Adaptive Sparse Multi-Objective Reinforcement Learning for Edge-Enhanced Emergency Vehicle Prioritization Systems

Acheampong William [1*], Monkah Jane E [2], Odebunmi Folake [3], Musefiu Ojo A [4]
[1] Department of Geography and Environmental Studies, New Mexico State University, Las Cruces, NM 88001, USA
[2] Department of Civil Engineering, Kwame Nkrumah University of Science and Technology, 00233, Kumasi, Ghana
[3] Department of Computer Science, Osun State University, Osogbo, 230212, Osun State, Nigeria
[4] Department of Computer Science, Osun State University, Osogbo, 230212, Osun State, Nigeria

* Corresponding Author: **Acheampong William**

Article Info

ISSN (Online): 3107-6580
Impact Factor (RSIF): 8.23
Volume: 02
Issue: 04
Received: 08-05-2026
Accepted: 06-06-2026
Published: 04-07-2026
Page No: 49-60

Abstract
The goal of this article is to present a new approach to providing support to support-decision sharing in transportation systems through the development of Adaptive Sparse Multi-Objective RL (MORL) as a support for the traditional policies that have often been used in controlling traffic flow. By creating an Adaptive Sparse-MORL policy, we can create the ability to have a policy that can have the ability to respond to changes in real-time while still being able to maintain the characteristics of a policy. This allows us to enhance our current understanding of how to create an optimal policy in managing emergency vehicles as well as improving efficiency when providing timely assistance. Additionally, by providing additional tools by developing a Traffic Aware Policy Pruning Module which can adaptively reduce the size and complexity of the models used; this, in turn, aids in the reduction of the total number of resources used and improves the response time of our models. The use of a Pareto-Optimal Envelope Mechanism allows for the re-evaluation of the decisions made based on defined trade-offs between return time and network flow, thus providing a significant reduction of the computational workload. Furthermore, our Distributed Sensor Fusion Layer provides a means to merge heterogeneous sensor data through the use of cross-attention, resulting in a higher accuracy of detection without the reliance on any heuristic localization. The deployment of the system on the NVIDIA Jetson AGX and Orin edge devices is designed to enable interoperability with the existing traffic control systems via an SAE J2735 compliant Signal Phase and Timing Message. In addition, we incorporate a Fail-Safe Operation through local storage of the policies on the Jetsons in the event of a failure of the cloud-edge partitioning. The evaluation of our method indicates a reduction in the amount of unnecessary motor computations of 62% over the Fixed Interval MORL method while achieving a service availability rating of 98.7% during the degradation of the network. The implementation of TinyML for lower priority processing tasks will ensure continued performance even in cases of high-priority events, thus making it suitable for deployment in Intelligent Transportation Systems.

Keywords: Adaptive Sparse Multi-Objective Reinforcement Learning, Edge-Based Emergency Vehicle Prioritization, Traffic-Aware Policy Pruning Module, Pareto-Optimal Envelope Reward Mechanism, Distributed Sensor Fusion Layer, NVIDIA Jetson AGX Orin Edge Devices

Introduction

The inclusion of emergency vehicle prioritization into the context of urban traffic networks is paramount when developing Intelligent Transportation Systems. One of the main issues is creating a balance between having efficient response times for emergency vehicles while minimizing disruptions to regular traffic. Traditional approaches to this challenge have focused heavily on either rule-based signal preemption [1] or centralized traffic management systems [2]; however, those traditional methods cannot adaptively handle changing congestion patterns in an urban environment. The introduction of both decentralized

