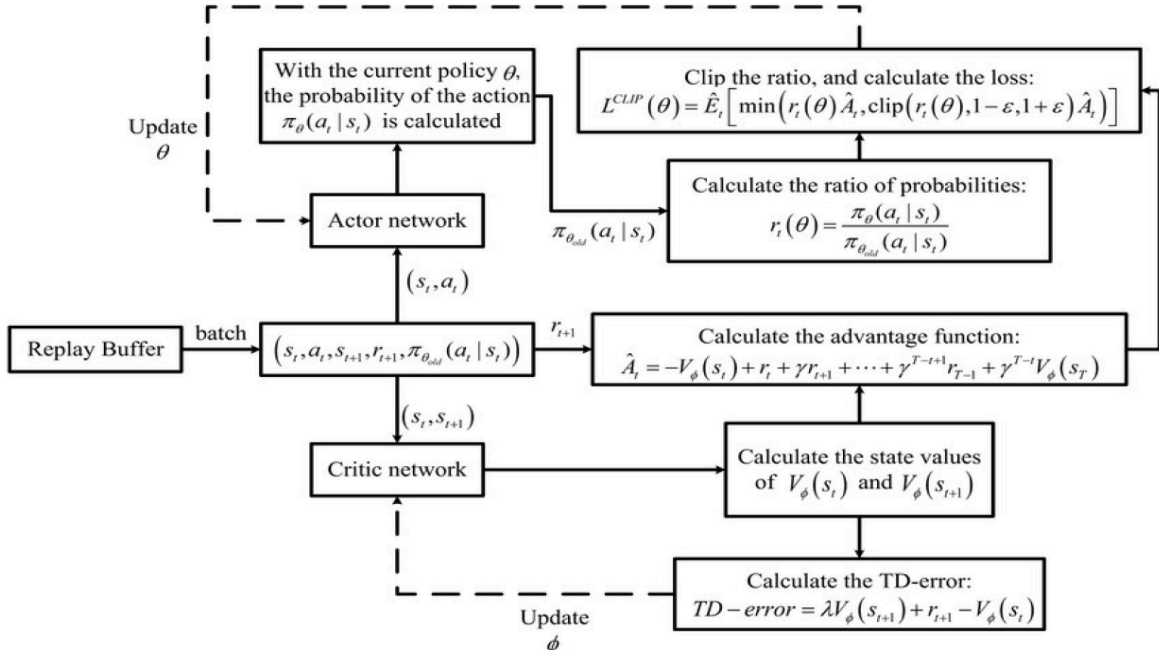


Fig. 3: PPO Reinforcement Learning Framework. Reproduced from Shi et al. (2022).

Figure 3 illustrates the training process of the PPO reinforcement learning algorithm. Firstly, a batch of experiences is sampled from the replay buffer and provided to both the actor and critic networks. The actor network uses the current policy to calculate the probability of selecting each action, while the critic network estimates the value of the current and next states. These value estimates are used to calculate the temporal-difference (TD) error and the advantage function, which measure how beneficial each action was compared to the expected outcome. The ratio between the current and previous policy probabilities is then calculated and used to compute the PPO loss function. Then, the loss is clipped to limit the magnitude of policy updates before the actor network parameters are updated, while the critic network is updated using the TD error. In this study, the environment shown in the framework corresponds to the SUMO traffic simulation, where the selected actions determine the traffic signal phase at the intersection.

States

At each decision step, the agent observes the intersection through a 19-dimensional state vector provided by the SUMO-RL environment.

Vector Component	Number of Values	Description
Phase one-hot encoding	2	A one-hot encoding vectoring which indicates the current active green phase.
Minimum green met flag	1	A binary variable indicating whether the current green phase has been active for the minimum green time (15 in this case).
Vehicle density per lane	8	The normalised count of all vehicles present on each incoming lane across all four directions.
Queue length per lane	8	The normalised count of all halting vehicles on each incoming lane. A vehicle is halting if its speed is below 0.1 metres per second.

Table 1: Reinforcement Learning State Vector Components.

