ISSN: 2456-8503

International Journal for Development of Science and Technology (IJDST)

Adaptive traffic signal control using reinforcement learning

RESEARCH

Dhanush[1], Ragunath[1], Visweshwaran[1], Senthil Prakash[1]*

Abstract

This paper presents the design, development, and experimental evaluation of a Reinforcement Learning (RL) based Adaptive Traffic Signal Control (ATSC) system for intelligent urban traffic management. The proposed approach formulates traffic signal optimisation as a Markov Decision Process (MDP) and applies Deep Reinforcement Learning (DRL) algorithms specifically Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Advantage Actor-Critic (A2C) to enable real-time adaptive decision-making under dynamic traffic conditions. The system is organized into four integrated modules covering user interaction and visualization, RL algorithm execution, reward-based feedback, and traffic simulation. Experimental outcomes show that RL agents, particularly PPO, consistently surpass conventional fixed-time and actuated control methods across key metrics including average waiting time, queue length, throughput, and estimated emissions. Network-wide efficiency is further improved through multi-agent coordination using a Centralised Training and Decentralised Execution (CTDE) strategy. The framework provides a scalable, modular, and extensible platform for future research and smart city integration.

Keywords: *Reinforcement learning, traffic signal control, markov decision process, deep q-network, ppo, sumo, multi-agent systems, smart city, adaptive control.*

1. Introduction

Traffic congestion in urban areas remains a persistent and growing challenge for transportation systems in rapidly expanding cities. As more people migrate to urban centres and vehicle ownership rises steadily, existing road infrastructure is increasingly unable to meet growing demand. The impact extends well beyond everyday inconvenience.

[1]*Department of Computer Science and Engineering, Shree Venkateswara Hi-Tech Engineering College (Autonomous), Tamilnadu, India.*

[2]*Department of Computer Science and Engineering, Shree Venkateswara Hi-Tech Engineering College (Autonomous), Tamilnadu, India.*

[3]*Department of Computer Science and Engineering, Shree Venkateswara Hi-Tech Engineering College (Autonomous), Tamilnadu, India.*

[4]*Professor, Head of the Department, Department of Computer Science and Engineering, Shree Venkateswara Hi-Tech Engineering College (Autonomous), Tamilnadu, India.*

**Corresponding Author: jtysp14@gmail.com*

It carries significant economic, environmental, and social consequences [1]. From an economic standpoint, congestion places a heavy financial burden on society. Reports estimate the United States incurs annual economic losses of around $74 billion due to traffic delays, translating to roughly $771 per individual driver. Similarly, the UK and Germany face substantial losses of £7.8 billion and €3.6 billion respectively. The environmental damage is equally concerning vehicles stuck in stop-and-go conditions operate far below peak efficiency, using more fuel per kilometre and releasing significantly higher levels of CO_2, particulate matter, and nitrogen oxides. Research indicates that intelligent traffic flow optimisation can cut harmful emissions by 20% or more [2].

At the core of urban traffic management are traffic signal control systems. Intersections are the primary points where vehicle paths conflict, congestion originates, and delays propagate across the entire network. Despite their critical role, many of today's signals still rely on control methods developed decades ago methods poorly suited to handle the complex, dynamic, and unpredictable nature of real-world traffic [3]. This paper introduces a solution that transitions traffic control from traditional rule-based reactive mechanisms toward intelligent, learning-driven predictive optimization. The proposed system leverages DRL algorithms trained within the SUMO microscopic traffic simulator to discover optimal signal timing strategies across a wide range of traffic conditions.

2. Background and related work

2.1. Conventional Traffic Signal Control

Fixed-Time Control operates on predetermined signal cycles with fixed phase sequences. While straightforward to implement, it is entirely incapable of adjusting to real-time variations in traffic demand. A timing plan optimized for morning rush hours may degrade significantly during off-peak periods or unexpected incidents [4]. Actuated Control introduced vehicle detection through inductive loop sensors, enabling some responsiveness to immediate demand. However, it remains fundamentally reactive and localised it can only extend green phases after vehicles have arrived and started waiting, and lacks the capacity to optimise network-wide coordination or learn from historical patterns. Coordinated Control attempts to link multiple intersections along a corridor to create green wave progressions. While effective under steady conditions, this approach is vulnerable to disruptions any irregularity upstream breaks synchronisation and cascades delays downstream. Centralised Adaptive Systems such as SCOOT, SCATS, and RHODES represent the current peak of conventional technology. They incorporate real-time sensor data into optimisation models but rely on simplified mathematical traffic representations and, critically, cannot learn from accumulated experience [5].

