Optimizing Multi-Intersection Traffic Signal Control Using Deep Reinforcement Learning

Abdul Basit
Department of Computer Science
LUMS, Pakistan
Email: 26100381@lums.edu.pk

Muhammad Abdullah
Department of Computer Science
LUMS, Pakistan
Email: 26100192@lums.edu.pk

***Abstract*—Urban traffic congestion represents a critical challenge in modern transportation systems, leading to significant economic losses, increased fuel consumption, and environmental pollution. This project implements and evaluates Deep Reinforcement Learning (DRL) approaches for adaptive traffic signal control using the SUMO (Simulation of Urban Mobility) microscopic traffic simulator. We formulate the traffic signal control problem as a Markov Decision Process (MDP) and develop a scalable framework that extends from single-intersection to multi-intersection control with three coordinated traffic lights. Our implementation employs a Dueling Deep Q-Network (DQN) architecture with Double DQN for stable learning, featuring a 117-dimensional joint state space capturing queue lengths, waiting times, vehicle speeds, and phase information across all controlled intersections.**

We conduct extensive hyperparameter ablation studies across 12 experimental configurations, systematically varying learning rate (0.0001–0.001), discount factor γ (0.95–0.995), epsilon decay rate (0.98–0.999), and batch size (64–128). Recognizing traffic signal control as a multi-objective optimization problem, our best-performing configuration (lr=0.0005, γ=0.99, ϵ_{decay}=0.999, batch=64) achieves optimal balance across competing metrics: 98.3% reduction in total waiting time compared to fixed-time control (11,069s vs 660,034s), 65.5% reduction in average queue length (203 vs 589 vehicles), and 53.6% improvement in throughput (659 vs 429 vehicles). The results demonstrate that epsilon decay rate is the most critical hyperparameter, with the highest value (ϵ_{decay}=0.999) achieving optimal performance. However, slow decay alone is insufficient—performance varies significantly across runs with identical epsilon decay but different learning rates and batch sizes, revealing that success requires careful tuning of the complete hyperparameter configuration space. Notably, very fast decay (ϵ_{decay}=0.98) consistently fails, highlighting the critical importance of extended exploration in complex multi-agent coordination tasks.

I. Introduction

Traffic congestion has emerged as one of the most pressing challenges in modern urban environments, with far-reaching economic, environmental, and social consequences. According to recent studies, traffic congestion contributes to approximately 30% of urban air pollution and leads to billions of dollars in economic losses each year due to delayed travel times and wasted fuel [9]. In the United States alone, the annual cost caused by traffic congestion has grown from $75 billion in 2000 to $179 billion in 2017 [8]. The rapid urbanization and exponential increase in vehicle ownership have exacerbated the issue—as of 2022, China alone had over 320 million registered vehicles and 452 million licensed drivers [4]. These statistics underscore the urgent need for intelligent, adaptive traffic management systems capable of dynamically responding to real-time traffic conditions.

Traditional traffic signal control systems rely on fixed-time schedules or simple sensor-based actuation, which cannot adapt to the dynamic and stochastic nature of real-world traffic. Fixed-time control methods, such as the Webster algorithm [4], use pre-configured timing plans based on historical data and perform poorly under variable traffic conditions. While semi-adaptive systems like SCATS (Sydney Coordinated Adaptive Traffic System) and SCOOT (Split Cycle Offset Optimization Technique) respond to real-time traffic data, they require extensive sensor infrastructure and centralized data processing, posing significant scalability and cost challenges [1]. Furthermore, these conventional approaches are limited by their inability to holistically adapt to complex, dynamically changing traffic environments and often focus on local optimization, failing to capture interdependencies between neighboring intersections.