Actions

The agent selects an action to make based on the 19-dimensional state vector at each decision interval of 10 seconds:

Action	Value	Description
Keep the current traffic light phase	0	The agent maintains the current active light phase, allowing vehicles in the direction with a green light to continue moving through the intersection.
Switch the current traffic light phase	1	The agent switches the traffic light phase, giving the opposite direction the green light after a 5-second yellow transition phase.

Table 2: Reinforcement Learning Action Space.

Three constraints are enforced on the RL agent to ensure safe and realistic traffic signal operation:

- The agent cannot switch if the current phase has been active for less than 15 seconds, ensuring vehicles have sufficient time to clear the intersection safely.
- The traffic light has a yellow transition phase of 5 seconds when the agent changes phases.
- A phase cannot remain active for more than 60 seconds, preventing queue buildup in the opposing direction.

Reward Function

Every 10 seconds, the agent receives a reward score that is calculated using the following reward function:

$$R = -(\Delta D + 0.3Q + 1.0S)$$

The function has three components:

- **Change in delay (ΔD):** SUMO tracks the accumulated waiting time of every vehicle in the simulation at each simulation second. If the total increases, more vehicles are waiting at the intersection, so the situation worsens. If it decreases, fewer vehicles are waiting to cross the intersection, so the situation improves. The change rather than the total is used because the total accumulated delay always grows over time and provides no useful information about whether the agent's action was beneficial or harmful.
- **Queue length ($0.3Q$):** Counts the total number of vehicles currently stopped across all four directions. Even if the delay did not change drastically in a single step, a growing queue is an early warning signal of building traffic congestion.
- **Switching penalty ($1.0S$):** A penalty of 1.0 is applied every time the agent switches the signal phase. In real-world intersections, switching is not instantaneous: vehicles require time to accelerate from rest, and drivers need reaction time to respond to a change in the traffic light. Punishing unnecessary switching encourages the agent not to switch phases erratically.

Hyperparameters

Parameter	Value	Description
Learning rate	0.0003	Controls the size of each weight update. A conservative value that ensures stable incremental learning without overshooting.
Rollout buffer size	720	Number of simulation steps collected before each training update.

Minibatch size	72	Number of experiences sampled per update pass. Represents 10% of the buffer, balancing update quality and computational cost.
Total timesteps	100,000	Total simulation steps across the entire training process
Discount factor	0.99	Determines how much the agent values future rewards relative to immediate ones. A value near 1 encourages long-term planning, appropriate for traffic control, where clearing queues early prevents larger queues from forming later.
Clip range	0.2	Maximum permitted policy change per update. PPO's clipping mechanism limits the old-to-new policy ratio to 1:1.2, ensuring stable incremental improvement.
Delta time	10s	Decision interval (how frequently the agent observes the intersection and selects an action)
Yellow time	5s	Duration of the yellow transition phase between the two main green phases.
Minimum green	15s	Minimum active green light time before the agent is permitted to switch.
Maximum green	60s	Maximum time a green light can remain active before the agent forces a transition.

Table 3: PPO Training Hyperparameters.

Hybrid System

The hybrid system combines the PPO reinforcement learning model with a Claude Sonnet 4.6 LLM to handle edge cases that the RL agent is not able to efficiently detect on its own. Under normal traffic conditions, the hybrid model operates identically to the RL agent model; the LLM is activated only when an emergency vehicle is detected in the simulation. This preserves the computational efficiency of the RL agent during operation.

At every decision step, the hybrid model checks all vehicles in the simulation for any vehicle classified as an emergency vehicle. When an ambulance or police car enters the simulation, it is detected, and the LLM is activated. The LLM uses the current vehicle count in each direction and the location of the emergency vehicle to determine the appropriate phase and duration for the traffic light. The LLM ensures that the emergency vehicle is never stopped at the intersection when making its traffic phase and duration

decision. Each emergency vehicle triggers the LLM only once, preventing unnecessary interventions for the same vehicle.

The LLM receives the following natural language prompt template from the user describing the simulation environment and the current vehicle counts per direction:

You are an AI traffic light controller. EMERGENCY: [vehicle type] detected on [direction] approach. Current vehicles: N=[count] S=[count] E=[count] W=[count]. The emergency vehicle must not be stopped. Determine required green light time for [direction]. Reply in exactly this format: PHASE: NS or EW / DURATION: 60 / REASON: one sentence summary.

