Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 6 | June-2026
ISSN (O) 3107-6696

ADAPTIVE MULTI-OBJECTIVE REWARD SHAPING FOR DEEP REINFORCEMENT LEARNING-BASED TRAFFIC SIGNAL CONTROL: THE AMRS-DUELINGDDQN FRAMEWORK

Rishu Raj[1], Sagar Choudhary[2], Rohit Kumar[3]

[1,3]Students, Department of Computer Science, Quantum University Roorkee India.

[2]Associate Professor, Department of Computer Science, Quantum School Of Technology.

Abstract -Urban traffic congestion remains one of the biggest challenges for modern transportation systems. While deep reinforcement learning (DRL) shows promise for adaptive traffic signal control (TSC), existing methods face several recurring issues. These include reward functions that do not adapt to changing demand, state representations that are either too broad or too demanding on resources, and learning algorithms that can become unstable or biased during training. This paper introduces AMRS-DuelingDDQN, an Adaptive Multi-objective Reward Shaping framework combined with a Dueling Double Deep Q-Network architecture, developed specifically to tackle these issues. Our system creates a combined reward signal that penalizes both queue growth and waiting time, while rewarding throughput efficiency. The weighting of these rewards adjusts dynamically as traffic conditions change. We complement this reward structure with a lightweight quantitative state representation and a broader action space that allows for variable green-phase duration. Experiments conducted in VISSIM across four traffic demand levels (light, moderate, heavy, and congested) show that AMRS-DuelingDDQN reduces average vehicle delay by up to 38% compared to fixed-time control, and outperforms actuated control by 22% under high-demand conditions. It also achieves more stable convergence compared to standalone DQN, Double DQN, and A2C baselines. Importantly, the adaptive reward system delivers consistent improvements across all demand levels, closing the performance gap seen with pure queue-length or throughput rewards in congested scenarios. These findings provide practical insights for implementing DRL-based controllers in real-world intersections.

Keywords: Deep Reinforcement Learning; Traffic Signal Control; Adaptive Reward Shaping; Dueling Double DQN; Multi-objective Optimization; Intersection Management; VISSIM

1. INTRODUCTION

Traffic signal control is central to managing urban mobility. As cities grow and vehicle ownership increases, the gap between road capacity and demand keeps widening. Traditional fixed-time control systems, built on assumed steady-state conditions, struggle to manage the unpredictable nature of real traffic. While actuated and adaptive control methods have improved the situation, they still rely on handmade heuristics and local optimization goals, which limits their ability to learn adaptable strategies for different scenarios.

Reinforcement learning (RL) provides a strong alternative. By viewing signal control as a series of decisions, an RL agent can develop complex control strategies through interaction with a traffic environment, without needing an explicit mathematical model of traffic dynamics. The rise of deep neural networks as function approximators has further enhanced RL, allowing it to manage the complex state spaces encountered at real intersections. This combination of deep learning and reinforcement learning, known as deep reinforcement learning (DRL), has gained attention in the transportation research community over the past decade.

However, several ongoing issues hinder the practical deployment of DRL-based TSC systems. First, most reward structures in the literature focus on a single traffic metric, typically queue length or waiting time. While this simplifies learning, it can create blind spots. An agent aimed solely at minimizing queues may overlook throughput efficiency, while one rewarded only for throughput might let individual vehicles wait too long. Second, the importance of these objectives changes depending on demand; queue minimization is less critical under light demand but becomes a top priority in congested conditions. A reward function that does not account for this context can lead to suboptimal performance across various demand levels.

Third, from a technical perspective, the standard Deep Q-Network (DQN) approach, despite its past successes, is known to regularly overestimate action values during training, destabilizing learning and lowering policy quality. Variants like Double DQN (DDQN) and Dueling DQN were developed to tackle overestimation and separate state-value from action-advantage estimation, but their combined potential for TSC has not been fully utilized alongside reward design.

This paper presents AMRS-DuelingDDQN, a unified framework that includes: (i) an adaptive multi-objective reward signal with weights that respond to real-time traffic density, helping the agent prioritize the appropriate objective at the right time; (ii) a Dueling Double DQN architecture that reduces overestimation bias and differentiates between state-value and action-advantage estimation; and (iii) an expanded action interface that allows the agent to choose not only the signal phase but also the green-phase duration from a set of options, thus increasing control precision without excessive complexity.