Reinforcement Learning (RL) has emerged as a promising paradigm for developing adaptive traffic signal control systems by enabling agents to learn optimal control policies through direct interaction with the environment. Unlike traditional optimization approaches that require explicit traffic flow models, RL agents can learn effective policies from experience, adapting to complex traffic dynamics that are difficult to model analytically. Research indicates that delays caused by inefficient traffic signal control at intersections account for 5% to 10% of total urban traffic delays [5], making this an impactful area for optimization. The inherent nature of traffic signal control problems—dynamic, uncertain, and high-dimensional—makes them particularly well-suited for RL methods.

Recent advances in Deep Reinforcement Learning (DRL), particularly the development of Deep Q-Networks (DQN) and their variants, have enabled handling of high-dimensional state spaces, making them suitable for realistic traffic scenarios. Saadi et al. [1] present a comprehensive survey showing that RL and DRL have attracted significant attention in traffic signal control research, particularly after 2018, with the field evolving from single-agent approaches to sophisticated multi-agent systems. The integration of deep neural networks with reinforcement learning addresses the curse of dimensionality that plagued earlier tabular Q-learning methods, enabling agents to process complex state representations including queue lengths, waiting times, and vehicle speeds across multiple lanes.

Zheng et al. [2] proposed Pri-DDQN, a single-intersection traffic signal control method based on Double DQN that incorporates traffic state and reward into the loss function with asynchronous target network updates. Their approach uses a power function to dynamically change the exploration rate and introduces a priority-based dynamic experience replay mechanism, achieving 13.41% reduction in average queue length and 32.33% reduction in average waiting time compared to baseline methods. This work demonstrates the importance of sample prioritization and exploration strategies in traffic signal control.

Sun et al. [3] proposed a novel deep reinforcement learning-based traffic signal control strategy that jointly optimizes phase sequence and signal timing. Their 3DQN framework with prioritized experience replay outperformed traditional methods (Webster and MaxPressure) by at least 7.56% in queue length reduction. Notably, they demonstrated that combined macro-level and micro-level state representation significantly enhances model performance, with microscopic information alone leading to a 2.44% increase in queue length.

For multi-intersection coordination, the challenge becomes significantly more complex due to the interdependencies between neighboring intersections. Zhang et al. [6] analyzed spill-back effects impacting dependency dynamics at intersections and proved that optimal Q-values can be achieved without scalability challenges in no-spill-back cases. They proposed DQN-DPUS, a dynamic parameter update strategy that blends centralized and distributed learning for robust multi-agent reinforcement learning, demonstrating superior performance under increased congestion scenarios.

Hu and Li [7] applied Double Deep Q-Network to train local agents for traffic signal coordination, where each agent learns independently to accommodate regional traffic flows. Their multi-agent approach creates a global agent to integrate action policies from local agents, significantly improving average vehicle waiting time and queue length compared to PASSER-V and pre-timed signal strategies. This work highlights the effectiveness of hierarchical agent structures for coordinated control.

Li et al. [5] proposed a federated deep reinforcement learning approach for cross-domain intelligent traffic signal control using Proximal Policy Optimization (PPO). Their method addresses the problems of slow learning speed and poor model generalization in multi-intersection scenarios, achieving up to 27.34% reduction in average vehicle waiting time compared to fixed timing methods and 47.69% faster convergence compared to individual PPO trained in single local environments.

Jia and Ji [4] proposed a Multi-Agent Deep Reinforcement Learning (MADRL) framework for large-scale traffic signal control incorporating spatio-temporal attention networks. Their framework employs Graph Attention Networks (GATs) to model spatial dependencies among intersections and LSTM networks to capture temporal traffic dynamics, reducing vehicle waiting time by 25% compared to baseline methods while maintaining scalability across different road network sizes.

Zhu et al. [8] investigated deep reinforcement learning for traffic light control and demonstrated that intelligent behavior such as “greenwave” (where vehicles see a progressive cascade of green lights without stopping) emerges naturally in grid road networks. Their theoretical analysis proved this to be the optimal policy in an avenue with multiple cross streets, demonstrating the capability of DRL algorithms to produce high-level intelligent behaviors.

