Optimization of Traffic Lights: A Deep Reinforcement Learning Approach for Adaptive and Emergency-Aware Control

Executive Summary

Urban centers worldwide face escalating traffic congestion, a critical challenge that impedes mobility, wastes energy, and compromises the efficacy of emergency services. Traditional traffic signal control systems, often static or limited in their adaptability, are proving insufficient for managing dynamic and unpredictable traffic flows. This report synthesizes cutting-edge research in Deep Reinforcement Learning (DRL) for Adaptive Traffic Signal Control (ATSC), focusing specifically on systems that can automatically adjust to real-time traffic conditions and prioritize emergency vehicles. A comprehensive analysis of recent literature reveals the transformative potential of DRL in developing intelligent traffic management systems that learn and evolve through interaction with the environment. Key findings highlight diverse approaches to state representation, action space design, and reward function engineering, alongside the crucial role of multi-agent coordination and robust communication infrastructure. The report concludes with strategic recommendations for developing an advanced traffic optimization model, emphasizing the integration of multimodal considerations, sophisticated DRL architectures, and mechanisms to

address real-world data imperfections and scalability challenges.

1. Introduction: The Landscape of Intelligent Traffic Management

1.1. The Global Challenge of Urban Traffic Congestion

The rapid pace of urbanization and the continuous increase in vehicle ownership have placed unprecedented strain on existing urban transportation networks. This escalating demand frequently leads to severe traffic congestion, manifesting as prolonged travel delays, significant energy waste, heightened risks of vehicular accidents, and adverse environmental impacts due to increased emissions.[1] Conventional traffic signal control systems, which typically operate on fixed-time schedules or employ rudimentary actuated logic, are inherently limited in their capacity to respond effectively to the dynamic and often chaotic nature of real-time traffic patterns and unforeseen events. These systems, relying on historical data or basic sensor inputs, lack the agility required to optimize flow under fluctuating conditions.[1] Furthermore, the physical expansion of transportation infrastructure, such as building new roads or widening existing ones, is frequently constrained by spatial limitations, prohibitive economic costs, and growing environmental concerns. This necessitates a strategic shift towards leveraging advanced technological solutions to enhance the efficiency and utilization of existing infrastructure.[1] The fundamental challenge lies not merely in the volume of traffic, but in its inherent

dynamism and unpredictability. This characteristic underscores why static or rule-based control systems are fundamentally inadequate, thereby compelling the development of intelligent, adaptive solutions capable of continuous learning and evolution.

1.2. Evolution of Adaptive Traffic Signal Control (ATSC)

The pursuit of more responsive traffic management has driven the evolution of Adaptive Traffic Signal Control (ATSC) systems. Early iterations, such as SCOOT and SCATS, represented significant advancements by attempting to adjust signal timings based on observed traffic conditions. However, these systems often relied on internal models of traffic flow and heuristic optimization algorithms, which, while innovative for their time, frequently incorporated simplifications that prevented them from accurately reflecting or adapting to real-time stochastic conditions.[1] The complex nature of traffic control, viewed as a sequential decision-making process where each action taken by the signal controller directly influences future traffic states, inherently defies straightforward computation of a globally optimal solution. The difficulty in constructing precise, real-time predictive models for traffic is formidable, primarily due to the stochastic nature of traffic patterns and highly variable driver behavior. This inherent modeling challenge directly paved the way for the adoption of Reinforcement Learning (RL) approaches. RL, being model-free, learns directly from interactions with the environment rather than relying on pre-defined or imperfect models, thereby offering a more robust and adaptable framework for dynamic traffic management.

1.3. The Role of Reinforcement Learning in Modern Traffic Systems

Reinforcement Learning (RL) algorithms have emerged as a powerful paradigm for addressing complex sequential decision-making problems, making them particularly well-suited for traffic signal optimization. These algorithms operate by iteratively learning optimal control strategies through continuous interaction with the environment and self-assessment of their performance, effectively discovering optimal policies through trial and error.[1] A significant advancement in this field is Deep Reinforcement Learning (DRL), which integrates deep neural networks with traditional RL. This combination has revolutionized the ability of RL systems to handle large and continuous state-action spaces, effectively overcoming the "curse of dimensionality" that traditionally limited tabular RL methods.[1] DRL's capacity to process raw, high-dimensional data, such as real-time image pixels from cameras, and automatically extract meaningful features and representations, allows for a more nuanced and comprehensive understanding of the traffic environment.[1] The transition from classical RL to DRL in traffic control represents more than just an algorithmic enhancement; it acts as a fundamental enabler for incorporating richer and more realistic state representations, including detailed vehicle attributes and visual data, which were previously computationally intractable. This deeper understanding of the traffic environment by the intelligent agent facilitates the development of more sophisticated and effective control strategies.

1.4. Scope and Objectives of This Review

This report undertakes a comprehensive analysis of state-of-the-art research in adaptive traffic signal control, specifically focusing on the application of Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL). The review is specifically tailored to systems that dynamically adjust traffic signal timings based on real-time traffic flow conditions and, critically, prioritize the passage of emergency vehicles. The primary objective is to synthesize the diverse methodologies employed in the reviewed literature, meticulously identify and elaborate on the key components of their RL models—namely, state representation, action space design, reward function engineering, and communication mechanisms. Furthermore, the report aims to critically evaluate the inherent limitations observed across these studies and, based on this evaluation, propose concrete and actionable improvements for developing a robust and highly effective traffic optimization model.

2. Foundational Concepts in Reinforcement Learning for Traffic Optimization

2.1. Markov Decision Processes (MDPs) and their Application

The theoretical bedrock for most reinforcement learning applications, including traffic signal optimization, is the Markov Decision Process (MDP). An MDP provides a formal mathematical

framework for modeling sequential decision-making in situations where outcomes are partly random and partly under the control of a decision-maker. It is formally defined by a five-tuple: a set of states (S), a set of possible actions (A), transition probabilities (T) between states given an action, a reward function (R) that provides immediate feedback, and a policy (π) that dictates action selection.[1] In the context of traffic control, the system's state, encompassing various traffic conditions, evolves over discrete time steps, with each decision made by the traffic signal controller directly influencing future traffic states. The overarching goal within an MDP is to discover an optimal policy that maximizes the expected cumulative reward over time, effectively guiding the system towards desired outcomes such as reduced congestion or improved throughput.[1] The MDP framework inherently captures the causal loop fundamental to adaptive traffic control: observing the current traffic state, executing a specific action (e.g., changing a light phase), receiving immediate feedback in the form of a reward (e.g., reduced delay), and subsequently transitioning to a new traffic state. This continuous feedback loop is indispensable for enabling the system to learn autonomously and adapt effectively within dynamic traffic environments. The success of RL in traffic control is largely attributable to this fundamental alignment, allowing the system to learn optimal behaviors through continuous trial and error in a complex, evolving environment.

