# TrafficLightRL

Kristian Diana
*McMaster University*
dianak@mcmaster.ca

Tridib Banik
*McMaster University*
banikt@mcmaster.ca

Varun Pathak
*McMaster University*
pathav4@mcmaster.ca

Ryan Li
*McMaster University*
li3018@mcmaster.ca

Clara Wong
*McMaster University*
wongc148@mcmaster.ca

***Abstract*—This paper presents TrafficLightRL, a reinforcement learning (RL)-based traffic light control system designed to reduce vehicle wait times and emissions. Using SUMO for simulation and Stable-Baselines3 for RL training, our system dynamically adapts to real-time traffic conditions. Results show that the RL agent reduces wait times by up to 24% and emissions by up to 29% compared to traditional systems, particularly in medium to high traffic densities. The project demonstrates the potential of RL to improve urban mobility and reduce environmental impact.**

## I. Introduction

As one of the McMaster AI Society's many projects for the 2024/25 season, our team decided to investigate a reinforcement learning solution that could tackle a relevant and significant real-world issue. Out of our passion for the environment, or perhaps our impatience waiting to cross the road when walking to class, we noticed that the current system for handling traffic light signals is inefficient and outdated. Our team of creative and passionate students knew there had to be a better solution to this problem, and thus the idea for TrafficLightRL was born. In this document we will discuss in detail the issue we address with our project, the design process and actual implementation of our solution, the quantifiable results we were able to achieve and our reasoning, and finally next steps for the development of our project.

### *A. Motivation*

Traffic congestion is a major contributor to environmental degradation and inefficiencies in urban mobility. Inefficient traffic light systems, such as fixed-timer schedules and sensor-based controls, often lead to prolonged idling times, increasing vehicle emissions by up to 20% during peak hours [1]. Additionally, drivers in urban areas spend an average of 54 hours per year idling in traffic, wasting time and fuel [2]. These inefficiencies highlight the need for a more adaptive and intelligent approach to traffic management.

Reinforcement learning (RL) offers a promising solution by enabling traffic signals to dynamically adapt to real-time conditions. Unlike traditional systems, RL-based approaches can reduce both waiting times and emissions by learning optimal control policies through interaction with the environment. This project explores how an RL-based traffic light control system can provide a more sustainable and efficient alternative to conventional methods, addressing both environmental and everyday concerns.

### *B. Related Works*

Traditional traffic light systems, such as fixed-timer schedules and sensor-based controls, rely on predefined rules and historical data. While simple to implement, these systems cannot adapt to real-time changes in traffic flow, leading to inefficiencies during unexpected congestion or accidents [3]. Adaptive systems like SCOOT and SCATS improve upon this by dynamically adjusting signal timings, but they still rely on rule-based optimizations that require manual tuning and struggle with complex traffic scenarios [4].

Reinforcement learning (RL) addresses these limitations by enabling traffic signals to learn and adapt dynamically. RL-based systems do not rely on predefined rules; instead, they optimize signal timings based on real-time data, reducing waiting times and emissions [5]. This adaptability makes RL particularly well-suited for modern traffic management, as it can handle complex aspects of real-world traffic conditions, such as variable traffic density.

### *C. Problem Definition*

This project aims to develop an RL-based traffic light control system that dynamically adjusts signal timings to reduce vehicle waiting times, and lower emissions. As we progress through this document, we will be frequently using the following terms, so here we have provided a brief overview of what they represent:

- **State:** Real-time traffic conditions, including traffic light phases and individual vehicle metrics.
- **Action:** Choosing the next traffic signal phase.
- **Reward:** A combination of reduced waiting time and lower emissions.

The goal is to train an RL model that optimizes traffic flow while adapting to different traffic conditions, providing a more sustainable and efficient solution for modern traffic management.

## II. Methodology

This section describes the design and implementation of our RL-based traffic light control system. We first present the tools and technologies used, followed by the design of the RL agent, the simulation environment, and the evaluation metrics. Finally, we discuss how our approach reflects real-world traffic conditions and ensures generalization.

## A. Tools and Technologies

The project leverages two primary technologies: **SUMO (Simulation of Urban MObility)** and **Stable-Baselines3**. SUMO provides a realistic traffic simulation environment, including real-time visualizations and dynamic traffic scenarios. Stable-Baselines3, a popular reinforcement learning library, is used to train the RL agent. The integration between SUMO and Stable-Baselines3 is facilitated by **TraCI**, an API that enables real-time communication between Python and SUMO. Additionally, we use **OpenAI Gymnasium** to create a consistent interface for the RL agent, abstracting SUMO's functionality into a format compatible with Stable-Baselines3. Figure 1 illustrates the process flow and interactions between these components.