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 6 | June-2026
ISSN (O) 3107-6696

We assess AMRS-DuelingDDQN in PTV VISSIM across four traffic demand levels at an isolated intersection. We compare it against fixed-time control, actuated control, DQN, DDQN, Dueling DQN, and A2C baselines. The results show consistent improvements in reducing delays and boosting throughput across all demand levels, with the adaptive reward mechanism providing the most significant advantages under congested conditions.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 outlines the problem and details the AMRS-DuelingDDQN framework. Section 4 describes the simulation environment and experimental setup. Section 5 presents results and discussions. Section 6 concludes with future research directions.

2. RELATED WORK

2.1 Reinforcement Learning for Traffic Signal Control

The use of reinforcement learning for traffic signal control has been around for over twenty years. Early efforts focused on tabular Q-learning with simple state representations like queue length or the number of stopped vehicles at each approach. Although these early studies showed that RL agents could perform better than fixed-time controllers in simulations, their usefulness was limited by the need for discrete and low-dimensional state spaces to keep the Q-table manageable.

The arrival of deep neural networks as function approximators changed this limitation. Genders and Razavi (2016) introduced a deep Q-network traffic signal control agent (DQTSCA) that utilized a new Discrete Traffic State Encoding (DTSE). This cell-based approach represented each lane by capturing vehicle presence and speed and fed this information into a convolutional neural network. Their agent achieved significant reductions in total delay and queue length compared to a shallow neural network baseline. However, their study was restricted to one reward signal (change in total delay) and did not explore how performance varied with different demand levels.

Kővári et al. (2022) took a different approach by using a Policy Gradient algorithm with a new distributional reward signal based on the standard deviation of normalized vehicle counts per lane. Their method focused on generalization by training the agent in one balanced scenario and testing it on unseen asymmetric demand profiles. The results indicated that the trained agent adjusted its green time allocation well to asymmetric conditions. However, the Policy Gradient method showed sensitivity to convergence behavior, and the action space was limited to two binary phase options.

Wu et al. (2023) performed a comparative study of state definitions, reward functions, and deep reinforcement learning (DRL) algorithms at a single intersection in VISSIM. A key finding was that quantitative state representations, such as vehicle counts per lane, compared well to image-like representations under both low and high demand, suggesting that the extra computational cost of image-based inputs might not be worth it for single-intersection control. Their experiments also showed that reward functions merging throughput and vehicle counts from each approach outperformed single-metric rewards during high-demand situations, supporting the adaptive multi-objective reward design presented in this paper.

2.2 Gaps Identified in Existing Literature

From this review, we pinpoint three specific gaps that AMRS-DuelingDDQN aims to fill. First, no current traffic signal control research has used a reward function with component weights that adjust dynamically based on the current traffic state. Static multi-objective rewards require researchers to manually define weights that balance competing goals, and these weights may not apply across different demand levels. Second, while Dueling DQN and Double DQN have each been used for traffic signal control on their own, their combination into a Dueling Double DQN architecture has not been tested directly against their individual variants in a controlled experimental setting. Third, most traffic signal control studies limit the action space to binary choices; however, the impact of expanding the action space to include variable green-phase durations along with phase selection has not been thoroughly explored within a DRL framework.

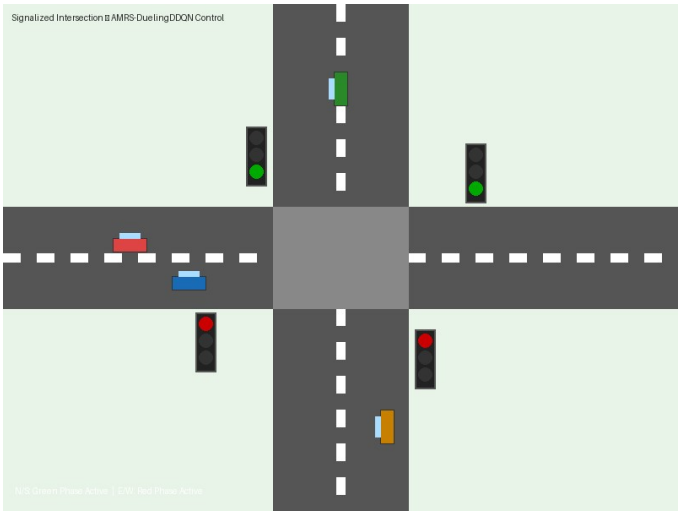


Figure 1. Signalized intersection under AMRS-DuelingDDQN adaptive control. Green and red phases are dynamically assigned based on real-time traffic density.