2.2. Overview of Deep Reinforcement Learning (DRL) Paradigms

Deep Reinforcement Learning (DRL) encompasses a powerful set of techniques that combine the principles of reinforcement learning

with the representational power of deep neural networks. Within DRL, two primary paradigms are commonly employed:

Value-Based Methods (e.g., Deep Q-Networks - DQN): These methods focus on estimating the optimal action-value function, often denoted as the Q-function, which quantifies the expected cumulative reward of taking a specific action in a given state and then following an optimal policy thereafter. Deep Q-Networks (DQN) utilize deep neural networks as powerful function approximators for this Q-function, enabling them to effectively handle large and high-dimensional state spaces that would be intractable for traditional tabular RL methods.[1] To enhance the stability and performance of DQN, several advanced techniques are frequently integrated. These include:

- **Dueling DQN:** This architecture decomposes the Q-function into two separate estimators: one for the state-value function (V) and another for the advantage function (A). This separation allows the network to learn the value of states independently of the actions, which can improve learning efficiency and generalization.[1]
- **Double DQN:** This technique addresses the problem of overestimation of Q-values, a common issue in standard DQN, by decoupling the action selection from action evaluation. It uses the online Q-network to select the best action and the target Q-network to evaluate its value, leading to more accurate value estimates and improved stability.[1]
- **Prioritized Experience Replay:** Instead of uniformly sampling experiences from the replay buffer, this method prioritizes sampling experiences that have a higher temporal difference (TD) error. This means the agent focuses more on learning from "surprising" or "important" experiences, which can significantly

accelerate learning and lead to better final policies.[1]

Policy Gradient Methods (e.g., Deep Policy Gradient - DPG): In contrast to value-based methods, policy gradient approaches directly optimize the policy function itself. This function maps states directly to a probability distribution over actions, allowing the agent to learn the optimal actions without explicitly computing action-value estimates. Policy gradient methods are often preferred in environments with continuous action spaces or highly stochastic dynamics, where value-based methods might struggle.[1] However, a notable challenge with DPG methods is their susceptibility to high variance in gradient estimates, which can make training unstable and slow. To mitigate this, common techniques involve subtracting a baseline function, such as the state-value function, from the total reward, effectively reducing the variance of the gradient and promoting more stable learning.[1]

The selection between value-based (DQN) and policy-gradient (DPG) methods is a pivotal design decision, often involving a trade-off between training stability and sample efficiency (characteristic of DQN) versus the ability to handle continuous or highly stochastic action spaces (characteristic of DPG). For real-world traffic control, where safety and smooth transitions are paramount, the stability enhancements inherent in advanced DQN architectures are crucial considerations. Alternatively, if a DPG approach is chosen, significant effort must be dedicated to variance reduction techniques to ensure reliable and safe operation. The choice ultimately depends on the specific characteristics of the traffic environment, the available sensor data, and the computational resources.

3. Comprehensive Analysis of Reviewed Research Papers

3.1. Overview of Selected Literature

The collection of research papers provided offers a representative snapshot of the cutting-edge advancements in applying Deep Reinforcement Learning to traffic signal control. These studies collectively span a spectrum of challenges within intelligent traffic management, ranging from the optimization of individual intersections to the complexities of multi-agent systems coordinating across networks, and from general vehicular flow management to specialized multimodal considerations and critical emergency vehicle prioritization. Each paper contributes a unique perspective, offering distinct methodologies for defining the core components of an RL system: state representation, action space design, and reward engineering. By synthesizing these diverse approaches, a clearer understanding of the current capabilities and persistent challenges in the field emerges.

3.2. Tabular Synthesis of Key Research Papers on Adaptive Traffic Light Optimization

This section presents a detailed comparative analysis of the provided research papers, summarizing their core contributions, methodologies, and findings in a structured tabular format. This

table directly addresses the user's request for a comprehensive overview of each study's problem statement, relevance, dataset, approach, RL structure, neural network limitations, and suggestions for model improvement.

Table 1: Comparative Analysis of Reviewed Research Papers

Feature	[1]: Othman et al. (2025) - Multimodal Adaptive Traffic Signal Control	[1]: Liang et al. (2019) - A Deep Reinforcement Learning Network for Traffic Light Cycle Control	[1]: Garg et al. (2018) - Deep Reinforcement Learning for Autonomous Traffic Light Control	[1]: Coşkun et al. (2018) - Deep Reinforcement Learning for Traffic Light Optimization	[1]: Al-Heety et al. (2025) - Traffic Congestion Control with Emergency Awareness...
Problem Statement	Optimize traffic signals to simultaneously reduce total person delays for both general traffic and public	Inefficient traffic light cycle control causes long delays and energy waste. Goal is to dynamically adjust traffic	Congestion around road intersections due to traffic lights failing to adapt. Aim is to develop an adaptive, real-time	Optimize traffic light timings (phase and duration) to minimize overall vehicle travel time in a road	Increasing urban traffic congestion hinders mobility and emergency response. Current systems fail to dynamically prioritize emergency vehicles while maintaining optimal flow.

