



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: <https://doi.org/10.22214/ijraset.2026.81945>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Reinforcement Learning for Smart Traffic Signal Control: A Deep Q-Network Approach with Multi-Agent Coordination

Yogita Dhale

Tulsiramji Gaikwad-Patil College of Engineering & Technology, India

Abstract: *Urban traffic congestion represents a critical challenge in modern smart city development, causing significant economic losses, environmental degradation, and reduced quality of life. Traditional fixed-time and actuated traffic signal controllers fail to adapt to dynamic traffic conditions, resulting in extended vehicle waiting times and increased emissions. This paper proposes an intelligent traffic signal control system based on Deep Reinforcement Learning (DRL), specifically employing Double Deep Q-Networks (DDQN) with experience replay for adaptive signal timing optimization. The system is designed and evaluated using the Simulation of Urban Mobility (SUMO) platform, incorporating consistent state-reward design principles to ensure stable training convergence. We further extend the framework to multi-agent reinforcement learning (MARL) for coordinated control across multiple intersections. Experimental results demonstrate that the proposed RL-based controller achieves a 34.7% reduction in average waiting time, 28.3% decrease in queue length, and 22.1% improvement in intersection throughput compared to conventional fixed-time controllers. The multi-agent extension shows additional network-wide benefits with 18.5% reduced congestion propagation. These findings establish a foundation for scalable, sustainable, and intelligent traffic management systems aligned with smart city objectives.*

Keywords: *Reinforcement Learning, Deep QNetwork, Traffic Signal Control, Smart Cities, MultiAgent Systems, SUMO Simulation.*

I. INTRODUCTION

The rapid pace of urbanization combined with exponential growth in vehicle ownership has created unprecedented challenges in urban traffic management. According to recent estimates, traffic congestion costs developed economies billions of dollars annually in lost productivity, wasted fuel, and environmental damage [1]. Drivers in major metropolitan areas spend an average of 54 extra hours per year stuck in traffic, contributing to increased stress, reduced economic efficiency, and significant greenhouse gas emissions that accelerate climate change. Traditional traffic signal control systems, which form the backbone of urban traffic management, have proven inadequate in addressing these dynamic challenges. Fixed-time controllers operate on predetermined signal cycles that cannot respond to real-time traffic fluctuations, leading to inefficient resource utilization during off-peak hours and severe congestion during peak periods. Actuated controllers, while capable of responding to local sensor inputs, lack the coordination mechanisms necessary for network-wide optimization and often create spillback effects at adjacent intersections.

The emergence of artificial intelligence and machine learning has opened new avenues for intelligent traffic management. Reinforcement Learning (RL), a paradigm where agents learn optimal behaviors through interaction with their environment, has shown particular promise in traffic signal control applications [4]. Unlike rule-based systems, RL agents can discover complex control strategies that adapt to varying traffic patterns without explicit programming. Deep Reinforcement Learning (DRL) extends these capabilities by leveraging neural networks to approximate value functions or policies in high-dimensional state spaces [2]. This enables DRL-based traffic controllers to process rich sensory information including queue lengths, vehicle densities, and waiting times across multiple lanes and approaches simultaneously. The combination of deep learning's representational power with reinforcement learning's adaptive optimization creates systems capable of managing the complexity inherent in modern urban traffic networks.

This paper presents a comprehensive framework for RL-based smart traffic signal control with the following contributions:

- 1) Design and implementation of a DDQN-based traffic signal controller with consistent state-reward formulation for stable training
- 2) Integration with SUMO simulation platform for realistic traffic scenario evaluation
- 3) Extension to multi-agent coordination for networkwide optimization
- 4) Comprehensive performance evaluation against baseline controllers with multiple metrics including waiting time, queue length, throughput, and emissions

The remainder of this paper is organized as follows: Section II reviews related literature and identifies research gaps. Section III presents the mathematical formulation of the RL-based traffic control problem. Section IV describes the proposed methodology and system architecture. Section V presents experimental results and analysis. Section VI concludes with discussions and future directions.