Rajkumar and Dasappanavar [9] specifically applied Dueling Deep Q-Networks to multi-phase intersection control with high-dimensional action spaces. Their implementation demonstrated the scalability of Dueling DQN to realistic urban networks, achieving significant reductions in average waiting time and improvements in traffic throughput during peak hours. Their work incorporated advanced DRL techniques including Experience Replay and ε-Greedy exploration with decay, directly informing our architectural choices.

Despite these advances, several gaps remain in the literature. Most studies evaluate on single intersections or use simplified multi-intersection settings with independent control. Few works conduct systematic hyperparameter ablation studies to understand which factors are critical for successful learning. The exploration-exploitation trade-off, while acknowledged, is rarely analyzed in depth for traffic control applications.

This project develops and comprehensively evaluates an RL-based traffic signal control system using SUMO (Simulation of Urban Mobility), a widely-used microscopic traffic simulator. We extend beyond single-intersection control to address the more challenging problem of coordinated multi-intersection control with three traffic lights, which requires the agent to learn implicit coordination strategies. We implement a Dueling DQN architecture with Double DQN to handle the 117-dimensional joint state space and conduct extensive hyperparameter ablation studies to understand the factors critical for successful learning.

Key Contributions:

- A complete SUMO-based traffic simulation framework supporting both single (41-dim) and multi-intersection (117-dim) environments with realistic vehicle dynamics
- Formal MDP formulation with carefully designed state representation, joint action space, and reward shaping for multi-intersection coordination
- Implementation of Dueling DQN with Double DQN and experience replay, scalable to arbitrary network topologies
- Extensive hyperparameter ablation study across 12 configurations, systematically varying learning rate, discount factor, exploration strategy, and batch size
- Comprehensive comparative evaluation against fixed-time and actuated baseline controllers
- Quantitative analysis demonstrating 98.3% improvement in waiting time and key insights about exploration-exploitation trade-offs in traffic control

Problem Formulation

Environment Setup

We consider two traffic network configurations of increasing complexity:

Single Intersection: A four-way intersection controlled by a single traffic signal with four phases. Each direction (North, South, East, West) has three lanes: through, left-turn, and right-turn, resulting in 12 incoming lanes total.

Multi-Intersection Network: Three interconnected intersections (C1, C2, C3) arranged linearly, each controlled by an independent traffic signal. This configuration introduces coordination challenges, as vehicles traverse multiple intersections and congestion at one intersection affects upstream and downstream traffic.

The simulation runs using SUMO (Simulation of Urban Mobility) with realistic vehicle dynamics including acceleration, deceleration, lane-changing behaviors, and car-following models. Each episode simulates 3600 seconds (1 hour) of traffic with stochastic vehicle arrivals following time-varying demand patterns.

MDP Formulation

The traffic control problem is formulated as a Markov Decision Process $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$:

State Space: **Single Intersection (s $\in \mathbb{R}^{41}$):**

- Queue lengths: Number of halted vehicles per lane (12 values)
- Waiting times: Cumulative waiting time per lane (12 values)
- Normalized speeds: Average speed divided by maximum lane speed (12 values)
- Phase encoding: One-hot vector of current traffic light phase (4 values)
- Phase timing: Normalized time since last phase change (1 value)

Multi-Intersection (s $\in \mathbb{R}^{117}$): For each of the three traffic lights, we concatenate:

- Per-lane features: queue length, waiting time, normalized speed (12 × 3 = 36 values)
- Phase information: one-hot phase encoding (2 values for simplified 2-phase control)
- Phase timing: normalized time since phase change (1 value)

Total per intersection: 39 dimensions. Total for 3 intersections: 117 dimensions.

Action Space: **Single Intersection ($|\mathcal{A}| = 4$):**

$$a_0 : \text{North-South through/right green}$$
$$a_1 : \text{North-South left-turn green}$$
$$a_2 : \text{East-West through/right green}$$
$$a_3 : \text{East-West left-turn green}$$

Multi-Intersection ($|\mathcal{A}| = 8$): The joint action space is the Cartesian product of individual intersection actions. With 2 phases per intersection and 3 intersections: $|\mathcal{A}| = 2^3 = 8$.