	transit, addressing the negative impact of traditional Transit Signal Priority (TSP) on surrounding traffic.	light duration based on real-time data to minimize delay, mimicking a human operator.	traffic optimization system using vision-based DRL to learn optimal policies.	network, addressing congestion from fast-growing populations.	Suboptimal communication infrastructure limits RL effectiveness.
Relevance to User's Research	Directly relevant for multimodal optimization (traffic + transit) and multi-agent coordination. Introduces "person-based" optimization, which can be extended to	Provides a robust DRL framework (3DQN) for single-intersection control, including advanced techniques (Dueling, Double Q, Prioritized Replay) for stability and performance. State	Focuses on vision-based DRL from raw pixels, offering an alternative state representation. Emphasizes learning and adaptation to various traffic conditions,	Proposes a novel reward function considering both traffic flow and delay, which is crucial for comprehensive optimization. Discusses stability issues (oscillations, catastrophic	Highly relevant as it directly integrates emergency awareness into Q-learning and optimizes communication infrastructure. Introduces specific emergency actions and multi-objective RSU placement, aligning perfectly with user's core requirements.

	emergency vehicle priority by weighting. Addresses conflict between prioritizing modes.	representation using grid-based position/speed is valuable.	including potential for emergency vehicle prioritization.	forgetting) in DRL for TLO.	
Dataset Used	Simulated model of five intersections in North York, Ontario, Canada, developed using Aimsun Next. Demand calibrated using 2016 Transportation Tomorrow Survey (TTS) data and City of	Simulated environment on Simulation of Urban Mobility (SUMO) simulator. Traffic parameters (vehicle arrival rates, speeds, accelerations) are defined for a 300x300 m intersection.	Custom-built traffic simulator on Unity3D (3D virtual reality software). Vehicle arrival rate conforms to random distribution.	Simulated dataset for the city of Bangalore, using SUMO (Simulation of Urban Mobility). Traffic demands generated with N=1000 steps and p=0.1 uniform probability distribution.	Simulated urban environment with multiple junctions. Vehicles generated using normal distribution for count and exponential distribution for time intervals; sensors subject to noise.

	Toronto turning movement counts.				
Where to Get the Dataset	Dataset is generated within the Aimsun Next simulation based on described calibration methods. No public link provided.	Dataset is generated dynamically within the SUMO simulation based on described traffic parameters. SUMO is open-source; replication requires SUMO installation and parameter configuration.	Dataset is generated by the custom Unity3D simulator. No public link or method to obtain this specific dataset is provided.	Dataset is generated within SUMO. No direct link to the Bangalore street simulation dataset. SUMO (open-source) is the primary resource for the simulator itself.	Dataset is generated within the simulation environment based on described probability density functions for vehicle generation and sensor noise. No public link or instructions provided.
Approach (along with EDA)	Decentralized multimodal	Deep Reinforcement Learning	Vision-based Deep Reinforcement	Proposes a new reward function	Integrated two-phase approach: Planning Phase

	Multi-Agent Reinforcement Learning (MARL) signal controller (eMARLIN-MM). Consists of encoder (observations to latent space) and executor (Q-network for decisions). Communication via shared embeddings between neighboring agents. Compares person-b	model (3DQN) to control traffic light cycle. Quantifies complex traffic scenarios as states by dividing intersection into small grids. Employs CNN to map states to rewards. Incorporates Dueling Network, Target Network, Double Q-learning, Prioritized Experience Replay.	Learning approach using a policy gradient algorithm. System learns from raw image pixels to detect inconsistent traffic flow and determine optimal traffic light policies. Aims to increase throughput and safety.	that simultaneously considers traffic flow and traffic delay. Uses both Deep Q-Learning (DQN) and Deep Policy Gradient (DPG) approaches to solve the RL problem. Evaluates performance in a micro-simulator.	(periodic RSU optimization using NSGA-II for coverage, cost, latency, reliability) and Operation Phase (continuous Q-learning for traffic control with emergency-aware module). "Avg Log" method for reward.

	ased vs. vehicle-based models.				
Steps / Process	1. Define MDP for multi-intersection system. 2. Design eMARLIN-MM: Encoder (raw observations to latent space), Executor (Q-network). 3. Agents disseminate embeddings to neighbors. 4. Executor uses concatenated embeddings for	1. Gather real-time traffic info (vehicular networks/ sensors). 2. Quantify states: divide intersection into grids, represent as	1. Initialize model parameters. 2. At each time-step: capture current visual (raw pixels) as state. 3. Select action (lane to green) based on policy. 4. Observe reward (+1 per vehicle passing) and next state. 5. Compute gradients and update policy	1. Formulate TLO as RL task. 2. Define state (binary presence vector). 3. Define action set (NSG, EWG, NWPG, EWPG) with safety transitions. 4. Propose two reward functions: cumulative delay, and delay + traffic flow (occupancy/halting	1. **Planning Phase (Periodic):** NSGA-II optimizes RSU placement (coverage, cost, latency, reliability). Communication infrastructure updated. 2. **Operation Phase (Continuous):** Observe state. Check for emergency. If emergency, select emergency action (CPE, RNO). Else, select action via ϵ-greedy Q-learning. Execute action, observe new state/reward. Update Q-table.

	Q-value estimation. 5. Train using DQN end-to-end. 6. Compare person-based vs. vehicle-based, independent vs. coordinated agents.		parameters (after batch of 10 simulations).	vehicles). 5. Train DQN and DPG using SUMO. 6. Evaluate average travel time and queue length.	
RL Structure	**State:** 7 components: cars/buses count/speed per lane, bus occupancy, current phase index, elapsed phase duration. **Action:** Extend	**State:** Grid-based representation (e.g., 60x60x2) of vehicle position (binary) and speed (m/s). **Action:** Update duration of one	**State:** Raw image pixels (100x100x3) from simulator. **Action:** 2 discrete actions {0, 1} (which lane to give green). **Reward:** +1 for	**State:** Vector (binary 1 for vehicle presence, 0 absence) from loop detectors. **Action:** NSG, EWG, NWPG, EWPG. Safety transition	**State:** Number of vehicles on each of 4 roads, classified as congested (C) or non-congested (N) (16 potential states). **Action:** Standard (N-S, E-W, E, W, N, S), Emergency (CPE, RNO). **Reward:** Log-based function balancing AWT/MWT std