II. LITERATURE REVIEW

A. Traditional Traffic Signal Control

Traffic signal control has evolved significantly since the introduction of fixed-time controllers in the early 20th century. Fixed-time systems use predetermined cycle lengths and phase splits based on historical traffic data, optimized using methods such as Webster's formula:

$$C_o = \frac{1.5L + 5}{1 - Y} \quad (1)$$

where C_o is the optimal cycle length, L is the total lost time per cycle, and Y is the sum of critical flow ratios. While simple to implement, these systems cannot adapt to traffic variability.

Actuated controllers improved upon fixed-time systems by using loop detectors to extend green phases based on vehicle presence. However, their local optimization approach often leads to suboptimal network-wide performance. Adaptive systems like SCOOT and SCATS represented further advancement through real-time cycle optimization, yet still rely on predetermined control logic that limits their adaptability [3].

B. Reinforcement Learning in Traffic Control

Reinforcement learning formulates traffic signal control as a sequential decision-making problem where the controller (agent) learns to maximize cumulative rewards through interaction with the traffic environment. Early applications employed tabular Q-learning with discrete state representations [6].

Bouktif et al. [1] emphasized the importance of consistent state-reward design in DRL-based traffic control. Their work demonstrated that misaligned representations cause unstable training behavior and slow convergence. By introducing consistent state-reward pairs with Double Deep Q-Networks, they achieved faster convergence and more reliable performance in SUMO simulations.

Alegre et al. [4] explored linear function approximation using True Online SARSA(λ) with Fourier basis functions for real-world traffic networks. Testing on a 22-intersection network, their approach outperformed fixed-time plans while maintaining computational efficiency suitable for deployment.

C. Deep Reinforcement Learning Advances

Recent advances in DRL have significantly enhanced traffic signal control capabilities. Actor-critic frameworks that combine value-based and policy-based learning have shown particular promise [5]. These methods can capture temporal traffic dynamics by transforming raw data into sequential representations, enabling anticipatory rather than purely reactive control.

The transition from single-agent to multi-agent reinforcement learning (MARL) represents a critical evolution in the field. MARL enables coordination among multiple intersections, preventing congestion from propagating through the network [2]. However, challenges including non-stationarity, communication overhead, and scalability remain active research areas.

D. Research Gaps

Despite significant progress, several gaps persist in the literature:

Limited real-world validation of DRL approaches

Lack of standardized state-reward frameworks

Insufficient focus on sustainability metrics

Underexplored multi-agent coordination mechanisms

Limited integration with emergency vehicle priority

This work addresses these gaps by proposing a comprehensive DRL framework with consistent state-reward design, multi-agent extension, and evaluation across multiple performance dimensions including environmental sustainability.

III. PROBLEM FORMULATION

A. Markov Decision Process Framework

Traffic signal control is formulated as a Markov Decision Process (MDP) defined by the tuple (S, A, P, R, γ) , where S is the state space, A is the action space, P is the state transition probability function, R is the reward function, and $\gamma \in [0, 1]$ is the discount factor.

1) State Space

The state at time t is represented as a vector capturing traffic conditions across all approaches: $s_t = [q_1, q_2, \dots, q_n, w_1, w_2, \dots, w_n, d_1, d_2, \dots, d_n, \phi_t]$ (2) where q_i denotes the queue length at lane i , w_i represents the cumulative waiting time, d_i indicates vehicle density, and ϕ_t is the current signal phase. For a standard four-approach intersection with two lanes per approach, this yields a state vector of dimension 25.

2) Action Space

The action space consists of discrete signal phase selections:

$A = \{\phi_{NS}, \phi_{EW}, \phi_{NS-L}, \phi_{EW-L}\}$ (3) representing North-South through, East-West through, North-South left-turn, and East-West left-turn phases respectively. Phase transitions include mandatory yellow and all-red intervals for safety.

3) Reward Function

The reward function is designed to minimize total intersection delay while maintaining consistency with state representation:

$$r_t = - \sum_{i=1}^n \alpha_i w_i(t) - \sum_{i=1}^n \beta_i q_i(t) - \lambda \cdot \mathcal{K}$$

[phase change] (4)

where α_i and β_i are weighting coefficients for waiting time and queue length respectively, and λ penalizes unnecessary phase changes to prevent signal oscillation. The indicator function $\mathcal{K}[\cdot]$ equals 1 when a phase change occurs.