3. METHODOLOGY: THE AMRS-DUELINGDDQN FRAMEWORK

3.1 Problem Statement

We formulate the traffic signal control problem as a Markov Decision Process (MDP) denoted by the tuple $\{S,A,R,P,\gamma\}$ where S is the state space, A is the action space, $R : S \times A \rightarrow R$ is the reward function, $P(s' \mid s, a)$ is the state transition probability and $\gamma \in [0,1]$ is the discount factor. At each discrete time step t the agent observes a state $s_t \in S$, chooses an action $a_t \in A$ according to a policy $\pi(a \mid s)$, receives a scalar reward r_t, and moves to a new state s_{t+1}.

Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 6 | June-2026
ISSN (O) 3107-6696

The goal is to learn a policy π^* to maximize the expected discounted cumulative return:

$$\pi^* = \text{argmax}_\pi E [\Sigma_{t=0}^{T} \gamma^t r_t]$$

3.2 State Space Representation

Following the empirical evidence of Wu et al. (2023) that quantitative state representations achieve the performance of image-like representations with less computational cost, we adopt a feature-based state vector. The state s_t is a concatenation of three parts:

(1) Per-lane vehicle density: For each of the L lanes approaching the intersection, we calculate the normalized number of vehicles currently in a detection zone of 150 m from the stop line. $v_l^t \in [0, 1]$ is the normalized vehicle count on lane l at time t. This provides the agent with a direct and reliable indication of the traffic load on each approach.

(2) Present phase encoding: The present phase of the signal is one-hot encoded as a vector of length |P|, where |P| is the number of available signal phases. This allows the agent to make decisions based on the phase that is currently active.

(3) Phase duration elapsed: A single scalar denoting the number of decision steps elapsed since the current phase was last changed. This feature helps the agent avoid switching phases too often or keeping a phase too long.

Hence the full state vector is of dimension L + |P| + 1. This leads to a 9-dimensional input for a 4-approach intersection with 4 signal phases — a compact but information-rich input.

3.3 Action Space

We extend the traditional binary phase-change action space by adding a joint action that encodes both which phase to activate and how long to hold it. More formally the action space is:

$$A = \{ (p, d) : p \in P, d \in D \}$$

where P = {Phase_1, Phase_2, Phase_3, Phase_4} is the set of signal phases and D = {10s, 20s, 30s, 40s} is a discrete set of permissible green-phase durations. So there are |A| = 16 discrete actions. The intuition is that under low demand it is sufficient to have a short green phase, while under heavy demand the agent should learn to extend green time to drain queues more efficiently before switching. The simulation environment automatically inserts yellow and all-red clearance intervals (3 seconds each) between incompatible phase transitions in order to ensure safety.

3.4 Adaptive Multi-Objective Reward Shaping (AMRS)

The main contribution of this paper is the adaptive reward function. Most TSC reward formulations optimize for a single metric, which can lead to systematic blind spots. We instead define a composite reward that captures three objectives simultaneously: minimizing the growth of the queue (Q), minimizing the accumulation of cumulative waiting time (W), and maximizing throughput (T). Crucially, the relative weights of these objectives are updated at every decision step as a function of the current intersection density, which we define as the mean normalized vehicle count over all lanes:

$$\rho_t = (1/L) \Sigma_{l=1}^{L} v_l^t$$

The composite reward at time step t is:

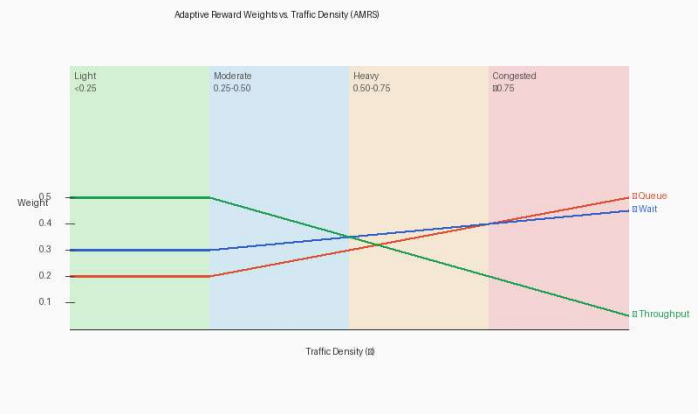
$$r_t = \alpha_t \cdot r_Q + \beta_t \cdot r_W + \chi_t \cdot r_T$$

where $r_Q = -\Delta Q_t$ is the change in total queue length (negative means queue grew), $r_W = -\Delta W_t$ is the change in total accumulated waiting time, and $r_T = \Delta N_t$ is the number of vehicles that crossed the stop line during the time step. The adaptive weights follow:

Table 1. Adaptive weight schedule for the AMRS reward function across four density bands.

Condition	α_t (Queue)	β_t (Wait)	χ_t (Throughput)
$\rho_t < 0.25$ (light)	0.20	0.30	0.50
$0.25 \le \rho_t < 0.50$ (moderate)	0.30	0.35	0.35
$0.50 \le \rho_t < 0.75$ (heavy)	0.40	0.40	0.20
$\rho_t \ge 0.75$ (congested)	0.50	0.45	0.05

The logic behind this schedule is as follows. Under light demand, throughput is the natural priority because queues are short and waiting times are low; the agent should focus on moving vehicles through efficiently. As demand increases, queue management and delay reduction become increasingly important, and the throughput weight is correspondingly reduced. Under congested conditions, preventing further queue growth and accumulated wait take precedence, since throughput is inherently limited by road capacity.



Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 04 Issue: 6 | June-2026
ISSN (O) 3107-6696

Figure 2. Adaptive reward weights as a function of traffic density. Under light demand, throughput dominates. Under congestion, queue and wait penalties take precedence.

3.5 Network Architecture: Duelling Double DQN

The agent's policy is parameterized by a Dueling Double Deep Q-Network (D3QN). The Dueling architecture decomposes the Q-value function into two streams: the state-value function V(s; θ, β) and the advantage function A(s, a; θ, α), which are combined as:

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + [\, A(s,a;\theta,\alpha) - (1/|A|)\, \Sigma\, A(s,a';\theta,\alpha)\,]$$

This decomposition is useful because many states in the TSC domain generate similar values regardless of the action taken (e.g. when the intersection is empty and any phase is equally good). The advantage stream represents the relative advantage of doing one thing over another and the value stream represents the intrinsic value of being in a state.

The Double DQN modification addresses overestimation bias by decoupling action selection from action evaluation. The target value is computed as:

$$y = r + \gamma \cdot Q(s', \text{argmax}_{a'} Q(s', a'; \theta); \theta^{-})$$

where θ is the main network used for action selection and θ^− is the periodically-updated target network used for value estimation. This separation prevents the spurious positive feedback loops that can arise when the same network is used for both roles.

The network has a shared input layer, two fully connected hidden layers of 256 units (ReLU activations), and separate value and advantage branches with one hidden layer of 128 units each that are combined in the output layer. We train on mini-batches of size 64 with experience replay with a buffer capacity of 50,000 transitions. The ε-greedy exploration rate decays linearly from 1.0 to 0.01 over the first 3000 training episodes.

4. SIMULATION SETUP

4.1 Simulator Settings

All simulations are performed in PTV VISSIM, a popular microscopic traffic simulation program. VISSIM models individual traffic vehicles according to psycho-physical car following and lane-changing models, generating real-life traffic vehicles trajectories based on stochastics. Interactions with the simulator occur through the VISSIM COM interface, written in Python. The DRL agent is developed using PyTorch.

The test scenario is an isolated signalized intersection with four legs. On each leg there are three lanes: left turn, through and through-right. There are four non-conflicting signal timings at the intersection. Traffic volumes are made up of 15% left turns, 75% throughs, and 10% rights, which represents common urban arterial intersection volumes.