	current phase, Switch to permitted subsequent phase (CVPS). **Reward:** Negative of total person delays (person-based) or total vehicle delays (vehicle-based). **Hyperparameters:** Initial ϵ=1, Final ϵ=0.01, Target network update freq=2k, Batch size=256, Buffer size=2M, Encoder/Executor layers/siz	phase by +/- 5 seconds. Max/Min duration (60s/0s). **Reward:** Cumulative waiting time difference between two cycles. **Hyperparameters:** Replay memory=20000, Minibatch=64, Starting ϵ=1, Ending ϵ=0.01 (10000 steps), Pre-train=2000, Target update rate=0.001, γ=0.99, Learning	every vehicle passing through the junction. **Hyperparameters:** Batch size=10 simulations, Episode length=100 time-steps. **Communication:** Real-time interface between Unity3D (client) and Python (server) using socket programming for bilateral data exchange.	s between phases. **Reward:** 1) Dt−1−Dt (cumulative delay). 2) Dt−1−Dt+NumberOfHaltingVehicles+cOccupancy (delay + flow). **Hyperparameters:** γ=0.995, α=0.001, Memory size=2000, Mini-batch=200. **Communication:** SUMO's TRACI (Traffic Control Interface) in Python.	dev and throughput (e.g., log(w·tanh(std(AWT))+(1−w)·tanh(τth))). **Hyperparameters:** α=0.01, γ=0.9, Initial ϵ=0.99 (decaying to 0.0001), Weighting Factor=0.6. **Communication:** V2V (IEEE 802.11p) and V2I via optimized Roadside Units (RSUs).

	es. **Communication:** Agents share fixed-length observation embeddings with immediate neighbors.	rate=0.0001. **Communication:** Information gathered from vehicular networks/sensors.			
Neural Network Limitations	Not explicitly discussed as a limitation, but the complexity of MARL models and the need for higher computational power for coordinated MARL are noted.	Implicitly addresses limitations of simpler NNs: inaccurate state representation (queue length only), fixed-time intervals, random signal	Criticizes existing simulators (SUMO) for unrealistic aspects: no collision count, constant speed, no dynamic vehicle generation, lack of variability, unrealisti	Instability and oscillation (catastrophic forgetting). Difficulty recovering from suboptimal actions. Convergence to suboptimal configurations (fixed/flic	Not explicitly discussed as neural network limitations, but the Q-learning approach uses a small state space (16 states), so deep neural networks might not be strictly necessary for the Q-table approximation.

		sequences, overfitting of single Q-networks, and limited information usage of ATSC.	c collision handling (teleporting).	kering lights, starving minor roads). Exponential growth of action space in real-world.	
Improvement Suggestions for User's Model	Use person-based models, especially with high bus occupancy, for maximizing person throughput. Use coordinated MARL (eMARLIN-MM) over independent agents for better performa	Use comprehensive state representation (position + speed grids). Implement dynamic phase duration control. Ensure smooth duration changes. Integrate advanced RL techniques	Extend to more complex intersections with multiple lanes. Implement real-time coordination between multiple intersections. Ensure robustness to stresses: camera degradation, single	Refine explore-exploit strategy. Incorporate constrained optimization (TRPO) to prevent starving minor roads. Enhance reward function with safety/comfort metrics (e.g.,	Conduct real-world pilot studies. Investigate scalability to very large networks (distributed computing). Develop robust methods for imperfect data. Collaborate with emergency services to refine handling protocols. Research fallback strategies for limited communication infrastructure.

	nce. Consider other modes (pedestrians) and transit metrics (headway regularity). Coordinate with GLOSA/DBLs/DAS.	(Dueling, Double Q, Prioritized Replay). Use robust optimization (Adam). Penalize illegal actions.	point of failure, unreliable communication.	penalize action flickering). Address action space complexity (clustering states). Consider multi-agent scenarios.	Adapt to diverse urban layouts. Seamless integration with legacy systems. Assess long-term stability. Explore environmental impacts, advanced predictive models, computational efficiency, security (blockchain, homomorphic encryption).

3.3. Thematic Deep Dive into State-of-the-Art Approaches

The comparative analysis reveals several overarching themes and critical design considerations in the application of DRL to traffic signal optimization. These themes represent key areas of advancement and ongoing research within the field.

3.3.1. Multi-Agent Reinforcement Learning (MARL) for Network Coordination

Traffic management in urban environments is inherently a distributed

problem, involving multiple interconnected intersections. Multi-Agent Reinforcement Learning (MARL) offers a compelling framework for addressing this complexity by allowing individual traffic signal controllers (agents) to learn and coordinate their actions across a network. The Othman et al. (2025) study [1] explicitly highlights the need for multimodal MARL ATSC models, noting that while single-agent RL has shown promise for individual intersections, it falls short in optimizing across multiple intersections. The paper introduces eMARLIN-MM, a decentralized MARL controller where each intersection communicates exclusively with its immediate preceding and succeeding neighbors by sharing "observation embeddings" rather than raw data. This approach is designed to facilitate coordination without significantly increasing computational complexity or communication bandwidth, as gradients are not transmitted between agents.[1]

The distinction between independent and coordinated agents is crucial. Independent agents (like iDQN-VB or iDQN-PB in [1]) optimize their own intersection without sharing information. However, the performance of an intersection is often influenced by its neighbors. Coordinated MARL models, by contrast, explicitly share information, leading to superior network-wide performance. For example, eMARLIN-MM-PB significantly reduced total person delays (51% to 66%) compared to pre-timed signals, outperforming independent DQN agents.[1] This performance gain underscores a critical point: MARL addresses network-level optimization, but effective coordination, such as through shared embeddings, is essential to overcome the non-stationarity introduced by evolving policies of neighboring agents and to achieve superior global optimality over independent agents. This implies a fundamental trade-off that researchers must consider between the computational overhead of

inter-agent communication and the benefits of global optimization.

3.3.2. Multimodal Traffic Optimization (Vehicular and Transit Priority)

Traditionally, traffic signal optimization has focused primarily on general vehicular flow, often overlooking or treating public transit as a secondary consideration. However, as highlighted by Othman et al. (2025) [1], transit vehicles, despite their high passenger occupancy, experience significant delays at traffic signals. While Transit Signal Priority (TSP) strategies exist, they often negatively impact surrounding traffic, creating a conflict between prioritizing cars versus transit. This necessitates a multimodal approach that optimizes for all road users. The eMARLIN-MM system proposes to simultaneously optimize total person delays for both traffic and transit by incorporating bus occupancy levels into the reward function.[1] This "person-based" approach assigns weights to different vehicle types based on their occupancy, inherently favoring vehicles with higher passenger loads, such as buses.