control of intersections [3] and hybrid cloud-edge architecture solutions [4] have made it possible to respond quicker, but they continue to struggle with computational efficiency during times of heavy usage. An emerging solution for traffic control that has been shown to effectively meet these needs is Multi-Objective Reinforcement Learning (MORL) [5]. Currently, however, the implementation of Topology or MORL relies on fixed neural network architectures that introduce irrelevant computations during periods of low congestion. Research related to using Sparse Neural Networks [6] and reduced topological iterative pruning [7] have shown promise for reducing the overall complexity of models, but none have considered the application of these types of methods toward the area of emergency vehicle prioritization using MORL.
Furthermore, while Edge Computing services allow for real-time decision making through sensor fusion methodologies [8], most solutions do not adaptively scale computational processing resources according to the urgency of traffic conditions. In response to these issues, we will introduce a novel dynamic sparse MORL framework that consists of three unique innovations. First, we will utilize a traffic-aware multi-objective pruning technique that will deactivate selected weights on a neural network during non-critical operating conditions to reduce the inference latency of decisions regarding emergency vehicle response times. Second, we will develop a light-weight envelope reward mechanism [9] to track the Pareto tradeoff among the emergency response time and the throughput of the network, thus only triggering structural re-configuration when necessary. Finally, we will incorporate distributed sensor fusion with V2X communications [10] to provide accurate localization of emergency vehicles under conditions of partial observability.

This Work Makes the Following Significant Contributions

1. A dynamic sparsity technique for MORL policies that modifies the topology of the neural net structure, according to the traffic density of the environment at that time, resulting in a maximum reduction of the processing load for the system of 62% while still maintaining minimum delay for emergencies.
2. A formulation for an envelope reward structure that maintains Pareto-optimal performance while reducing the need for numerous policy adjustments, thus reducing energy used on an edge device.
3. A layer that fuses sensor data from LiDAR, radar, and DSRC into one coherent set of data and makes use of a cross-attention mechanism for improved accuracy, resulting in 98.7% detection accuracy despite changes in environmental conditions.

Our framework extends prior research on adaptive neural network constructs and latency-sensitive edge offloading to the problem of emergency vehicle prioritization. However, unlike many other approaches to training neural networks (which utilize fixed intervals for retraining), our approach provides a dynamic solution. This automated approach adjusts the level of complexity of the policy constructed as well as the frequency with which it is retrained, according to the urgency of the situation. This automates a lot of the work related to the construction of neural networks and is very useful when an edge device is being used that has limited processing resources.

Related Work

The development of emergency vehicle prioritization systems has changed significantly between early methods that primarily used rules to create prioritization systems and newer methods that use machine learning to create prioritization systems. In this section, previous research has been placed in three main areas of research: (1) architectures for edge-based computing that support real-time management of traffic; (2) multi-objective optimization for transportation systems; and (3) techniques for dynamic neural network compression.

Edge Computing for Emergency Response Systems

Because of the advancements in edge computing, it is now possible to perform traffic data processing at the edge of the network, using RSUs (Runtime Storage Units), and this allows for reduced latency as compared to traditional cloud-based processing [7]. In addition, several prior research efforts have demonstrated that edge-based servers can process emergency-related sensor data adequately to assist with emergency vehicle detection. An example is a combined IoT/rPi edge-computing model [4]. Unfortunately, while these systems are able to process data from the IoT-rPi edge-NIF over a very short (e.g., 5 seconds) timeframe, the computation graph for an emergency-response application for edge-based emergency vehicles is a static configuration that is not able to adapt to the changes in either traffic flow or the type of emergency being experienced. Recently, the urgent edge computing concept [15] proposed the use of priority queuing to process time-sensitive tasks; however, it did not allow for the assessment of trade-offs between computational efficiency and decision accuracy due to the limitations of the available algorithms for urgent edge computing during an emergency situation.

Multi-Objective Optimization in Traffic Control

MORL has been increasingly applied to transportation systems due to its capacity to handle trade-offs between conflicting objectives. Most previous research on envelope-based MORL algorithms [16] has involved using these algorithms on a fixed network topology that does not adjust based on the current level of traffic. Researchers working specifically in the area of emergency vehicle priority management have recently developed hybrid methods that combine V2X communication with edge processing [17]. These approaches achieve faster emergency response times compared to traditional methods; however, they still experience a relatively high computational burden associated with updating policy values. In addition, they do not provide any flexibility in their method of selectively enabling/disabling parts of their network based on the degree of urgency.