2.2. Reinforcement Learning for Traffic Control

Reinforcement Learning offers a substantially different paradigm for traffic signal control. Instead of depending on manually designed rules or simplified models, an RL agent develops an effective decision-making policy through continuous trial-and-error interaction with the environment. Early studies showed Q-learning agents could outperform fixed-time control at single intersections. More recent Deep RL advances including DQN, PPO, and actor-critic architectures have produced notable improvements in multi-intersection and network-scale settings [6]. The SUMO simulation platform has become the widely accepted research environment for RL-based traffic control, offering microscopic vehicle dynamics, flexible demand configuration, and a comprehensive TraCI API for external agent interaction. The SUMO-RL library further streamlines integration by providing a Gymnasium-compatible interface for rapid algorithm prototyping and comparison.

3. Proposed system architecture

The proposed RL-ATSC framework is built upon an MDP formulation of the traffic signal control problem and comprises four tightly integrated modules. The architecture is summarised in (Table 1).

Table 1: Key modules of the RL-ATSC architecture

Module	Primary Function	Key Technology	Output
Front-end	Real-time	Dash	Interactive
Module	Visualisation and operator control	Plotly/Websocket	Dashboard
RL	Policy	DQN, PPO	Signal
Algorithm Module	Learning and Action Selection	A2C (PyTorch)	Phase Decisions
Reward System	Multi-Objective	Reward	Reward
Module	Feedback to Agent	Functions	Signal

3.1. MDP Formulation

The traffic signal control problem is formally represented as a Markov Decision Process. The State space S captures queue lengths at each lane approach (vehicles with near-zero speed), vehicle density and occupancy across lane segments, cumulative waiting time per lane, the current active signal phase and elapsed duration, and optionally data from adjacent intersections. The Action space A supports two configurations discrete phase selection, where the agent picks the next phase for a fixed minimum duration, and phase extension, where every five seconds the agent decides whether to prolong the active green phase or switch. The Reward function R encodes project objectives as negative quantities total vehicle waiting time, cumulative queue length, estimated fuel consumption and emissions, and time-to-collision conflict metrics. Weighted combinations support multi-objective optimisation, allowing practitioners to balance mobility, environmental sustainability, and safety goals.

3.2. Deep RL Algorithms

Three DRL algorithms are implemented and evaluated. DQN uses a neural network to approximate Q-values for every state-action pair, with experience replay and periodic target network updates to maintain stable learning. PPO applies a clipped surrogate objective with advantage estimation, delivering stable policy gradient updates well-suited to the non-stationary traffic environment. A2C combines a shared neural backbone for the policy (actor) and value function (critic), supporting efficient on-policy learning with reduced variance compared to standard policy gradient methods.

3.3. Multi-Agent Extension

For network-level control, the framework supports Multi-Agent Reinforcement Learning (MARL) with three coordination strategies Independent Learners, where each intersection trains its own policy without communication, CTDE, which uses a centralised critic during training to support coordination while keeping local execution scalable, and Parameter Sharing, where all agents use a single shared policy network with agent-specific inputs, reducing parameter count and improving sample efficiency.

4. Implementation methodology

4.1. Development Environment

The system is developed on Ubuntu 22.04 LTS using AMD Ryzen 3 7320U processors with 8 GB RAM. SUMO version 1.18.0 serves as the traffic simulation engine and Python 3.9 is the primary programming language. PyTorch provides the deep learning infrastructure, while the SUMO-RL library offers a Gymnasium-compatible wrapper for RL environment integration. NumPy, Pandas, and Matplotlib support data handling and result visualisation.

4.2. Front-End Module

The front-end is a dual-interface system. A web-based dashboard built with the Dash framework delivers real-time visual updates through WebSocket-based data streaming from the simulation backend. Key components include traffic network and signal state visualisation using Plotly graphics with colour-coded congestion indicators, control panels for algorithm selection and demand configuration, live time-series graphs of queue lengths and waiting times, and heatmap displays for bottleneck identification. A lightweight desktop interface is also available for local deployment with reduced latency.