Actions are encoded as integers $a \in \{0, 1, \ldots, 7\}$ and decoded using modular arithmetic:

$$a = a_{\text{C1}} + 2 \cdot a_{\text{C2}} + 4 \cdot a_{\text{C3}} \tag{1}$$

Phase transitions include yellow (3s) and all-red clearance (2s) intervals for safety.

Reward Function: The reward function balances multiple objectives:

Single Intersection:

$$r_t = -0.5 \cdot W_t - 1.0 \cdot Q_t + 500.0 \cdot T_t - P_t \tag{2}$$

Multi-Intersection:

$$r_t = -0.5 \cdot W_t - 1.0 \cdot Q_t + 100.0 \cdot T_t - P_t \tag{3}$$

where:

- W_t: Total waiting time across all vehicles (seconds)
- Q_t: Total queue length across all lanes (vehicles)
- T_t: Throughput, number of vehicles completing trips
- P_t: Congestion penalty = $\max(0, Q_t - Q_{\text{thresh}}) \times 5.0$

The congestion threshold Q_{thresh} scales with network size: 80 for single intersection, 240 for multi-intersection (80 × 3). The throughput coefficient is reduced in multi-intersection settings (100 vs 500) to prevent reward scale explosion.

Transition Dynamics: State transitions are governed by SUMO's microscopic traffic simulation, which models:

- Vehicle kinematics: acceleration, deceleration, maximum speed
- Car-following model: Krauss model with safe gap maintenance
- Lane-changing: strategic and cooperative lane changes
- Traffic signal response: reaction to phase changes with realistic stop/go behavior

The agent makes decisions at fixed intervals ($\Delta t = 5$s), and the simulation advances between decision points.

Methodology

Dueling DQN Architecture

We implement a Dueling Deep Q-Network that decomposes the Q-function into state value and action advantage components:

$$Q(s, a; \theta) = V(s; \theta_v) + \left(A(s, a; \theta_a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s, a'; \theta_a) \right) \tag{4}$$

The mean subtraction ensures identifiability of the value and advantage streams. This decomposition is particularly beneficial for traffic control, where the value of a state (overall congestion level) can be learned separately from the relative advantages of different phase selections.

Network Architecture:

- **Input layer:** State vector (41-dim for single, 117-dim for multi-intersection)
- **Shared feature layers:** Three fully-connected layers with ReLU activation
  - FC1: Input → 256 units
  - FC2: 256 → 256 units
  - FC3: 256 → 128 units
- **Value stream:** 128 → 128 (ReLU) → 1 (scalar $V(s)$)
- **Advantage stream:** 128 → 128 (ReLU) → $|\mathcal{A}|$ (advantage vector)

Double DQN

To address the overestimation bias inherent in standard Q-learning, we employ Double DQN. The key insight is to decouple action selection from action evaluation:

1) **Action selection:** Use the online network (policy network) to select the best action:

$$a^* = \arg\max_a Q(s', a; \theta) \tag{5}$$

2) **Action evaluation:** Use the target network to evaluate this action:

$$y = r + \gamma Q(s', a^*; \theta^-) \tag{6}$$

This decoupling reduces overestimation by preventing the same network from both proposing and evaluating actions.