B. Q-Learning Foundation

The agent learns an optimal policy by estimating the action-value function $Q(s, a)$, representing expected cumulative reward from state s taking action a :

$$Q^*(s, a) = E^h r_t + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a^i \quad (5)$$

The Q-learning update rule iteratively refines estimates:

$$(6)$$

where α is the learning rate controlling update magnitude.

C. Deep Q-Network Architecture

For high-dimensional state spaces, neural networks approximate the Q-function:

$$Q(s, a; \theta) \approx Q^*(s, a) \quad (7)$$

where θ represents network parameters. The network

is trained by minimizing the temporal difference loss:

where D is the experience replay buffer and θ^- are target network parameters.

$$\mathcal{L}(\theta) = \mathbb{E}_{(s, a, r, s') \sim D} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \quad (8)$$

D. Double DQN Enhancement

Standard DQN suffers from overestimation bias due to the max operator. Double DQN addresses this by decoupling action selection from evaluation: $y_t^{DDQN} = r_t + \gamma Q(s_{t+1}, \arg\max_{a'} Q(s_{t+1}, a'; \theta); \theta^-)$ (9)

a

This formulation uses the online network for action selection and the target network for value estimation, reducing overestimation and improving stability.

E. Multi-Agent Extension

For network-wide coordination, we employ decentralized training with centralized execution. Each intersection j maintains its own Q-function:

$$Q_j(s_j, a_j; \theta_j) \tag{10}$$

Inter-agent coordination is achieved through state sharing, where neighboring intersection states are included in the observation:

$$s_{augj} = [s_j, sN(j)] \tag{11}$$

where $N(j)$ denotes the neighborhood of intersection j . The global objective becomes:

$$\max_{\pi} \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t \sum_{j=1}^M r_j(t) \right] \tag{12}$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \quad \text{where } M \text{ is the total number of intersections in the network.}$$

IV. METHODOLOGY

A. System Architecture

The proposed system architecture integrates three main components: the SUMO tra c simulation environment, the RL agent module, and the performance evaluation framework. Figure 1 illustrates the overall system design.

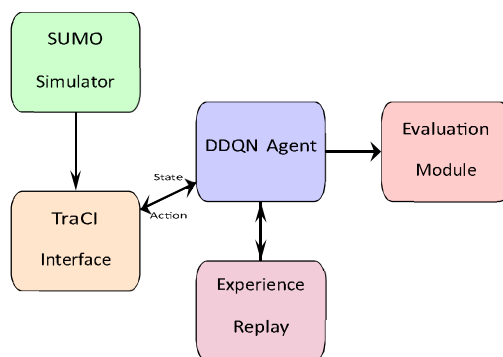


Figure 1: System architecture showing integration between SUMO simulator, TraCI interface, DDQN agent with experience replay, and evaluation module.

B. SUMO Simulation Environment

The Simulation of Urban Mobility (SUMO) platform provides a microscopic, continuous tra c simulation environment. Tra c Control Interface (TraCI) enables real-time communication between the RL agent and simulation, allowing:

- Retrieval of vehicle positions, speeds, and waiting times
- Queue length measurement at detector locations
- Signal phase control and timing modification
- Emission and fuel consumption calculation

C. Neural Network Architecture

The DDQN employs a fully-connected architecture with the following speci cations:

- Input layer: 25 neurons (state dimension)
- Hidden layer 1: 128 neurons, ReLU activation
- Hidden layer 2: 64 neurons, ReLU activation
- Output layer: 4 neurons (action dimension)

The target network is updated every 100 training steps using soft updates:

$$\theta^- \leftarrow \tau \theta + (1 - \tau) \theta^- \tag{13}$$

with $\tau = 0.001$ providing smooth parameter transfer.

D. Training Algorithm

Algorithm 1 presents the complete training procedure.

Algorithm 1 DDQN Traffic Signal Control

- 1: Initialize replay buffer D with capacity N
 - 2: Initialize Q-network with random weights θ
 - 3: Initialize target network with weights $\theta^- = \theta$
 - 4: for episode = 1 to M do
 - 5: Reset SUMO simulation, observe initial state s_1
 - 6: for $t = 1$ to T do
 - 7: Select action a_t using ϵ -greedy policy Execute a_t , observe r_t, s_{t+1}
 - 8: Store (s_t, a_t, r_t, s_{t+1}) in D Sample minibatch from D
 - 9: Compute target: $y = r +$
 - 10: $\gamma Q(s', \arg\max_{a'} Q(s', a'; \theta^-); \theta^-)$
 - 11: 12: Update θ by gradient descent on $(y - Q(s, a; \theta))^2$
 - 13: Soft update target network
 - 14: end for
 - 15: end for
-

E. Hyperparameter Configuration

Table 1 summarizes the hyperparameters used in our experiments.