The study demonstrates that person-based models can substantially minimize total person delays (54% to 66% reduction) compared to vehicle-based models, especially when bus occupancy is high.[1] This shift in objective from pure "vehicle throughput" to "person throughput" fundamentally redefines what constitutes optimal traffic flow, aligning it more closely with societal benefit and sustainable mobility goals. The implication is profound: optimizing for "person delay" or "emergency awareness" (as seen in [1]) moves beyond a purely vehicle-centric view to one that prioritizes the movement of people, requiring more nuanced reward functions and potentially

complex state representations that capture passenger counts or emergency status.

3.3.3. Dedicated Emergency Vehicle Prioritization Mechanisms

A critical aspect of modern traffic management, and a core focus of the user's research, is the ability to prioritize emergency vehicles. Traditional TSP methods, while giving preference to transit, often increase delays for general traffic.[1] The Al-Heety et al. (2025) study [1] directly addresses this by proposing an "emergency-aware" adaptive Q-learning strategy. This system integrates an emergency-specific overriding process within the Q-learning framework, allowing it to dynamically recognize and prioritize emergency vehicles in real-time. This is achieved by adding specific emergency actions, such as "clear path for emergency" (CPE) and "resume normal operation" (RNO), to the agent's action set.[1]

The "Avg Log" method in [1] demonstrates superior long-term adaptability and effectively manages emergency situations, significantly reducing delays for emergency vehicles compared to traditional methods. This highlights that integrating emergency awareness requires not just a detection mechanism but also a dedicated overriding control logic and specific actions/rewards designed to balance emergency priority with minimal disruption to general traffic. This capability is a critical differentiator for traffic management systems, moving beyond simple priority to a more integrated, responsive, and safety-critical function.

3.3.4. Communication Infrastructure and its Optimization

The effectiveness of real-time adaptive traffic signal control systems, particularly those relying on DRL, is profoundly dependent on a robust and efficient communication infrastructure. Liang et al. (2019) [1] emphasize the role of vehicular networks and sensors in providing real-time traffic information as input to their DRL model. Garg et al. (2018) [1] utilize socket programming for real-time data exchange between their Unity3D simulator and Python-based policy network, underscoring the need for low-latency communication channels.

Al-Heety et al. (2025) [1] take this a step further by explicitly optimizing the communication infrastructure itself. Their approach employs a multi-objective optimization algorithm (NSGA-II) to strategically place Roadside Units (RSUs), considering critical factors such as coverage, cost, latency, and reliability. This "Planning Phase" periodically optimizes RSU deployment, which in turn enhances the communication infrastructure supporting the "Operation Phase" (the RL-based traffic control). The study highlights that poorly optimized RSU placement can lead to coverage gaps, increased latency, and reduced reliability, directly impacting the system's ability to respond to emergencies and maintain efficient traffic flow.[1] This comprehensive perspective indicates that robust and optimized communication infrastructure is a foundational requirement for real-time, adaptive intelligent transportation systems. Its quality directly influences data accuracy, transmission latency, and overall system reliability, extending beyond just the signal control logic to the underlying network that enables it.

4. Critical Evaluation of RL Model Components in Traffic Control

The design of a Deep Reinforcement Learning agent for traffic signal optimization necessitates careful consideration of its core components: state representation, action space, reward function, hyperparameters, and neural network architecture. Each choice carries implications for the model's performance, data requirements, and real-world applicability.

4.1. State Representation: Design Choices and Impact

The state representation defines how the RL agent perceives the environment, directly influencing its ability to learn an effective policy. The reviewed papers demonstrate a variety of approaches, each with its own trade-offs:

- **Detailed Vehicular Data:** Othman et al. (2025) [1] use a comprehensive state that includes the number of cars and buses in each lane, their speeds, bus occupancy, the current signal phase index, and its elapsed duration. This rich, multi-faceted representation provides the agent with granular information about the traffic flow and multimodal conditions.
- **Grid-Based Spatial Information:** Liang et al. (2019) [1] discretize the intersection into small square grids, representing the state as a matrix containing binary vehicle presence and their speeds. This visual-like representation, processed by Convolutional Neural Networks (CNNs), captures spatial relationships of vehicles.
- **Raw Image Pixels:** Garg et al. (2018) [1] take a more direct

approach, feeding raw image pixels from their simulator as the state input to a CNN. This method aims to allow the network to automatically discover relevant visual features without explicit feature engineering.

- **Discretized Occupancy/Presence:** Coşkun et al. (2018) [1] represent the state as a binary vector indicating the presence or absence of vehicles in discretized cells, often derived from loop detectors. Al-Heety et al. (2025) [1] simplify this further by classifying each road as either congested or non-congested based on vehicle count, resulting in a compact 16-state space for a four-road intersection.

A critical observation is that a richer state space, such as raw image pixels or detailed vehicle attributes, has the potential to lead to superior performance by providing the agent with a more complete understanding of the environment. However, this comes at the cost of increased computational complexity and higher data acquisition requirements. Conversely, simplified or discretized state representations, while reducing computational burden and relying on more readily available sensor data (e.g., loop detectors), might inadvertently lose crucial information, potentially limiting the agent's ability to learn optimal, nuanced policies. The choice of state representation is therefore a fundamental design decision that must balance the desired level of control granularity with the practical constraints of sensor availability and computational resources.