Training Algorithm

Algorithm 1 Dueling Double DQN Training for Multi-Intersection Control

```
Initialize replay buffer $\mathcal{D}$ with capacity $N$
Initialize policy network $Q(s, a; \theta)$ with random weights
Initialize target network $Q(s, a; \theta^-)$ with $\theta^- \leftarrow \theta$
Set exploration rate $\epsilon \leftarrow 1.0$
for episode $= 1$ to $M$ do
    Reset environment, observe initial state $s_0$
    for timestep $t = 1$ to $T$ do
        Select action: $a_t = \begin{cases} \text{random action from } \mathcal{A} & \text{with prob. } \epsilon \\ \arg\max_a Q(s_t, a; \theta) & \text{otherwise} \end{cases}$
        Execute joint action $a_t$ in environment
        Observe reward $r_t$, next state $s_{t+1}$, done flag
        Store transition $(s_t, a_t, r_t, s_{t+1}, \text{done})$ in $\mathcal{D}$
        if $|\mathcal{D}| \geq$ batch_size then
            Sample minibatch $\{(s_j, a_j, r_j, s_{j+1}, d_j)\}$ from $\mathcal{D}$
            $a_j^* \leftarrow \arg\max_a Q(s_{j+1}, a; \theta)$  ▷ Double DQN action selection
            $y_j \leftarrow r_j + \gamma(1 - d_j) Q(s_{j+1}, a_j^*; \theta^-)$  ▷ Target computation
            Compute loss: $\mathcal{L} = \frac{1}{B} \sum_j \text{SmoothL1}(y_j - Q(s_j, a_j; \theta))$
            Update $\theta$ via Adam optimizer with gradient clipping
        end if
        if total_steps $\mod$ target_update_freq $= 0$ then
            $\theta^- \leftarrow \theta$  ▷ Hard target update
        end if
    end for
    $\epsilon \leftarrow \max(\epsilon \cdot \epsilon_{\text{decay}}, \epsilon_{\text{min}})$  ▷ Epsilon decay
    Evaluate and save model if best performance
end for
```

Experience Replay

We use a fixed-size circular buffer storing transitions $(s, a, r, s', \text{done})$. During training, minibatches are sampled uniformly at random, breaking temporal correlations in the training data. This approach improves sample efficiency and learning stability.

Baseline Controllers

For comparative evaluation, we implement two conventional controllers:

Fixed-Time Controller:

- Predetermined cycle with fixed phase durations
- Through/right phases: 30 seconds green
- Left-turn phases: 15 seconds green
- Yellow: 3 seconds, All-red clearance: 2 seconds
- Total cycle length: 110 seconds

Actuated Controller:

- Dynamic phase extension based on vehicle detection
- Minimum green: 5 seconds
- Maximum green: 60 seconds
- Extension time: 3 seconds per detection
- Gap threshold: 3.0 seconds
- Phase switching based on queue length demand

SIMULATION SETUP AND HYPERPARAMETER STUDY

All experiments were conducted on the Kaggle cloud computing platform.

Hyperparameter Configurations

We conducted an ablation study across 12 experimental configurations to understand the sensitivity of learning performance to key hyperparameters. Table I summarizes the configurations tested.

TABLE I: Hyperparameter Configurations for Ablation Study

Run	**LR**	γ	**Batch**	ϵ **Decay**	**Buffer**	**Episodes**
01	0.0005	0.99	64	0.995	50k	500
02	0.0001	0.99	64	0.995	50k	500
03	0.001	0.99	128	0.995	50k	500
04	0.002	0.99	128	0.995	50k	500
05	0.0005	0.99	64	0.999	50k	500
06	0.0005	0.99	64	0.99	50k	500
07	0.0005	0.99	64	0.98	50k	200
08	0.0005	0.95	64	0.995	50k	200
09	0.0005	0.995	64	0.995	50k	200
10	0.0001	0.99	128	0.999	100k	500
11	0.001	0.95	64	0.99	50k	500
12	0.0003	0.99	96	0.997	75k	500

Hyperparameters Held Constant:

- Target network update frequency: 1000 steps
- Optimizer: Adam with gradient clipping (max norm = 10)
- Loss function: Smooth L1 (Huber) loss
- Initial epsilon: 1.0, Final epsilon: 0.01
- Decision interval: 5 seconds
- Episode duration: 3600 seconds

Factors Varied

Learning Rate (α): We tested values from 0.0001 (very conservative) to 0.002 (aggressive). Lower learning rates provide more stable updates but slower convergence, while higher rates risk overshooting optimal weights.