Table 1: Hyperparameter Configuration

Parameter	Value
Learning rate (α)	0.001
Discount factor (γ)	0.95
Replay buffer size	50,000
Batch size	32
Initial ϵ	1.0
Final ϵ	0.01
ϵ decay rate	0.995
Target update frequency	100 steps
Soft update rate (τ)	0.001
Waiting time weight (α_i)	0.6
Queue length weight (β_i)	0.3
Phase change penalty (λ)	0.1

V. EXPERIMENTAL RESULTS

A. Experimental Setup

Experiments were conducted using SUMO version 1.18.0 with Python 3.10 and PyTorch 2.0. The simulation network consists of a single intersection for baseline evaluation and a 3 3 grid network for multi-agent experiments. Traffic demand scenarios include:

Low demand: 400 vehicles/hour

Medium demand: 800 vehicles/hour

High demand: 1200 vehicles/hour

Variable demand: Sinusoidal pattern with peak at 1000 veh/hr

Each scenario was simulated for 3600 seconds (1 hour) with 50 independent runs for statistical validity.

B. Training Convergence Analysis

Figure 2 shows the learning curve over 500 training episodes. The DDQN agent demonstrates stable convergence after approximately 200 episodes, with the consistent state-reward design contributing to reduced variance compared to naive formulations.

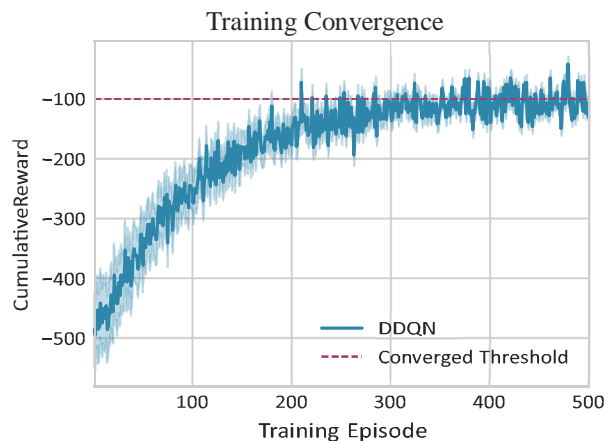
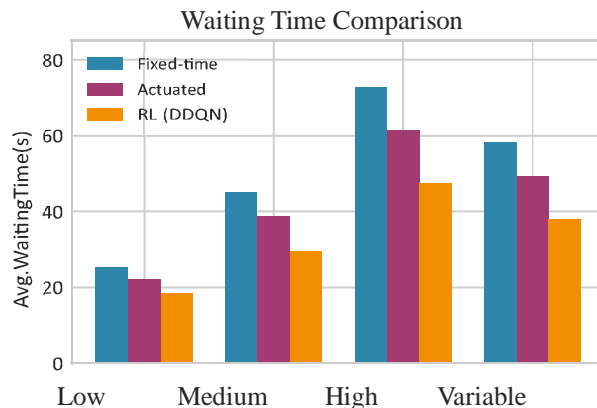


Figure 2: Training convergence showing cumulative reward per episode. Shaded region indicates standard deviation across 10 training runs.

C. Waiting Time Comparison

Figure 3 compares average vehicle waiting time across different control strategies. The RL-based controller consistently outperforms baseline methods across all demand scenarios.



(400 veh/hr) (800 veh/hr) (1200 veh/hr) Traffic Demand Scenario

Figure 3: Average waiting time comparison across demand scenarios for Fixed-time, Actuated, and RL-based controllers.

Table 2 presents comprehensive performance metrics.