It responds with a phase decision, phase length and a logical reasoning behind its decision. After the LLM makes the decision, the control of the traffic light system goes back to the RL agent.

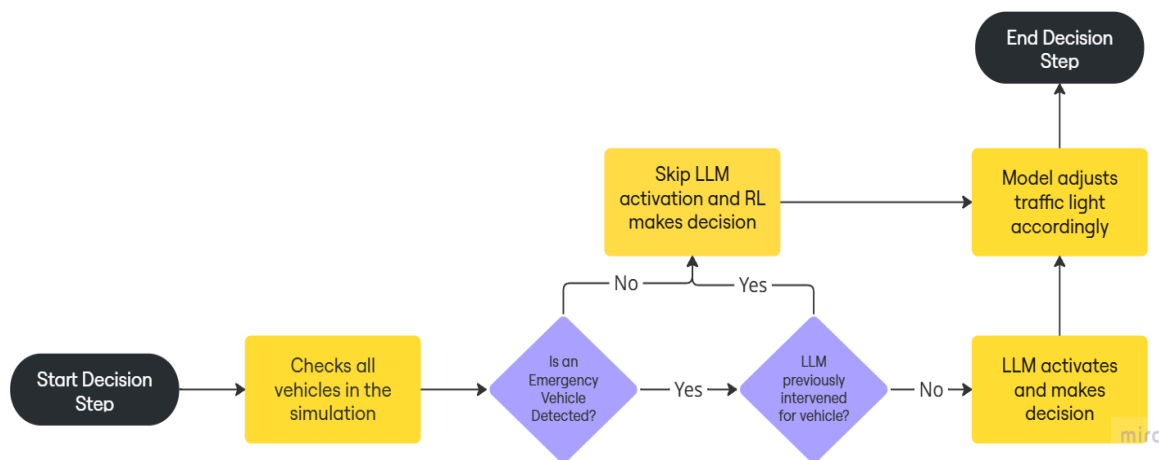


Fig. 4: Hybrid RL + LLM Agent Decision Flow.

Evaluation Metrics

Two evaluation metrics were used to compare the models' effectiveness in optimising traffic flow:

- **Average Halting Vehicles Per Step (AHVPS):** The average number of all halting vehicles across all four directions at each simulation second. A vehicle is classified as halting if its speed is below 0.1m/s.

AHVPS accurately provides the traffic congestion level in an intersection, and it is not affected by vehicle counting limitations present in the simulation environments. Lower AHVPS represents fewer vehicles waiting at an intersection, therefore indicating lower traffic congestion.

- **Average Waiting Time Per Vehicle (AWTPV):** SUMO captures the total accumulated time each vehicle spends halting at the intersection. The mean value across all vehicles is used for this metric

AWTPV effectively captures each driver's experience in the intersection since it is calculated through SUMO's built-in function that records a vehicle's time spent with a speed below 0.1 m/s. The average of this across all vehicles highlights the average driver experience while driving through the intersection, indicating the congestion level of the intersection. Lower AWTPV represents a better driver experience since each vehicle experiences less halting, illustrating lower traffic congestion in the intersection. Additionally, AWTPV for specifically emergency vehicles was also used as an evaluation metric to demonstrate the hybrid system's effectiveness in handling edge cases compared to the standalone reinforcement learning agent.