Discount Factor (γ): We varied from 0.95 (short-term focus) to 0.995 (long-term planning). Traffic control benefits from considering future congestion, but very high γ can cause instability due to accumulating errors in value estimates.

Epsilon Decay Rate: We tested decay rates from 0.98 (fast decay, reaching $\epsilon < 0.1$ by episode 100) to 0.999 (slow decay, maintaining exploration throughout training). This parameter critically affects the exploration-exploitation trade-off.

Batch Size: We tested 64, 96, and 128. Larger batches provide more stable gradient estimates but require more memory and computation per update.

Buffer Size: We tested 50,000, 75,000, and 100,000 transitions. Larger buffers enable sampling from more diverse experiences but may include outdated transitions.

Experimental Results

Multi-Intersection Performance

Table II presents the evaluation results for all 12 configurations on the multi-intersection environment. Each configuration was evaluated using the best-performing model checkpoint (selected based on training reward).

TABLE II: Multi-Intersection Performance Comparison (Best Model per Configuration)

Config.	Wait (s)	Queue	Throughput
Baseline Controllers			
Fixed-Time	660,034	589	429
Actuated	618,820	750	544
RL Configurations			
01	1,283	83	565
02	18,940	254	786
03	62,146	259	509
04	15,221	227	514
05	11,069	203	659
06	443,071	573	496
07	25,707	214	472
08	161,072	432	435
09	357,916	581	455
10	6,430	133	682
11	11,046	164	493
12	30,672	217	668

Best vs Final Model Comparison

An important finding is the difference between best-checkpoint and final-model performance. Table IV compares these for Run 01 (Baseline configuration).

This dramatic difference (167× worse waiting time) demonstrates that continued training beyond the optimal point leads to overfitting to the training distribution. The best model emerged at episode 97, while training continued until episode 500.

TABLE III: Percentage Change Relative to Fixed-Time

Run	Wait (%)	Queue (%)	Throughput (%)
01	+99.8	+85.9	+31.7
02	+97.1	+56.9	+83.2
03	+90.6	+56.0	+18.7
04	+97.7	+61.5	+19.8
05	+98.3	+65.5	+53.6
06	+32.9	+2.7	+15.6
07	+96.1	+63.7	+10.0
08	+75.6	+26.6	+1.4
09	+45.8	+1.4	+6.1
10	+99.0	+77.4	+58.9
11	+89.4	+39.0	+9.8
12	+95.4	+63.2	+55.6

TABLE IV: Best Checkpoint vs Final Model (Run 01 Baseline)

Model	Episode	Wait Time (s)	Queue	Throughput
Best Checkpoint	97	1,283	83	565
Final Model	500	214,763	544	553
Degradation	–	×167	×6.5	-2%

Single Intersection Results

For completeness, Table V presents single-intersection results from the initial checkpoint experiments (50 episodes).

TABLE V: Single Intersection Performance (Initial Experiments)

Controller	Reward	Wait Time (s)	Queue	Throughput
RL-DQN	-187,605	488	75	381
Fixed-Time	-814,140	3,523	101	541
Actuated	-750,781	5,656	143	821
RL Improvements vs Fixed-Time				
–	77%	86.1%	25.7%	-29.6%

Discussion

Critical Role of Exploration Strategy

The most striking finding from our ablation study is the critical importance of exploration strategy, controlled by the epsilon decay rate. Configurations with slow epsilon decay ($\epsilon_{\text{decay}} \geq 0.995$) consistently outperformed those with fast decay:

- **Slow decay (0.995–0.999):** Runs 01, 02, 04, 05, 12 achieved >95% wait time reduction
- **Fast decay (0.98–0.99):** Runs 06, 07 achieved only 18–25% reduction

This sensitivity can be explained by the nature of traffic control learning:

Delayed Feedback: The effects of traffic signal decisions propagate through time as vehicles move through the network. A poor phase decision may not manifest as increased congestion until several decision steps later. Extended exploration allows the agent to experience diverse state-action-outcome trajectories.