4.2. Action Space Design: Granularity, Safety, and Flexibility

The design of the action space dictates the decisions an RL agent

can make. Its granularity, safety considerations, and flexibility are paramount for effective and responsible traffic control:

- **Fixed-Step Duration Adjustments:** Liang et al. (2019) [1] define actions as changing the duration of a single phase by a fixed step (e.g., +/- 5 seconds) in the next cycle, within minimum and maximum limits. This provides fine-grained control over phase durations.
- **Discrete Phase Transitions:** Othman et al. (2025) [1] allow agents to either "Extend" the current phase or "Switch" to one of the permitted subsequent phases, adhering to Constrained Variable Phasing Schemes (CVPS) for safety and ensuring all major movements are served. Coşkun etun al. (2018) [1] define specific phase actions (e.g., North-to-South Green) and explicitly incorporate "safety transitions" between configurations to prevent accidents.
- **Simplified Binary Actions:** Garg et al. (2018) [1] use a very simplified action space of only two discrete actions, indicating which of two lanes should receive a green light. While easy to train, this limits the system's flexibility in complex intersections.
- **Emergency-Specific Actions:** Al-Heety et al. (2025) [1] introduce specialized actions like "clear path for emergency" (CPE) and "resume normal operation" (RNO) in addition to standard phase controls. This direct integration of emergency responses within the action space is crucial for the user's research.

The granularity of actions (e.g., fixed-step duration adjustments versus discrete phase switches) directly impacts the system's flexibility and responsiveness. More granular control allows for finer optimization but can increase the complexity of the action space. The inclusion of explicit "emergency actions" [1] is a direct response to the user's specific problem, enabling the model to directly intervene

for emergency prioritization. Furthermore, safety constraints, such as ensuring proper yellow light intervals and permissible phase transitions [1], are not merely technical details but critical requirements for real-world deployment, preventing hazardous traffic conditions.

4.3. Reward Function Engineering: Balancing Competing Objectives

The reward function is arguably the most critical component in RL, as it defines the objective the agent seeks to maximize. Its careful design is essential for guiding the agent towards desired behaviors and preventing unintended consequences. The reviewed papers highlight the complexity of this design:

- **Delay Minimization:** A common objective is to minimize vehicle waiting time or delay. Liang et al. (2019) [1] define reward as the change in cumulative waiting time, aiming to reduce it. Coşkun et al. (2018) [1] use cumulative delay difference as one reward function.
- **Throughput Maximization:** Garg et al. (2018) [1] use a simple reward of +1 for every vehicle passing through the junction, directly incentivizing throughput.
- **Person-Based Optimization:** Othman etman et al. (2025) [1] introduce a "person-based" reward function that minimizes total person delays, accounting for vehicle occupancy (e.g., buses carrying more people). This shifts the focus from vehicle flow to human mobility.
- **Multi-Component Rewards:** Coşkun et al. (2018) [1] propose a novel reward function that combines cumulative delay with traffic flow (occupancy and number of halting vehicles), aiming to encourage flowing traffic and reduce queues. Al-Heety et al.

(2025) [1] develop a log-based reward function that balances the standard deviation of average/maximum waiting times with throughput, emphasizing a holistic approach.

The design of the reward function is inherently complex and critical. Different objectives, such as maximizing vehicle throughput, minimizing person delay, or prioritizing emergency vehicles, necessitate distinct reward structures. A significant challenge arises when simple reward functions, particularly those solely focused on delay or throughput, can lead to suboptimal or unsafe behaviors, such as perpetually starving minor roads of green time.[1] This can create an unfair distribution of delays across the network. More sophisticated, multi-objective reward functions, as seen in [1], are designed to address these trade-offs but are inherently harder to tune effectively to achieve the desired balance.

4.4. Hyperparameter Tuning and Training Stability Considerations

Hyperparameters are external configurations that are set before the training process begins, significantly influencing the learning dynamics and final performance of an RL model. The reviewed papers provide insights into typical hyperparameter choices and the challenges of training stability:

- **Learning Rate (α):** Typically small (e.g., 0.001 in [1], 0.01 in [1]), controlling the step size of parameter updates.
- **Discount Factor (γ):** Close to 1 (e.g., 0.99 in [1], 0.995 in [1], 0.9 in [1]), balancing immediate versus future rewards.
- **Exploration Rate (ϵ):** Often starts high (e.g., 1.0 in [1], 0.99 in [1]) and decays over time to a minimum (e.g., 0.01 in [1], 0.0001 in [1]),

balancing exploration of new actions with exploitation of known good actions.
- **Experience Replay Buffer Size and Batch Size:** Large buffer sizes (e.g., 2M in [1], 20000 in [1], 2000 in [1]) and appropriate batch sizes (e.g., 256 in [1], 64 in [1], 200 in [1]) are crucial for stabilizing training by breaking correlations in sequential observations.
- **Target Network Update Frequency:** Used in DQN variants (e.g., 2k in [1]), this parameter controls how often the target Q-network is updated from the online Q-network, further contributing to training stability.

A recurring challenge highlighted in the literature is the issue of training stability, often manifested as oscillations in performance or "catastrophic forgetting" where the network loses previously learned knowledge while acquiring new information.[1] These stability issues are common in DRL applications, especially in dynamic environments like traffic. Techniques such as the use of target networks, experience replay, and dueling architectures [1] are not merely performance enhancements but are critical mechanisms for mitigating these instabilities and ensuring reliable learning. Proper hyperparameter tuning, therefore, becomes an iterative and often empirical process essential for achieving robust and effective traffic control policies.

4.5. Neural Network Architectures and their Performance Characteristics

Deep neural networks serve as the function approximators in DRL, enabling the handling of complex state and action spaces. Various architectures are employed, each tailored to the specific problem

characteristics:

- **Convolutional Neural Networks (CNNs):** Widely used when state representations involve spatial data, such as grid-based vehicle positions and speeds [1] or raw image pixels.[1] CNNs excel at extracting hierarchical features from such inputs. Liang et al. (2019) [1] use a CNN with multiple convolutional and fully-connected layers.
- **Encoder-Executor Architecture:** Othman et al. (2025) [1] propose an eMARLIN-MM framework with an encoder module that transforms raw observations into a lower-dimensional "observation embedding," and an executor module (Q-network) that makes timing decisions based on its own embedding concatenated with those from neighboring agents. This design aims to reduce communication bandwidth and computational load.
- **Dueling, Double Q-learning, and Prioritized Experience Replay Integration:** Liang et al. (2019) [1] demonstrate that combining these advanced DQN techniques within a single framework (3DQN) significantly improves performance by reducing overestimation and accelerating learning.
- **Policy Gradient Networks:** Garg et al. (2018) [1] and Coşkun et al. (2018) [1] explore Policy Gradient methods, where the neural network directly outputs action probabilities. While potentially better for continuous action spaces or highly stochastic environments, DPG can suffer from high variance in gradient estimates.[1]

The choice and configuration of the neural network architecture are pivotal. CNNs are particularly effective for processing spatial state representations. Furthermore, the integration of multiple DRL enhancements within a single architecture, as exemplified by the

3DQN in [1], has been shown to yield substantial performance improvements and enhance training stability. The fundamental distinction between value-based methods (like DQN) and policy-gradient methods (like DPG) presents different strengths and weaknesses, particularly concerning training stability and the handling of variance. For a robust traffic control system, selecting an architecture that can effectively manage the complexity of the state space while ensuring stable and reliable learning is paramount.