RESULTS AND DISCUSSION

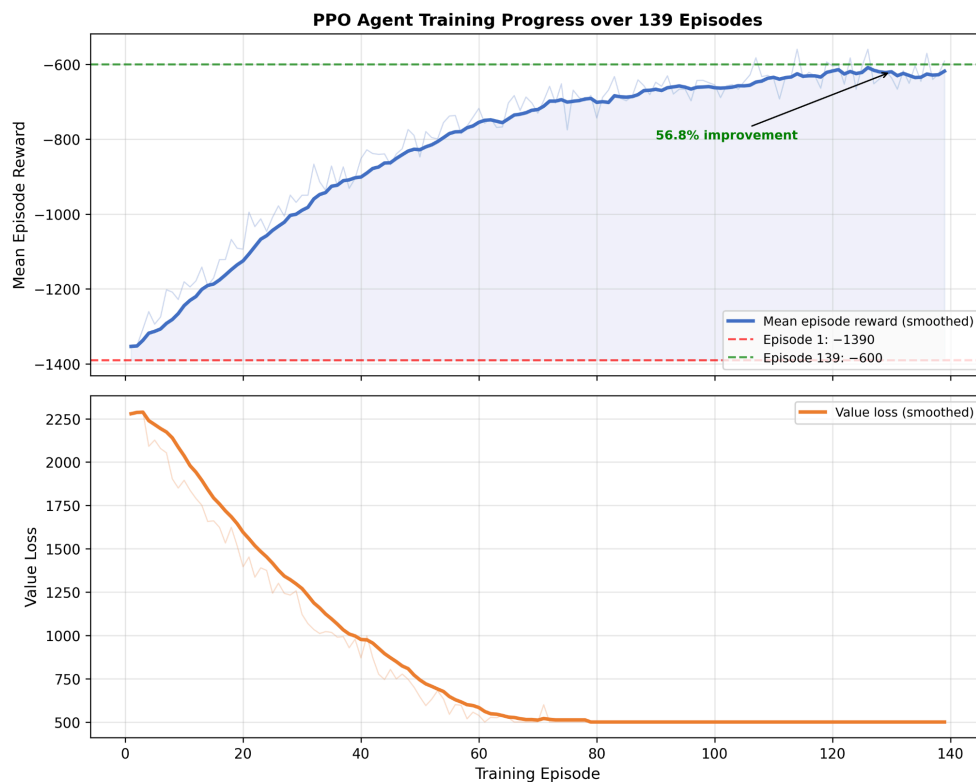


Fig. 5: PPO Training Performance.

To evaluate the learning progression of the PPO RL agent, training performance was recorded across 139 simulation episodes. Figure 5 illustrates the change in episode reward throughout the learning process, representing a 56.8% improvement. The increasing reward indicated by the graph demonstrates that the PPO agent successfully learned an effective traffic signal control policy. The bottom graph illustrates value loss decreasing over training, indicating the critic network became more accurate at predicting outcomes over time.

Each traffic control system was tested under the traffic conditions mentioned in the simulation environment across the full 3600-second simulation. The results are the comparison between the fixed-timer configurations, the RL agent, and the hybrid model performance on our chosen evaluation metrics.

System	Average halting vehicles/step	Average waiting time/vehicle (s)	Comment
Fixed 10s	27.51	36.27	Fixed 10s green light timer results in 27.51 average halting vehicles/step and average wait time of 36.27s, significantly worse than all other traffic systems tested.
Fixed 15s	7.52	8.99	Fixed 15s green light timer results in 7.52 average halting vehicles/step and 8.99s average wait time. Significant improvement over 10s
Fixed 20s	7.27	8.63	Fixed 20s green light timer results in 7.27 average halting vehicles/step and 8.63s average wait time. Best performing fixed timer.
Fixed 25s	8.08	9.66	Fixed 25s green light timer results in 8.08 average halting vehicles/step and 9.66s average wait time. Performs worse as green light time keeps increasing.

Fixed 30s	9.65	11.81	Fixed 30s green light timer results in 9.65 average halting vehicles/step and 11.81s average wait time. Further increase results in worse performance in both metrics.
Fixed 42s	12.19	15.21	Fixed 42s green light timer results in 12.19 average halting vehicles/step and 15.21s average wait time. Poor performance, even though it is considered a commonly deployed timer.
RL PPO agent	5.02	7.25	Our adaptive traffic system using the RL agent results in 5.02 average halting vehicles/step and 7.25s average wait time, outperforming all fixed timer systems.
Hybrid Model	5.16	7.55	The hybrid model results in 5.16 average halting vehicles/step and 7.55s average wait time, slightly worse than the RL agent.

Table 4: Comparison of Traffic Signal Control Performance.