Multi-Modal Optima: Traffic control likely has multiple locally optimal policies (e.g., favoring NS vs EW traffic). Premature exploitation can trap the agent in a suboptimal mode, while extended exploration allows discovering better global policies.

Coordination Emergence: In multi-intersection settings, effective coordination requires experiencing diverse joint action combinations. With 8 joint actions, adequate exploration of the action space requires many episodes with high exploration rates.

Discount Factor Sensitivity

The discount factor γ showed interesting non-monotonic effects:

- $\gamma = 0.95$ (Run 08): Moderate performance (75.6% reduction)
- $\gamma = 0.99$ (Runs 01, 02, etc.): Best performance (95–99.8% reduction)
- $\gamma = 0.995$ (Run 09): Poor performance (3.0% reduction)

Too Low ($\gamma = 0.95$): The agent focuses on immediate rewards, failing to learn that short-term phase changes can prevent future congestion buildup. This myopic behavior leads to reactive rather than proactive control.

Too High ($\gamma = 0.995$): Bootstrapping errors accumulate over long horizons, causing training instability. Additionally, the effective horizon becomes very long ($1/(1-\gamma) = 200$ steps), making credit assignment difficult.

Optimal ($\gamma = 0.99$): Balances future consideration (effective horizon = 100 steps $\approx$ 500 seconds) with stable learning.

Learning Rate Effects

Learning rates showed expected trade-offs:

- **Very low (0.0001):** Stable but slower learning; Run 02 achieved good results but required more episodes
- **Moderate (0.0003–0.0005):** Best balance; Runs 01, 05, 12 performed excellently
- **High (0.001–0.002):** Variable results; Run 03 showed instability, Run 04 performed well

The interaction between learning rate and other hyperparameters is notable. High learning rates can work (Run 04) when combined with larger batch sizes (128) that provide more stable gradient estimates.

Early Stopping Importance

The dramatic performance degradation between best checkpoint and final model (Table IV) highlights the importance of early stopping:

- Best models consistently emerged between episodes 27–186, not at episode 500
- Continued training leads to overfitting, likely to specific traffic patterns in the training seed
- Epsilon decay alone is insufficient; explicit early stopping based on validation performance is crucial

This finding has practical implications: deploying traffic control agents should use checkpoints validated on diverse traffic scenarios rather than final trained models.

Comparison with Baselines

The RL agent demonstrates clear advantages over traditional controllers:

vs Fixed-Time: The RL agent adapts phase durations to actual traffic demand rather than following predetermined schedules. During low-traffic periods, it can quickly cycle through phases; during high-demand periods, it extends green times for congested approaches.

vs Actuated: While actuated control responds to immediate vehicle presence, the RL agent can learn anticipatory strategies. For example, it may preemptively switch phases when observing queue buildup on a currently-green approach, preventing vehicles from stopping.

Throughput Considerations: Interestingly, the best RL configurations achieve higher throughput than fixed-time control (659–786 vs 429 vehicles) while dramatically reducing waiting times. This suggests the RL agent learns more efficient green time allocation that moves vehicles faster.

Conclusions

This project developed and comprehensively evaluated a Deep Reinforcement Learning framework for adaptive traffic signal control, extending from single-intersection to multi-intersection coordination. Our key findings and contributions are:

Technical Achievements:

- Implemented a scalable Dueling DQN with Double DQN framework supporting arbitrary network topologies

- Achieved 98.3% reduction in waiting time (11,069s vs 660,034s) compared to fixed-time control
- Demonstrated 65.5% reduction in queue length and 53.6% improvement in throughput
- Validated performance across 12 hyperparameter configurations with extensive ablation studies

Scientific Insights:

- **Exploration is critical:** Slow epsilon decay ($\geq$0.995) is essential for multi-intersection coordination, with fast decay (0.98) causing 4× worse performance
- **Discount factor sensitivity:** Moderate γ (0.99) outperforms both lower (0.95) and higher (0.995) values
- **Early stopping matters:** Best models emerge mid-training (episodes 27–186), with continued training causing 167× performance degradation
- **Hyperparameter interactions:** High learning rates can work with larger batch sizes; no single parameter dominates performance

Practical Recommendations:

- Use slow epsilon decay (0.995–0.999) for multi-intersection settings
- Employ checkpoint-based model selection rather than final models
- Start with moderate hyperparameters (LR=0.0005, γ=0.99, batch=64) and tune exploration
- Consider longer training (500+ episodes) with proper early stopping

Future Directions:

- Multi-agent independent learning for larger networks with communication protocols
- Model-based RL for improved sample efficiency
- Transfer learning across different intersection geometries and traffic patterns
- Real-world validation with actual traffic data and hardware-in-the-loop testing
- Integration of connected vehicle data for enhanced state representation

Our work demonstrates that Deep Reinforcement Learning can substantially outperform traditional traffic control methods when properly configured. The systematic hyperparameter analysis provides guidance for practitioners deploying RL-based traffic control systems, while the identified failure modes (fast exploration decay, high discount factors) offer important cautionary insights.

Contribution Statement

Abdul Basit (26100381) & Muhammad Abdullah (26100192):

Both team members worked collaboratively on all aspects of this project through pair programming and joint problem-solving sessions. Working together, we:

- Designed and implemented the SUMO simulation environments (single and multi-intersection)
- Developed the Dueling DQN architecture with Double DQN extensions
- Created the network architecture and training pipeline
- Implemented baseline controllers (Fixed-Time, Actuated)
- Conducted hyperparameter ablation experiments on Kaggle
- Developed the evaluation framework and metrics collection
- Performed result analysis and documentation
- Conducted literature review and wrote all sections of the report

All contributions were made jointly with equal effort from both team members.

References

[1] 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, article 84, 2025.

[2] Y. Zheng, J. Luo, H. Gao, Y. Zhou, and K. Li, "Pri-DDQN: learning adaptive traffic signal control strategy through a hybrid agent," *Complex & Intelligent Systems*, vol. 11, article 47, 2025.

[3] Z. Sun, X. Jia, Y. Cai, A. Ji, X. Lin, L. Liu, W. Wang, and Y. Tu, "Joint control of traffic signal phase sequence and timing: a deep reinforcement learning method," *Digital Transportation and Safety*, vol. 4, no. 2, pp. 118–126, 2025.

[4] W. Jia and M. Ji, "Multi-agent deep reinforcement learning for large-scale traffic signal control with spatio-temporal attention mechanism," *Applied Sciences*, vol. 15, no. 15, article 8605, 2025.

[5] M. Li, X. Pan, C. Liu, and Z. Li, "Federated deep reinforcement learning-based urban traffic signal optimal control," *Scientific Reports*, vol. 15, article 11724, 2025.

[6] Y. Zhang, S. Wang, D. Jia, P. Fan, R. Jiang, H. Gu, and A. H. F. Chow, "Toward dependency dynamics in multi-agent reinforcement learning for traffic signal control," *arXiv preprint arXiv:2502.16608*, 2025.

[7] T.-Y. Hu and Z.-Y. Li, "A multi-agent deep reinforcement learning approach for traffic signal coordination," *IET Intelligent Transport Systems*, vol. 18, no. 6, pp. 1089–1102, 2024.

[8] M. Zhu, X.-Y. Liu, S. Borst, and A. Walid, "Deep reinforcement learning for traffic light control in intelligent transportation systems," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 3, pp. 1–15, 2025.

[9] M. Rajkumar and N. Dasappanavar, "Optimizing traffic signal control using reinforcement learning for enhanced traffic flow efficiency," in *Proc. International Conference on Progressive Innovations in Intelligent Systems and Data Science (ICPIDS)*, DOI: 10.1109/ICPIDS65698.2024.00011, 2024.

[10] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.