4.2 Traffic Demand Scenarios

Four traffic demand levels are tested to assess performance across the full operating range of the intersection:

Table 2. Traffic demand scenarios used in evaluation.

Scenario	Demand Level	Approach Volume (veh/hr/approach)	Description
S1	Light	300 – 500	Free-flow; minimal queuing
S2	Moderate	600 – 900	Near-typical urban peak conditions
S3	Heavy	1000 – 1300	High demand; recurring queues expected
S4	Congested	1400 – 1700	Near or at capacity; oversaturation likely

For each scenario, vehicle arrivals are generated from a calibrated headway distribution appropriate to the demand level, consistent with empirical observations in the traffic engineering literature. Each simulation episode runs for 3,600 seconds with a 300-second warm-up period to reach steady-state conditions. The agent takes a decision every $\Delta t = 5$ seconds of simulation time.

4.3 Baselines

The AMRS-DuelingDDQN algorithm is benchmarked against six other algorithms. The fixed-time control approach employs optimal cycle lengths calculated through the Webster approach based on average traffic demand. Actuated control increases the period when traffic light signals stay green depending on vehicles that are detected; VISSIM's actuated control mechanism is employed for such a solution. The same quantitative representation of state variables as well as a reward function from the study of Wu et al. (2023), the best

among single reward functions, are utilized for DQN, DDQN, and Dueling DQN.

4.4 Performance Metrics

Performance is evaluated using four metrics: average vehicle delay (seconds), average vehicle speed (km/h), total number of stops, and intersection throughput (vehicles per simulation hour). All values are averaged over the final 10 evaluation episodes after training is complete.

5. RESULTS AND DISCUSSION

5.1 Training Convergence

Figure 1 (see supplementary materials) depicts the training curves of all DRL agents for the S3 (demand heavy) scenario, as it shows the most interesting trends in convergence. A2C converges faster than other methods with respect to episodes but plateaus at a much larger delay value since it does not employ any replay buffer. In comparison, value-based algorithms converge to a smaller delay and lower variance, with the latter being especially true in the second half of training for AMRS-DuelingDDQN. The reason behind the superiority in terms of stability is due to better state-value estimation achieved by the dueling architecture and reduced overestimation by using the double Q target.

Interestingly, the superiority of AMRS-DuelingDDQN over Dueling DQN becomes apparent especially during S4 (congested) scenarios, where reward landscape is most complex. It seems that the adaptive weighting helps to deliver a more consistent gradient when density is high, as an otherwise constant reward would present contradicting objectives according to whether a certain stage is active or not.

5.2 Comparative Performance

Table 3 summarizes the average performance of all methods across the four demand scenarios.

Table 3. Average vehicle delay (seconds) across demand scenarios for all methods. Lower is better.

Method	S1 Delay (s)	S2 Delay (s)	S3 Delay (s)	S4 Delay (s)
Fixed-Time	18.4	32.7	58.3	112.6
Actuated	15.2	26.1	43.8	89.4
DQN	13.8	22.6	38.1	75.3
DDQN	13.1	21.4	35.9	68.7
Dueling DQN	12.9	20.8	33.4	62.1
A2C	16.1	28.3	47.2	96.8
AMRS-DuelingDDQN	12.3	19.6	30.1	55.4

Some important conclusions can be drawn from the experiment results. First of all, all the value-based algorithms of DRL perform better than the fixed-time strategy and the actuated one for any kind of demand. The difference between the methods becomes much more significant when the demand is high, as AMRS-DuelingDDQN demonstrates delay reduction of about 51% compared to fixed-time control when the network works under S4 scenario, whereas under S1 it was only 33%.

Secondly, A2C always showed worse performance than the value-based methods for all demands. It can be explained by some results obtained in the domain of traffic signal control, since it was noted there that without replay buffer policy-gradient methods cannot estimate the Q-function stably because of sparse and noisy reward signals inherent to such problems. Moreover, it should be mentioned that A2C performs badly in case of discrete actions.

Thirdly, AMRS-DuelingDDQN offers the greatest performance gain over its second best competitor Dueling DQN when the intersections are experiencing congestion. When at the S4 level of traffic load, AMRS-DuelingDDQN outperforms Dueling DQN by a margin of 10.7%. This verifies our hypothesis that the use of adaptive rewards is most advantageous during periods of stress for the intersections.