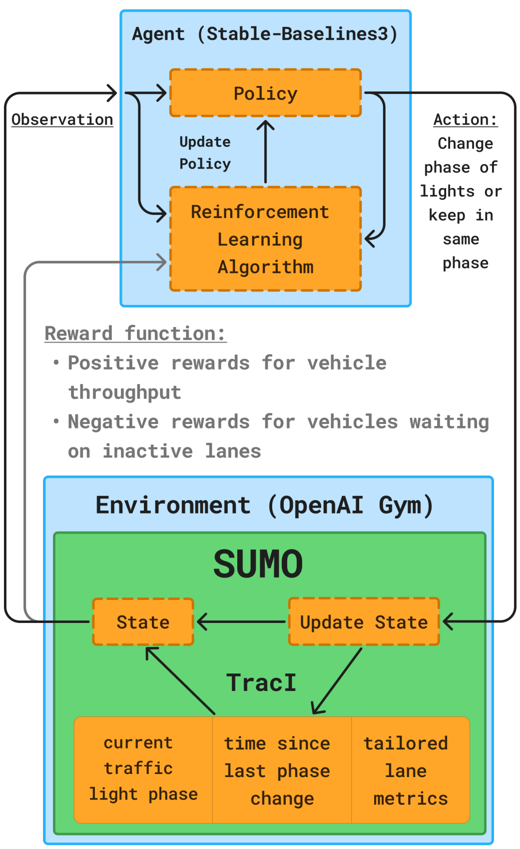


Fig. 1. Process flow diagram illustrating the interaction between SUMO, TraCI, OpenAI Gymnasium, and Stable-Baselines3.

## B. Simulation Environment

The simulation environment is designed to mimic real-world traffic conditions. Each episode represents a fixed time period during which a specific number of vehicles pass through the intersection. To reflect the randomness of real-life traffic, we implement **random vehicle deployments**, ensuring that no two episodes are identical. This promotes generalization, as the agent is exposed to a wide variety of traffic scenarios during training.

The environment enforces standard traffic safety rules, such as requiring the agent to pass through **transition phases** (e.g., green → yellow → red) before switching traffic lights. Additionally, each phase must be held for a **minimum duration** to ensure pedestrians have sufficient time to cross safely. This ensures that the agent adheres to real-world constraints and common safety practices.

## C. Reinforcement Learning Agent

The RL agent interacts with the environment by observing the current state and selecting actions to optimize traffic flow. The key components of the agent are as follows:

- **Observation Space:** The agent observes the current traffic light phase, the time since the last phase change, and lane-specific metrics such as the number of queued vehicles and average wait time.
- **Action Space:** The agent has a discrete action space, where each action corresponds to a specific traffic light phase. Each phase determines which lanes are active (green) and which are inactive (red).
- **Reward Function:** The agent receives positive reinforcement for vehicles passing through the intersection and negative reinforcement for vehicles queued in inactive lanes. This encourages the agent to minimize waiting times and congestion.

## D. Evaluation Metrics

To evaluate the performance of the RL agent, we measure two key metrics across multiple episodes:

- **Mean Wait Time:** The average time vehicles spend waiting at the intersection.
- **Mean Emissions:** The average emissions produced by vehicles during the simulation.

These metrics are calculated for each episode and compared against traditional traffic light systems to demonstrate the effectiveness of our RL-based approach.

## E. Real-World Reflection

Our simulation design reflects real-world traffic conditions in several ways:

- **Random Vehicle Deployments:** By introducing randomness into vehicle arrivals, we ensure that the agent generalizes well to unpredictable traffic patterns.
- **Transition Phases:** Enforcing transition phases ensures that the agent adheres to real-world safety standards.
- **Minimum Phase Durations:** Each phase is held for a minimum duration to ensure pedestrians have sufficient time to cross safely.
- **Dynamic State Updates:** The environment dynamically updates the state using TraCI, allowing the agent to respond to real-time traffic changes.

- **Real-World Networks:** To enhance realism, we use the **OSM Web Wizard** to export actual geographical location networks into SUMO. This allows us to simulate traffic in real-world environments, such as the road network around McMaster University. Figure 2 shows a side-by-side comparison of the Google Maps view of McMaster University and the corresponding SUMO simulation.