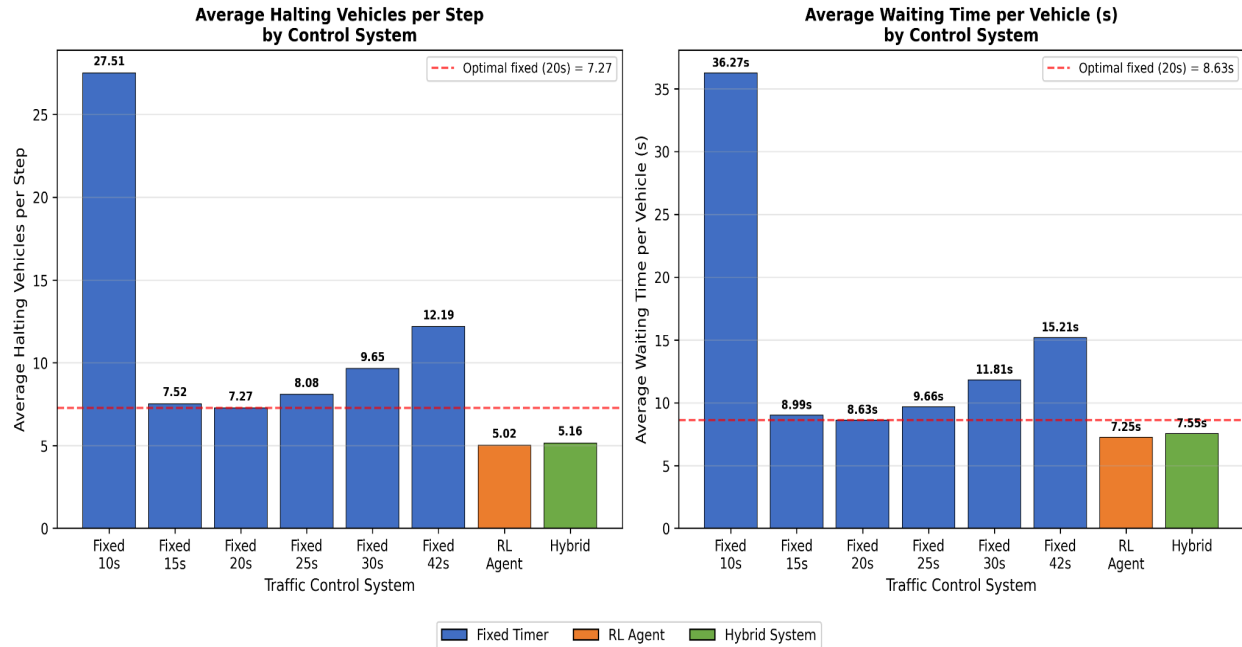


Fig. 6: Performance Comparison Using Average Halting Vehicles and Average Waiting Time.

These results show clear differences in performance between the various traffic configurations tested, which can be explained by the underlying characteristics of each traffic system:

Fixed-timer Configurations Evaluation

The fixed-timer results, as can be seen in Figure 6, demonstrate that the 20-second green light timer minimized average waiting time and halting vehicles compared with other fixed-timer configurations. Timers shorter than 15 seconds can not clear traffic queues before switching phases, resulting in queue buildup in the direction with heavy traffic flow. Timers longer than 20 seconds stay active after the queue clears up, resulting in inefficient traffic light phase-switching. Even though the 42-second fixed timer is supposed to be one of the more effective fixed timers for real-world traffic, it performed worse than most of the other fixed timers. It is worth noting that this may be due to our simulation not capturing left turns and human error.

Standalone Reinforcement Learning Agent Evaluation

The RL PPO agent outperforms the best fixed-timer configuration, lowering average halting vehicles per step by 31% and average waiting time per vehicle by 16%. The noticeable improvement over the fixed

timer configurations highlights the capability of integrating reinforcement learning in traffic systems rather than using an ineffective fixed-timer system. This performance gain is likely due to the agent's ability to continuously adapt its decisions based on real-time traffic conditions. Unlike fixed timer configurations, which follow a predetermined set of rules regardless of traffic conditions, the PPO agent learns traffic patterns through interaction with its environment and adjusts signal phases accordingly. This allows the agent to respond dynamically to fluctuations in traffic flow, reduce congestion and vehicle delays more effectively than fixed-timer systems.

Hybrid System Evaluation

The hybrid system achieves comparable performance to the standalone RL PPO agent, increasing average halting vehicles per step by 3% and average waiting time per vehicle by 4%. The hybrid model demonstrated slightly worse metrics compared to the RL PPO agent. This is reasonably due to traffic flow efficiency, as measured by these metrics, being sacrificed when prioritising emergency vehicles.