5. Strategic Recommendations for Enhancing Your Traffic Optimization Model

Based on the comprehensive analysis of the reviewed literature, several key limitations and challenges emerge, which, when addressed, present significant opportunities for enhancing your traffic optimization model.

5.1. Identified Limitations and Challenges Across Existing Studies

The current body of research, while advanced, faces several common hurdles that impede full real-world applicability and optimal performance:

- **Simulation Constraints and Real-World Gap:** Most studies rely heavily on simulated environments (e.g., SUMO, Aimsun Next, Unity3D).[1] While valuable for initial development, these simulations, despite efforts to mimic realism, may not fully capture the unpredictable nuances and complexities of actual

urban traffic conditions, such as diverse driver behaviors, unexpected incidents, or sensor failures.[1]

- **Scalability Challenges:** While some multi-intersection scenarios are explored [1], scaling DRL solutions to very large, complex urban networks remains a significant challenge. The computational requirements can grow exponentially with the number of intersections and agents, potentially affecting real-time performance and deployment feasibility.[1]
- **Data Dependence and Imperfections:** The effectiveness of DRL systems is highly contingent on the availability and accuracy of real-time traffic data.[1] In real-world deployments, data gaps, latency, noise, or inaccuracies from sensors can severely impact system performance and decision-making.[1]
- **Completeness of Emergency Handling:** While emergency vehicle prioritization is a stated goal for some [1], the models may not account for all possible emergency scenarios, the specific protocols of local emergency services, or the optimal balance between emergency priority and minimizing disruption to general traffic flow.[1]
- **Communication Infrastructure Gaps:** The full potential of V2V and V2I communication, crucial for real-time data exchange and coordination, is often assumed or not fully optimized.[1] The lack of robust and universally available communication networks can limit the practical deployment of these systems.[1]
- **Adaptation to Diverse Urban Layouts:** Models are often tested on specific intersection geometries or simplified grid networks.[1] Their generalizability to cities with unique, irregular, or historically complex road layouts requires further investigation and adaptation.[1]
- **Integration Challenges:** Seamless integration with existing,

often legacy, traffic management systems and infrastructure presents significant practical hurdles for widespread adoption.[1]

- **Long-term Stability and Learning Curve:** DRL algorithms can exhibit instability during training (e.g., oscillations, catastrophic forgetting).[1] Ensuring long-term stability and sustained optimal performance in continuously changing urban environments requires extensive study and robust learning mechanisms.[1]
- **Ethical Considerations:** While not extensively detailed in all snippets, the ethical implications of autonomous traffic control, particularly concerning fairness in resource allocation (e.g., potential for starving minor roads) and data privacy, are critical for public acceptance and responsible deployment.[1]

5.2. Advanced Strategies for Model Improvement

To develop a traffic optimization model that surpasses existing approaches and addresses the identified limitations, the following advanced strategies are recommended:

5.2.1. Refined State and Action Space Design for Comprehensive Scenarios

The efficacy of your model will heavily depend on its ability to accurately perceive and interact with the complex traffic environment.

- **Hybrid State Representation:** Combine the granularity of detailed vehicle data (counts, speeds, and lane-specific information for cars and buses, including occupancy [1]) with

higher-level, aggregated congestion indicators (e.g., road segment density, queue lengths [1]) and signal phase status.[1] This multi-scale approach can provide both fine-grained control and a broader understanding of network conditions. Consider incorporating historical traffic patterns or predictive elements (e.g., using LSTM or Transformer methods as suggested in [1]) into the state to enable the agent to anticipate future traffic demands.

- **Dynamic and Safe Action Space:** Adopt a flexible action space that allows for both extending current phases and switching to permitted subsequent phases, adhering to constrained variable phasing schemes (CVPS) to ensure safety and serve all major movements.[1] Crucially, integrate explicit emergency-specific actions, such as "clear path for emergency" (CPE) and "resume normal operation" (RNO).[1] These should be part of an overriding control mechanism that takes precedence when emergency vehicles are detected, ensuring rapid and safe prioritization while minimizing disruption to general traffic flow. The action space should also consider the "yellow signal" duration for safety transitions.[1]

5.2.2. Multi-Objective Reward Function Design for Traffic Flow and Emergency Response

The reward function must intricately balance competing objectives to achieve holistic optimization.

- **Weighted Multi-Component Reward:** Design a reward function that explicitly combines multiple objectives, moving beyond simple delay or throughput. This could involve a

weighted sum of:
    - **Negative total person delay:** Prioritizing the movement of people, not just vehicles, by incorporating vehicle occupancy (especially for transit and emergency vehicles).[1]
    - **Throughput maximization:** Incentivizing efficient flow of vehicles through intersections.[1]
    - **Queue length minimization:** Directly addressing congestion at approaches.[1]
    - **Emergency vehicle delay minimization:** A high-priority component that provides significant negative reward for delays to emergency vehicles.[1]
    - **Penalties for undesirable behaviors:** Include terms that penalize "starving" minor roads [1], excessive phase switching, or actions that lead to unsafe driving conditions (e.g., sudden braking, "action flickering" [1]).
- **Adaptive Weighting:** Explore adaptive weighting schemes for different components of the reward function, allowing the system to dynamically adjust priorities based on real-time conditions (e.g., increasing the weight of emergency vehicle delay when an emergency is detected). The log-based reward functions in [1] offer a sophisticated approach to balancing different metrics.