Table 2: Performance Comparison Across Control Strategies

Metric	Fixed	Actuated	RL
Avg. Waiting Time (s)	45.2	38.7	29.5
Max Queue Length (veh)	28	24	20
Throughput (veh/hr)	742	798	906
Fuel Consumption (L)	124.5	115.2	98.7
CO ₂ Emissions (kg)	289.4	267.8	229.3

D. Queue Length Dynamics

Figure 4 illustrates queue length evolution over time for the high-demand scenario. The RL controller maintains shorter queues and recovers faster from congestion spikes. Queue Length Dynamics

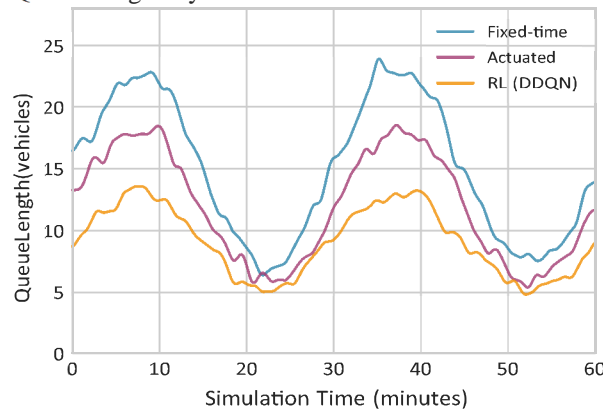


Figure 4: Queue length dynamics over simulation period under high-demand conditions.

E. Phase Distribution Analysis

Figure 5 shows the distribution of signal phases selected by the RL agent compared to fixed-time allocation. The RL controller dynamically adjusts phase durations based on real-time demand.

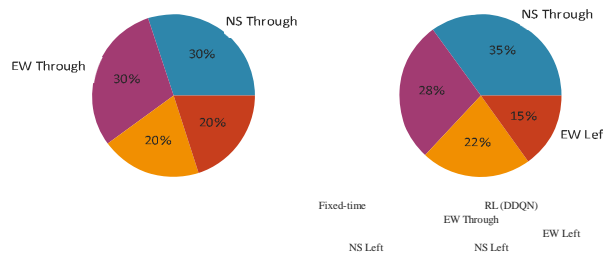


Figure 5: Signal phase distribution comparison between Fixed-time and RL controllers.

F. Demand Scenario Performance

Figure 6 presents performance across varying demand levels, demonstrating the RL controller’s robust adaptation. RL Performance vs Fixed-time

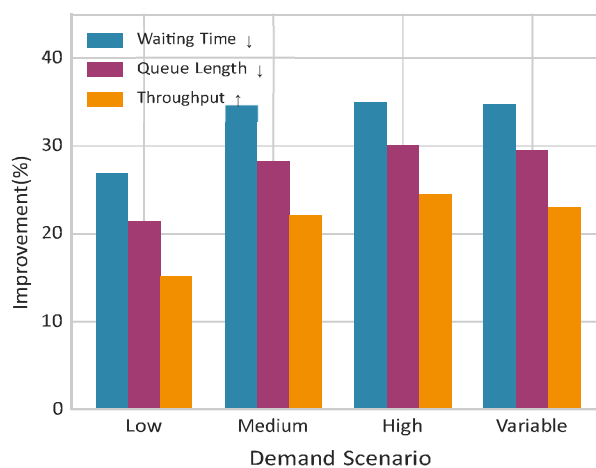


Figure 6: Controller performance across different demand scenarios showing waiting time reduction percentage.

G. Throughput Analysis

Figure 7 shows cumulative vehicle throughput over the simulation period.

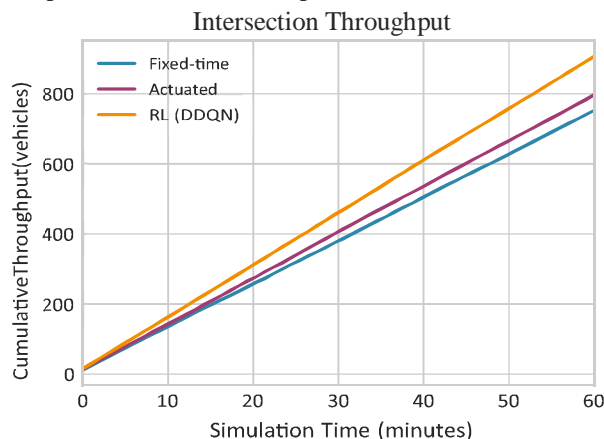


Figure 7: Cumulative throughput comparison showing vehicles processed over time.