To evaluate the hybrid system's capability to handle emergency vehicles, the waiting time of each emergency vehicle was measured under both the standalone RL agent and the hybrid system.

Type	RL Mean Wait Time (s)	RL Standard Deviation (s)	Hybrid Mean Wait Time (s)	Hybrid Standard Deviation (s)
Ambulance	8.0	1.5	0.0	0.0
Police	4.0	1.0	0.0	0.0

Table 5: Emergency Vehicle Waiting Time Comparison.

The hybrid model was able to allow the emergency vehicles to go through the intersection without experiencing any delays, whereas the standalone RL agent still resulted in some waiting time for the emergency vehicles at the intersection. This represents a substantial improvement over the fixed-timer configurations, where emergency vehicles are generally required to wait for the signal cycle or rely on external prioritization mechanisms. These results highlight a crucial factor in adaptive traffic light systems that standard traffic metrics do not fully capture. In real-world scenarios, the time an emergency

vehicle waits at an intersection directly affects response times for medical emergencies and law enforcement, where seconds saved can potentially save lives. The standalone RL agent, while effective at optimising traffic under regular conditions, is incapable of efficiently adjusting signal timings for emergency vehicles, which could lead to unnecessary delays caused by waiting at intersections. The hybrid system eliminates this problem by utilizing an LLM that prioritizes emergency vehicles to guarantee no wait time for them at intersections. The combination of strong regular traffic control from the RL agent and reliable emergency handling from the LLM demonstrates the potential of deploying hybrid AI systems to meaningfully improve traffic control.

Limitations

Several limitations should be considered when interpreting the results of this study. The simulation environment is based around one urban intersection, while real urban traffic systems consist of multiple interconnected networks, where one intersection's traffic signal decisions can affect other intersections. Therefore, the performance improvements observed in this study might not transfer to real-world scenarios without adjustments to account for the interconnectedness of traffic networks. Additionally, the traffic demand in the simulation was synthetically generated rather than obtained through the use of real-world datasets, which may limit the simulation's ability to represent a specific real-world intersection. Moreover, the LLM component uses API-based communication, which introduces potential latency that was not fully considered in the simulation. Real-world implementation would require the consideration of the LLM's response time to emergency vehicles entering the intersection.

Practical Implications and Deployment Considerations

The hybrid system proposed in this study demonstrates potential for improving both traffic efficiency and emergency response time. The system could be most beneficial in high-volume urban intersections with changing traffic patterns, especially near emergency facilities where emergency vehicle prioritization is important. Deployment for the system would likely require connecting the system with existing traffic controllers. The PPO RL agent can make signal decisions with relatively low computational requirements after training. However, the LLM component introduces more computational requirements and additional latency, which might require edge computing solutions in order to ensure the system runs reliably. The additional cost of deployment would primarily come from the edge computing hardware and software integration, making the system more practical for gradual adoption. Policymakers should consider initially implementing this system at intersections with high traffic congestion or near emergency facilities to evaluate its effectiveness before using it across larger urban networks.

Future Work

Future studies should evaluate the hybrid system across multiple intersections with multiple traffic agents to demonstrate the system's performance in city-wide traffic networks and examine its ability to manage more complex traffic behaviours, such as lane selection and turning movements. Additionally, future studies could explore LLM behaviour on other conditions, such as harsh weather conditions, traffic accidents or road closures. A vehicle detection component, such as YOLOv11, that provides the hybrid model with data from real-world traffic cameras should be integrated into the study when deployed in practical environments. Moreover, future research studies could evaluate the effectiveness of other reinforcement learning algorithms and large language models to possibly improve traffic flow and edge case handling.