5.2.3. Integration of Robust DRL Techniques for Stability and Performance

To ensure reliable and high-performing learning, leverage advanced DRL architectures and training methodologies.

- **Enhanced DQN Architecture:** Implement a Deep Q-Network (DQN) architecture that incorporates multiple

stability-enhancing techniques. This includes:
    - **Dueling Network Architecture:** To improve the estimation of Q-values by separating state value and action advantages.[1]
    - **Double DQN:** To mitigate the overestimation bias inherent in standard DQN, leading to more accurate value estimates.[1]
    - **Prioritized Experience Replay:** To accelerate learning by focusing on more "important" experiences (those with higher temporal difference errors).[1]
- **Multi-Agent Coordination:** For multi-intersection scenarios, adopt a coordinated Multi-Agent Reinforcement Learning (MARL) approach, similar to eMARLIN-MM.[1] This involves agents sharing learned information (e.g., observation embeddings) with neighboring intersections to achieve network-wide optimization, which consistently outperforms independent agents.[1]
- **Robust Optimization:** Utilize adaptive optimization algorithms like Adam, known for their fast convergence and adaptive learning rates, for training the neural networks.[1]

5.2.4. Addressing Real-World Data Imperfections and Communication Gaps

Real-world deployment necessitates robustness against imperfect data and communication challenges.

- **Sensor Data Fusion and Noise Handling:** Integrate data from multiple sensor types (e.g., loop detectors, cameras, V2I/V2V communication) [1] and implement techniques to handle sensor noise and inaccuracies.[1] This could involve Kalman filters or other state estimation techniques.
- **Optimized Communication Infrastructure:** Explicitly consider

and, if possible, optimize the underlying communication infrastructure, such as the placement and configuration of Roadside Units (RSUs).[1] This ensures low latency, high reliability, and adequate coverage for real-time data exchange between vehicles, infrastructure, and the control system. Research fallback strategies for situations with limited or unreliable communication.[1]

- **Historical Data Integration:** Incorporate historical traffic data and demand patterns (e.g., time of day, day of week) into the model's inputs. This can help the agent generalize better to dynamic traffic conditions and predict future scenarios, addressing limitations where current models rely solely on real-time observations.[1]

5.2.5. Scalability and Generalizability for Diverse Urban Environments

For practical utility, the model must be scalable and adaptable to various urban contexts.

- **Modular and Distributed Design:** Design the system with a modular and distributed architecture to facilitate scalability to large urban networks. This might involve hierarchical control or clustering techniques to manage computational complexity.[1]
- **Transfer Learning and Domain Adaptation:** Explore transfer learning techniques, where a model trained on one simulated or real-world environment can be adapted to new, unseen environments with minimal retraining. This enhances generalizability to diverse urban layouts and traffic conditions.[1]
- **Simulation-to-Reality (Sim2Real) Gap:** Plan for real-world pilot studies to validate the model's performance under actual

urban conditions, addressing the inherent gap between simulation and reality.[1] This iterative deployment framework, with continuous monitoring and improvement, is crucial for refining the system's effectiveness.

5.2.6. Incorporating Other Transportation Modes (e.g., Pedestrians, Cyclists)

To achieve truly comprehensive urban mobility optimization, expand the scope beyond just vehicular and transit traffic.

- **Pedestrian Integration:** Incorporate pedestrian demand and waiting times into the state and reward functions. Pedestrians are significantly impacted by traffic signals, especially in dense urban areas, yet are often overlooked in optimization problems.[1] This would require additional sensing capabilities (e.g., pedestrian detection sensors).
- **Cyclist Considerations:** Similarly, integrate cyclist presence and flow into the optimization process, particularly in cities with high cycling modal share.

6. Conclusion and Future Research Trajectories

The increasing complexity of urban traffic necessitates a paradigm shift from traditional, static signal control to intelligent, adaptive systems capable of real-time optimization and emergency response. This report has demonstrated that Deep Reinforcement Learning (DRL) offers a robust and promising framework for achieving this. The analysis of current research highlights significant advancements

in multi-agent coordination, multimodal optimization, and dedicated emergency vehicle prioritization.

Key findings indicate that sophisticated state representations, dynamic action spaces incorporating safety transitions, and multi-objective reward functions are critical for effective DRL-based traffic control. The transition from vehicle-centric to person-centric optimization, particularly with high-occupancy transit vehicles, yields substantial benefits in overall person delay reduction. Furthermore, the explicit integration of emergency-aware modules and the optimization of underlying communication infrastructure are paramount for ensuring both general traffic efficiency and critical emergency response capabilities.

Despite these advancements, challenges persist in areas such as scalability to very large networks, robustness against real-world data imperfections, and seamless integration with existing infrastructure. These limitations, however, present fertile ground for future research.

For your research paper, the path forward involves synthesizing the most effective strategies identified:

1. **Develop a hybrid state representation** that combines granular vehicular data with aggregated congestion levels and historical patterns.
2. **Design a dynamic action space** that allows flexible phase adjustments, incorporates safety transitions, and includes an overriding mechanism for emergency-specific actions.
3. **Engineer a multi-objective reward function** that explicitly balances person-based delay minimization, throughput maximization, queue length reduction, and critical emergency

vehicle prioritization, while penalizing undesirable behaviors.

4. **Implement a robust DRL architecture** by integrating advanced techniques such as Dueling DQN, Double DQN, and Prioritized Experience Replay, especially within a coordinated multi-agent framework for network-wide optimization.
5. **Address practical deployment challenges** by focusing on methods for handling imperfect sensor data, optimizing communication infrastructure, and ensuring the model's generalizability and long-term stability across diverse urban environments.
6. **Expand the scope** to include other vulnerable road users like pedestrians and cyclists, moving towards truly holistic urban mobility solutions.

By meticulously addressing these areas, your research can contribute significantly to the development of highly efficient, safe, and responsive intelligent traffic management systems, ultimately enhancing urban mobility and quality of life.

Works cited

1. Traffic_Congestion_Control_with_Emergency_Awareness_and_Optimized_Communication_Infrastructure_using_Reinforcement_Learning_and_Non-Dominated_Sorting_Genetic_Algorithm.pdf