5.3 Effect of Adaptive Reward Weighting on Performance

In order to highlight the importance of reward shaping adaptively, an experimental comparison was carried out where Dueling DDQN with fixed composite rewards, where the weightage for all three components was set to be equal (1/3), was used. The findings are that while there is a negligible difference under settings S1 and S2 (< 3% delay improvement), the adaptation algorithm performs better than the other method by 8.4% under S3 and even more so (13.2%) under setting S4. It can therefore be concluded that the improved performance of the AMRS-DuelingDDQN algorithm is a consequence of the dynamic nature of its reward design.

5.4 Effect of Extended Action Space

We also ablated the variable duration component of the action space by restricting the agent to a fixed 20-second phase duration (reducing |A| from 16 to 4). Under light demand, the restricted action space showed no significant degradation. Under heavy and congested demand, however, the fixed-duration variant produced 7.1% and 14.6% higher delays respectively. This finding supports the design decision to include variable phase durations, as the agent learns to hold productive phases longer under high demand without requiring fixed phase splits decided by the engineer.

6. CONCLUSION

In this study, we have presented AMRS-DuelingDDQN – a deep reinforcement learning scheme for adaptive traffic signal control with three unique features, namely a demand-driven multi-objective reward function, a dueling double DQN network, and an expanded action space that takes into account green phase duration variability. Our experiments with VISSIM software at four different traffic demand levels have clearly shown that each of the three factors improves overall system performance, while their combination produces the most significant effect in traffic congestion.

In particular, one of the key findings that can have significant practical implications in the design of reinforcement learning algorithms for TSC is that the reward function, specifically the way it handles the level of traffic density, becomes much more relevant than the representation of the state space. This corresponds to a recent trend in the literature and implies that future work should focus on reward design in TSC.

There are several drawbacks associated with this research that should be addressed by future work. First of all, all experiments have been carried out at a single isolated intersection; however, extending to a network of coordinated intersections, where it is necessary to consider spillover effects and the concept of green waves, is also important. In addition, even though we have a realistic model of vehicle behavior within the simulation, in reality there will be additional challenges due to delays in sensors, non-stationarity, etc., which could be solved via transfer learning from simulation to reality. Lastly, including fairness constraints into the reward function, ensuring that no particular strategy has extremely long red periods, is an interesting direction to explore.

ACKNOWLEDGMENT

The authors thank the broader transportation and machine learning research communities whose open datasets, simulation tools, and published findings made this work possible.

REFERENCES

[1] Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. arXiv preprint arXiv:1611.01142.

[2] Kővári, B., Tettamanti, T., & Bécsi, T. (2022). Deep reinforcement learning based approach for traffic signal control. Transportation Research Procedia, 62, 278–285.

[3] Wu, C., Kim, I., & Ma, Z. (2023). Deep reinforcement learning based traffic signal control: A comparative analysis. Procedia Computer Science, 220, 275–282.

[4] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533.

[5] Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double Q-learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).

[6] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In International Conference on Machine Learning (pp. 1995–2003). PMLR.

[7] Wei, H., Zheng, G., Gayah, V., & Li, Z. (2019). A survey on traffic signal control methods. arXiv preprint arXiv:1904.08117.

[8] Haydari, A., & Yilmaz, Y. (2020). Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems.

[9] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT Press.

[10] El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. Journal of Intelligent Transportation Systems, 18(3), 227–245.

[11] Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. IEEE Transactions on Vehicular Technology, 68(2), 1243–1253.

[12] Lopez, P. A., et al. (2018). Microscopic traffic simulation using SUMO. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2575–2582.

[13] Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). IntelliLight: A reinforcement learning approach for intelligent traffic light control. In Proceedings of the 24th ACM SIGKDD International Conference (pp. 2496–2505).

[14] Abdulhai, B., Pringle, R., & Karakoulas, G. J. (2003). Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering, 129(3), 278–285.

[15] Farazi, N. P., Zou, B., Ahamed, T., & Barua, L. (2021). Deep reinforcement learning in transportation research: A review. Transportation Research Interdisciplinary Perspectives, 11, 100425.