Table 2: System configuration parameters

Parameter	Configuration
State Dimension	Queue lengths, densities, waiting times, phase, phase duration
Action Space	Discrete Phase Selection (4-8 Phases per Intersection)
Decision Interval	5 Seconds
Minimum Green Time	10 Seconds
Maximum Green Time	60 Seconds
Yellow Phase Duration	3 Seconds
Reward Function	Negative Total Waiting Time (Primary); Multi-objective Variant
Neural Network	3 Hidden Layers, 256 Units, ReLU Activation
Replay Buffer Size	50,000 Transitions
Batch Size	64
Discount Factor (γ)	0.99

- *State Dimension-Queue Lengths, Densities, Waiting Times, Phase, Phase Duration:* The state captures prevailing traffic conditions at the intersection. Queue lengths indicate vehicles waiting at each lane, densities reflect occupancy per road segment; waiting times record cumulative stop durations; phase indicates the active signal cycle, and phase duration tracks how long the current phase has been running. This rich representation lets the agent make well-informed decisions based on real-time conditions and timing constraints.

- *Action Space-Discrete phase selection (4-8 phases per intersection):* At every decision step, the agent selects the next signal phase to activate. The number of phases depends on intersection geometry-simple crossings may use 4, while complex layouts with dedicated turn signals may require up to 8. Only one phase can be active at a time, making this a discrete action problem.

- *Decision Interval-5 seconds:* The agent makes a phase selection every 5 seconds unless constrained by minimum or maximum green time limits. Five seconds offers a practical balance between adaptability and stability.

- *Minimum Green Time-10 seconds:* Once activated, a phase must remain green for at least 10 seconds before the agent can switch. This prevents rapid phase changes, ensures adequate pedestrian crossing time, and meets standard driver expectation requirements.

- *Maximum Green Time-60 seconds:* A phase cannot stay active for more than 60 seconds, preventing other movements from being starved of green time and bounding vehicle waiting durations.

- *Yellow Phase Duration-3 seconds:* A fixed yellow interval of 3 seconds is inserted between phase transitions, warning drivers the current phase is ending. No RL decision is made during this interval.

- *Reward Function-Negative total waiting time (primary), multi-objective variant:* The agent minimises total vehicle waiting time Reward = − total waiting time in the last decision interval. A multi-objective variant may additionally incorporate queue length, stop count, fuel consumption, and fairness metrics.

- *Neural Network-3 hidden layers, 256 units, ReLU activation:* The agent uses a deep neural network with three 256-neuron hidden layers and ReLU activation. This architecture provides sufficient capacity to model complex traffic dynamics without excessive computational cost.

- *Replay Buffer Size-50,000 transitions:* The agent stores 50,000 past experiences (~ 69 simulated hours at 5-second intervals), large enough to reduce temporal correlations while remaining manageable in memory.

- *Batch Size-64:* During each training update, 64 transitions are randomly sampled, balancing computational efficiency with gradient stability.

- *Discount Factor (γ) – 0.99:* A discount factor of 0.99 places nearly equal weight on future and immediate rewards, encouraging long-term planning over extended traffic management horizons (Table 2).

4.3. RL Algorithm Module

Each algorithm is implemented as a dedicated agent class derived from a shared base class, ensuring a consistent. Interface for training, action selection, and model checkpointing. For DQN, a deque-based replay buffer and periodic target network updates stabilise convergence. The PPO implementation uses a clipped surrogate loss with separate policy and value networks to prevent destabilising updates. For multi-agent scenarios, the CTDE architecture maintains a centralised value function during training while agents execute locally using their own observations. Multi-objective rewards use a weighted sum with weights adjustable per deployment context.

5. Results and discussion

Experiments were conducted across three network configurations a single standalone intersection, 4-intersection linear corridor, and a 3×3 grid network. Each was tested under three demand patterns constant flow, simulated rush-hour peaks, and stochastic poisson-distributed arrivals. All RL agents were trained over 500 episodes and evaluated across 50 independent test episodes. Baseline comparators were fixed-time control with Webster-optimal cycle lengths and standard actuated control with gap-acceptance logic.