## III. Results

In this section, we present the performance of our RL-based traffic light control system compared to a traditional system. We evaluate the system using two key metrics: **mean average wait time** and **mean average emissions**. The results are analyzed across varying traffic densities, represented by the vehicle spawn rate on the x-axis. Each data point is the average of 100 episodes to ensure statistical reliability and smooth distributions.

### *A. Effect of Traffic Density on Mean Average Wait Time*

Figure 3 shows the effect of traffic density on the mean average wait time. The x-axis represents the vehicle spawn rate (ranging from 0 to 1), divided into three regions: **low traffic** (blue), **medium traffic** (green), and **high traffic** (orange). The y-axis represents the mean average wait time in seconds.

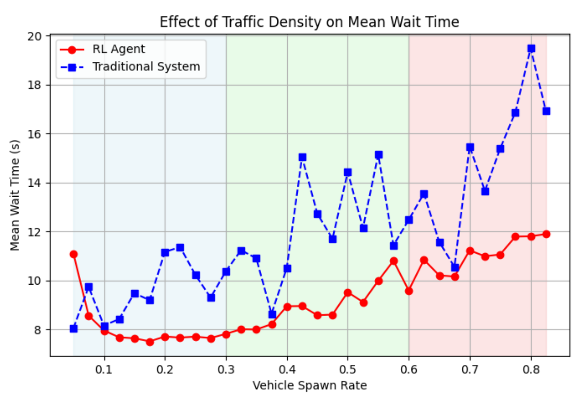


Fig. 3. Effect of traffic density on mean average wait time. The red line represents the RL agent, while the blue dashed line represents the traditional system.

The results demonstrate that the RL agent consistently outperforms the traditional system across all traffic densities. Specifically:

- In low traffic conditions (spawn rate $\leq 0.3$), the RL agent reduces wait times by approximately **13%** compared to the traditional system.
- In medium traffic conditions ($0.3 \leq$ spawn rate $\leq 0.6$), the RL agent achieves approximately a **24%** reduction in wait times.
- In high traffic conditions (spawn rate $\geq 0.6$), the RL agent maintains its advantage, reducing wait times by approximately **19%**.

These improvements highlight the RL agent's ability to adapt to varying traffic conditions, minimizing delays even as traffic density increases.

### *B. Effect of Traffic Density on Mean Average Emissions*

Figure 4 shows the effect of traffic density on the mean average emissions. The x-axis again represents the vehicle spawn rate, while the y-axis represents the mean average emissions in milligrams (mg).

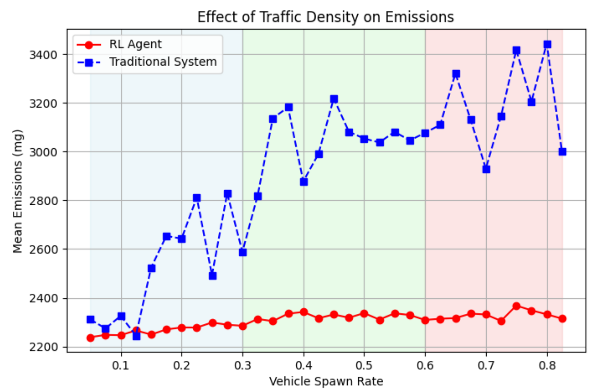


Fig. 4. Effect of traffic density on mean average emissions. The red line represents the RL agent, while the blue dashed line represents the traditional system.

The results indicate that the RL agent significantly reduces emissions compared to the traditional system. Key observations include:

- In low traffic conditions, the RL agent reduces emissions by approximately **10%**.
- In medium traffic conditions, the reduction in emissions is approximately **23%**.
- In high traffic conditions, the RL agent achieves approximately a **29%** reduction in emissions.

These findings underscore the environmental benefits of the RL-based system, particularly in high-density traffic scenarios where idling and stop-and-go driving are most prevalent.

## IV. Conclusion

In this project, we developed an RL-based traffic light control system using SUMO and Stable-Baselines3 to address the inefficiencies of traditional traffic management systems. Our results demonstrate that the RL agent significantly outperforms traditional systems in both reducing mean average wait times and lowering emissions across a wide range of traffic densities. By leveraging real-time data and dynamic decision-making, the RL agent adapts to varying traffic conditions, offering a more sustainable and efficient solution for urban mobility.

### *A. Mean Average Emissions*

As seen by the results above, the emissions are significantly decreased by the use of the RL agent. This is most significantly impacted by the accumulative decrease in number of stops.