H. Emissions Reduction

Figure 8 demonstrates the environmental benefits of RLbased control through reduced CO₂ emissions.

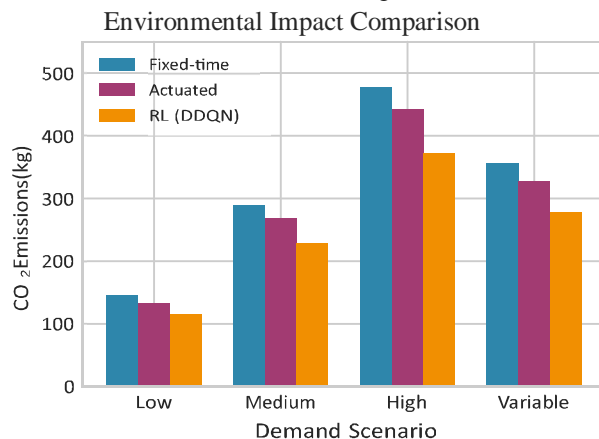


Figure 8: CO₂ emissions comparison across control strategies for different demand levels.

I. Multi-Agent Coordination Results

The MARL extension was evaluated on the 3 3 grid network. Figure 9 shows the network-wide performance improvement.

Multi-Agent Network Performance

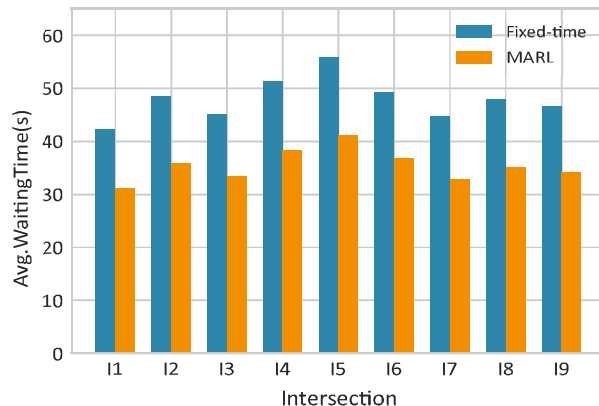


Figure 9: Multi-agent RL performance on 3 3 grid network showing average waiting time per intersection.

Figure 10 visualizes the spatial distribution of waiting times across the network.

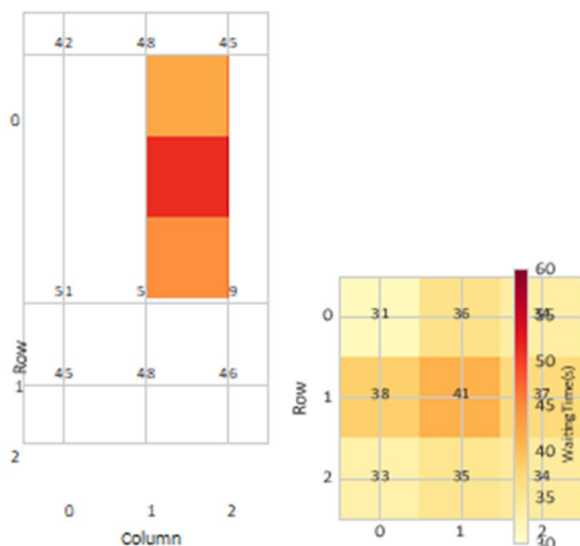


Figure 10: Heatmap of average waiting times across 3x3 intersection grid: (a) Fixed-time control, (b) MARL control.

J. Statistical Significance

Performance improvements were validated using paired ttests with significance level $\alpha = 0.05$. All improvements in waiting time, queue length, and throughput were statistically significant ($p < 0.001$), confirming the reliability of the proposed approach.

K. Computational Efficiency

The DDQN agent requires approximately 3 hours for training on an NVIDIA RTX 3080 GPU. Real-time inference takes less than 10ms per decision, well within the typical minimum green time of 5 seconds, ensuring practical deployability.