REFERENCES

- Anthropic. (2026). Claude Sonnet 4.6 [Large language model]. <https://www.anthropic.com>
- Azfar, T., Huang, K., Tracy, A., Misiewicz, S., Liu, C., & Ke, R. (2024). Traffic co-simulation framework empowered by infrastructure camera sensing and reinforcement learning. *Journal of Intelligent Transportation Systems*. <https://arxiv.org/abs/2412.03925>
- CARLA Team. (2024). CARLA documentation: Quick start. https://carla.readthedocs.io/en/0.9.16/start_quickstart/
- Chen, C., Wei, H., Xu, N., Zheng, G., Rong, Y., Qian, Z., Yang, L., & Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4), 3414–3421. <https://doi.org/10.1609/aaai.v34i04.5744>
- Delavari, E., Khanzada, F. K., & Kwon, J. (2025). A comprehensive review of reinforcement learning for autonomous driving in the CARLA simulator. *arXiv*. <https://arxiv.org/abs/2509.08221>
- Donatus, R., Ter, K., Ajayi, O.-O., & Udekwe, D. (2025). Multi-agent reinforcement learning in intelligent transportation systems: A comprehensive survey. *arXiv*. <https://arxiv.org/abs/2508.20315>
- Dosovitskiy, A., Ros, G., Codevilla, F., López, A. M., & Koltun, V. (2017). CARLA: An open urban driving simulator. *Proceedings of Machine Learning Research*, 78, 1–16. <https://proceedings.mlr.press/v78/dosovitskiy17a.html>
- Elbasha, A. M., & Abdellatif, M. M. (2025). AIoT-based smart traffic management system. *CS & IT – CSCP 2025*, 69–77. <https://arxiv.org/abs/2502.02821>
- Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. *arXiv*. <https://arxiv.org/abs/1611.01142>
- Guo, Q., Li, X., Chen, J., Guo, Z., Xu, S., Zhang, L., & Li, L. (2026). CuraLight: Debate-guided data curation for LLM-centered traffic signal control. *arXiv*. <https://arxiv.org/abs/2604.05663>
- Koonce, P. (2008). Traffic signal timing manual. Federal Highway Administration. https://ops.fhwa.dot.gov/publications/fhwahop08024/fhwa_hop_08_024.pdf

- Lai, S., Xu, Z., Zhang, W., Liu, H., & Xiong, H. (2023). LLMLight: Large language models as traffic signal control agents. arXiv. <https://arxiv.org/abs/2312.16044>
- Levy, J., Buonocore, J., & von Stackelberg, K. (2010). The public health costs of traffic congestion: A health risk assessment. International Bridge, Tunnel and Turnpike Association. <https://www.ibtta.org/sites/default/files/The%20Public%20Health%20Costs%20of%20Traffic%20Congestion.pdf>
- Li, X., Guo, Q., Chen, J., Guo, Z., Xu, S., Li, L., & Zhang, L. (2025). Retrieval augmented generation-enhanced distributed LLM agents for generalizable traffic signal control with emergency vehicles. arXiv. <https://arxiv.org/abs/2510.26242>
- Liu, Y., Liu, L., & Chen, W. P. (2021). Intelligent traffic light control using distributed multi-agent Q-learning. IEEE Intelligent Transportation Systems Conference. <https://arxiv.org/abs/1711.10941>
- Oroojlooy, A., Nazari, M., Hajinezhad, D., & Silva, J. (2020). AttendLight: Universal attention-based reinforcement learning model for traffic signal control. Advances in Neural Information Processing Systems, 33. <https://proceedings.neurips.cc/paper/2020/hash/29e48b79ae6fc68e9b6480b677453586-Abstract.html>
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv. <https://arxiv.org/abs/1707.06347>
- Texas Department of Transportation. (2024). Roadway design manual: Traffic characteristics. <https://www.txdot.gov/manuals/des/rdw/chapter-4--basic-design-criteria/4-3-traffic-characteristics/4-3-1-traffic-volume.html>
- UCLA Institute of Transportation Studies. (2025, September 29). Traffic congestion. <https://its.ucla.edu/news/for-the-press/traffic-congestion/>
- Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). IntelliLight: A reinforcement learning approach for intelligent traffic light control. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2496–2505. <https://dl.acm.org/doi/10.1145/3219819.3220096>
- Wei, S., Wang, Q., & Yang, K. (2026). Virtual Traffic Police: Large language model-augmented traffic signal control for unforeseen incidents. arXiv. <https://arxiv.org/abs/2601.15816>
- Zhang, Y., Li, P., Zhou, T., Fan, M., & Sartoretti, G. (2025). LATS: Large language model assisted teacher-student framework for multi-agent reinforcement learning in traffic signal control. arXiv. <https://arxiv.org/abs/2603.24361>