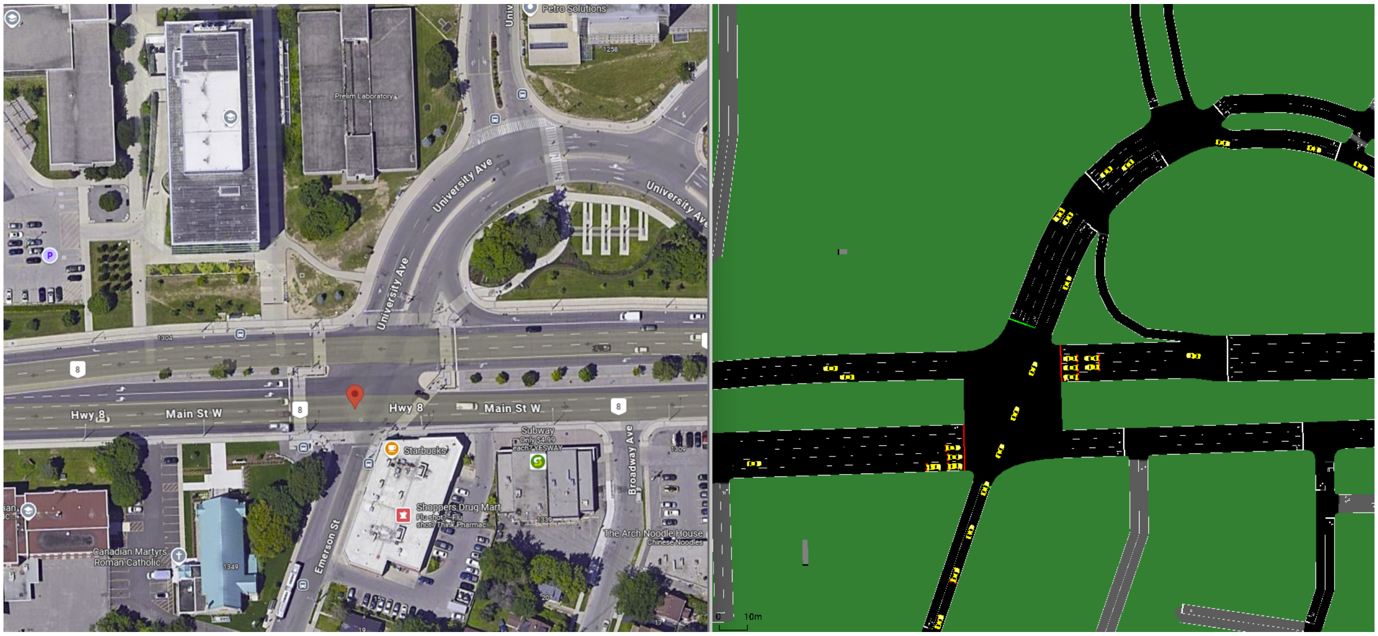


Fig. 2. Side-by-side comparison of the Google Maps view of McMaster University (left) and the corresponding SUMO simulation (right) generated using the OSM Web Wizard.

Particularly in medium to high traffic densities, the RL agent is able to distribute traffic for the most efficient signals, ultimately decreasing the number of total stops required, and overall emissions produced.

### B. Mean Average Wait Time

As seen by the results above, the mean average wait time is decreased most significantly over medium traffic densities. We believe this to be true as a result of the agent training process. The agent was trained on various traffic densities, however since our increments were linear, it encounters medium traffic densities most often, and thus is best at producing optimal results in these cases. In the future, we would fine-tune the training of our agent by implementing additional timesteps at the extreme ends of traffic densities to account for their low quantities in the overall training data.

### C. Next Steps and Challenges

Future steps are to continue to apply our project to various real-world locations, outside of the various Ontario University Campuses that we have currently chosen. Additionally we would like to implement additional considerations to our agent such as pedestrian advances, precedence for emergency vehicles, and other traffic conditions!

## References

[1] G. Zheng, H. Yao, Z. Xu, X. Liang, and Z. Zhang, "A survey of deep reinforcement learning in traffic signal control," ACM Computing Surveys (CSUR), vol. 52, no. 1, pp. 1–35, 2019.

[2] INRIX, "Global Traffic Scorecard," 2022. [Online]. Available: https://inrix.com/scorecard/

[3] P. Mirchandani and L. Head, "REALBAND: An approach for real-time coordinated traffic signal control," Transp. Res. C, vol. 9, no. 6, pp. 415–432, 2001.

[4] P. Hunt, D. Robertson, R. Bretherton, and R. Winton, "SCOOT—a traffic responsive method of coordinating signals," Tech. Rep., Transport and Road Research Laboratory, 1981.

[5] E. Van der Pol and F. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in Proc. NeurIPS, 2016.