6. Challenges and future scope

Despite the encouraging results, several limitations must be acknowledged. The system was developed and evaluated entirely in simulation; real-world deployment would introduce challenges including sensor noise, communication delays, hardware-in-the-loop constraints, and unpredictable driver responses to changed signal behaviour. The SUMO environment, though high-fidelity, still simplifies pedestrian interactions, bicycle traffic, and nuanced driver behaviour models training for large multi-intersection grid scenarios remains computationally intensive, requiring significant resources and careful hyper parameter tuning [7]. The current implementation focuses primarily on private vehicle traffic with limited attention to pedestrians, cyclists, and public transit. Adapting reward functions to

minimise person-level delay rather than vehicle-level delay would better align the system with sustainable mobility goals. The sim-to-real gap represents a key challenge before operational deployment [8]. Future enhancements include integration with Connected and Autonomous Vehicles (CAVs) via Vehicle-to-Infrastructure (V2I) communication, multi-modal extensions covering pedestrian crossings and transit priority, advanced MARL architectures such as hierarchical RL with graph neural networks for spatial modelling, and explainability tools for operator trust and regulatory compliance [9].Transfer and meta-learning approaches will reduce training requirements for new deployment sites [10].

7. Conclusion

This paper presented a comprehensive framework demonstrating meaningful improvements over conventional traffic management approaches. The system models the signal control problem as an MDP and trains DQN, PPO, and A2C agents within the SUMO simulator. The modular four-component architecture integrating visualization, RL decision-making, multi-objective reward design, and high-fidelity simulation provides a robust and extensible research platform. Testing across single-intersection, corridor, and grid scenarios confirms that RL agents-particularly PPO-consistently outperform fixed-time and actuated baselines on all evaluated metrics. Multi-objective reward design successfully balances mobility, environmental sustainability, and safety. CTDE-based multi-agent coordination achieves network-level gains while maintaining deployment scalability. These findings contribute a validated open framework for RL-based traffic control research, empirical benchmarks across multiple DRL algorithms, and evidence that multi-objective and multi-agent approaches can simultaneously improve urban mobility, reduce emissions, and enhance safety. Future work will target real-world pilot deployments, multi-modal integration, CAV cooperation, and explainable AI for operational smart city infrastructure.

Conflict of interest statement: The authors confirm that there are no competing interests related to the publication of this research paper.

Funding information: This research was carried out without any dedicated funding from public, commercial, or not-for-profit organisations.

Data availability statement: Simulation data were produced using the open-source SUMO platform. Code and configuration files are available from the corresponding author upon reasonable request.

Ethical approval statement: This study is entirely simulation-based and does not involve human subjects, animals, or personal data. Ethical approval was not required.

Acknowledgement: The authors thank the faculty and staff of the Department of Computer Science and Engineering at Shree Venkateshwara Hi-Tech Engineering College for their guidance and support throughout this project.

.

References

1. Sutton RS, Barto AG, Reinforcement learning: An introduction, 2nd ed, MIT Press. 2018
2. Texas AM, Transportation institute, Urban Mobility Report, College Station, TX. 2023
3. Webster FV, Traffic signal settings, Road Research Technical Paper No. 39, HMSO, London. 1958
4. Papageorgiou M, Review of road traffic control strategies proceedings of the IEEE. 2003, 91(12):2043-2067
5. Lowrie PR, SCATS: Sydney Coordinated Adaptive Traffic System, Roads and Traffic Authority, NSW. 1990
6. Wei H, IntelliLight: A reinforcement learning approach for intelligent traffic light control. Proc. 24th ACM SIGKDD. 2018, 2496-2505

7. Chu T, Multi-agent deep reinforcement learning for large-scale traffic signal control, IEEE Trans., Intelligent Transportation Systems. 2019, 21(3):1086-1095
8. Chen C, Toward a thousand lights: Decentralized deep RL for large-scale traffic signal control. Proc. AAAI. 2020, 34(04):3414-3421
9. Genders W, Razavi S, Using a deep reinforcement learning agent for traffic signal control. 2016, arXiv:1611.01142
10. Zeng J, Hu J, Zhang Y, City light: A universal model towards real-world city-scale traffic signal control coordination. 2024, arXiv:2406.02126