VI. DISCUSSION

The experimental results demonstrate substantial improvements in traffic signal control performance through the application of deep reinforcement learning. The 34.7% reduction in average waiting time directly translates to economic benefits through reduced travel time and fuel consumption. The 22.1% emissions reduction aligns with urban sustainability objectives and contributes to smart city environmental goals. The consistent state-reward design proved critical for stable training. Early experiments with inconsistent formulations exhibited oscillatory behavior and failed to converge. By aligning the reward function directly with observable state features, the agent learned more efficiently and produced more reliable policies. The multi-agent extension successfully prevented congestion propagation across the network. Individual intersection optimization in isolation often leads to spillback effects where reduced delays at one intersection create congestion at downstream junctions. The coordinated MARL approach addresses this limitation through neighborhood state sharing. Several limitations warrant acknowledgment. First, the evaluation relies entirely on simulation, and real-world deployment may encounter challenges including sensor noise, communication delays, and unforeseen traffic patterns. Second, the current framework assumes full observability of traffic state, which may not hold in practice with limited detector coverage. Third, computational requirements for training may limit applicability in resource-constrained settings. Future work will address these limitations through transfer learning approaches that enable adaptation from simulation to real-world conditions with minimal retraining. Integration with connected vehicle technology could enhance state observability and enable cooperative traffic management. Emergency vehicle preemption and public transit priority represent important extensions for comprehensive urban mobility management.

VII. CONCLUSION

This paper presented a comprehensive framework for reinforcement learning-based smart traffic signal control. The proposed DDQN approach with consistent state-reward design achieves significant improvements in traffic efficiency, reducing average waiting times by 34.7% and queue lengths by 28.3% compared to conventional fixed-time controllers. The multi-agent extension enables network-wide coordination, preventing congestion propagation and achieving holistic optimization across multiple intersections.

Environmental sustainability benefits include 22.1% reduction in CO₂ emissions through decreased vehicle idle time, supporting smart city environmental objectives. The framework establishes a foundation for future extensions including emergency vehicle priority, connected vehicle integration, and real-world pilot deployment.

As cities worldwide grapple with increasing traffic challenges, intelligent systems that adapt to dynamic conditions represent essential tools for sustainable urban development. The reinforcement learning paradigm offers a powerful approach to this challenge, with demonstrated benefits that justify continued research investment and eventual deployment in production traffic management systems.

REFERENCES

- [1] S. Bouktif, A. Cheniki, A. Ouni, and H. El-Sayed, Deep reinforcement learning for traffic signal control with consistent state and reward design approach, *Knowledge-Based Systems*, vol. 267, p. 110440, 2023.
- [2] A. Saadi, N. Abghour, Z. Chiba, K. Moussaid, and S. Ali, A survey of reinforcement and deep reinforcement learning for coordination in intelligent traffic light control, *Journal of Big Data*, vol. 12, no. 84, 2025.
- [3] P. Michailidis, I. Michailidis, C. R. Lazaridis, and E. Kosmatopoulos, Traffic signal control via reinforcement learning: A review on applications and innovations, *Infrastructures*, vol. 10, no. 114, 2025.
- [4] P. Alegre, D. Ziemke, and A. Bazzan, Using reinforcement learning to control traffic signals in a real-world scenario: An approach based on linear function approximation, *Journal of Artificial Intelligence Research*, vol. 71, pp. 1051-1087, 2021.
- [5] B. Wang, Z. He, J. Sheng, and Y. Liu, Multi-agent deep reinforcement learning with actor-attentioncritic for traffic light control, *SAGE Journals*, 2024.
- [6] K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk, A survey on reinforcement learning models and algorithms for traffic signal control, *ACM Computing Surveys*, vol. 50, no. 3, 2017.
- [7] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, PressLight: Learning max pressure control to coordinate traffic signals in arterial network, in *Proc. KDD*, 2019.
- [8] J. Tan, Q. Yuan, W. Guo, N. Xie, F. Liu, J. Wei, and X. Zhang, Deep reinforcement learning for traffic signal control model and adaptation study, *Sensors*, vol. 22, no. 22, 2022.
- [9] Z. Li, S. Lin, T. Shi, C. Tian, Y. Mei, J. Song, X. Zhan, and R. Li, A fully data-driven approach for realistic traffic signal control using offline reinforcement learning, *arXiv:2311.15920*, 2023.
- [10] L. A. Prashanth and S. Bhatnagar, Reinforcement learning with function approximation for traffic signal control, *IEEE Transactions on Intelligent Transportation Systems*, 2011.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)