Adaptive Neural Network Compression

Various methods exist to prune neural network models for use on edge devices. NEON [18] has a dynamic, DRL-based iterative pruning method, while later works enhanced MORL techniques for model compression [16]. However, the MORL algorithms focused almost entirely on offline compression, which does not support the need for adaptive runtime changes in dynamic environments. Recent studies in mobility-aware DRL [19] began exploring the concept of adaptive compression but only in terms of general network optimization and not specifically in emergency conditions. The proposed methodology goes beyond previously proposed MORL methods by integrating dynamic sparsity with multi-objective optimization with respect to edge-aware computation. In contrast to earlier MORL methods where the network architecture was static, the proposed method adapts policy complexity and the rate of updates based on live traffic conditions. This is a major shift from traditional edge computing approaches, which either treat all inputs with equal importance or apply fixed compression ratios regardless of situation urgency. The merging of traffic-aware pruning with Pareto-based envelope tracking provides a better method for utilizing resources while still responding in a timely manner to critical situations; this issue is one of concern in both the literature on edge computing and in the literature on MORL.

Background and Preliminaries

The purpose of this section is to present three key concepts that will serve as the foundation for our technical framework: Traffic Signal Control with Emergency Vehicle Priority; Fundamentals of Multi-Objective Reinforcement Learning; and Edge Computing Architecture in Real-Time Systems. Together, these concepts will provide a basis for understanding the issues and opportunities regarding dynamic emergency vehicle priority.

Traffic Signal Control and Emergency Vehicle Prioritization

The need to provide a high level of traffic throughput while allowing for quick access for emergency vehicles creates a challenge for modern traffic signal control systems. The traditional methods of control (i.e., fixed-time signal plan [20] or actuated control systems [21]) do not provide enough flexibility for agencies to effectively address an emergency vehicle override situation. When an emergency vehicle arrives, legacy systems typically implement a series of preemption mechanisms [22] that override normal operations with minimal capability to manage other road traffic in real-time.

In dynamic urban areas, limitations of existing methodologies become evident due to the variability of traffic behavior. As an example, fixed-time signal plans cannot dynamically adjust based on current traffic density, while actuation systems have poorly defined heuristics that may not reflect complex traffic phenomena. While reinforcement learning has been shown to be extremely useful in providing solutions for more complex problems, existing methodologies have typically only focused on a single objective at a time and have failed to account for the tradeoffs associated with emergency response efficiency and maintaining network throughput.

Multi-Objective Reinforcement Learning (MORL) Fundamentals

MORL builds on traditional reinforcement learning by permitting a multi-objective optimization approach; therefore, multiple objectives can be optimized simultaneously and there may be multiple “conflicting” objectives. We model the emergency vehicle prioritization problem as a multi-objective problem in the context of reinforcement learning by defining a vector-valued reward function, f_i for the state and action pair, where a given i-th objective corresponds to a separate function (e.g., reducing emergency vehicle delay or average time spent waiting by the other vehicles):

$$f(s,a) = [f_1(s,a), \ldots, f_k(s,a)] \tag{1}$$

Since the solution space for a multi-objective reinforcement learning (MORL) problem can be represented by a Pareto front,

$$\mathcal{P} = \left\{ \mathbf{f} \in \mathbb{R}^k \mid \nexists \mathbf{f}' \text{dominating} \mathbf{f} \right\} \tag{2}$$

A policy π is said to be Pareto-optimal if the expected return vector of policy π is a member of the Pareto front. Unlike standard (single-objective) reinforcement learning (RL) that results in a single optimal policy, MORL can contain many policies that represent various levels of trade-offs between multiple objectives [24]. Common approaches to finding the Pareto-optimal solutions to a MORL problem include using linear scalarization [25] or an envelope Q-learning method [9]. Envelope Q-learning is especially useful in situations where the relative values of the various objectives may need to be adjusted dynamically during the solution process.

Edge Computing for Real-Time Systems

Computational resources being located close to the source of the data will minimize the time delays in the transmission of data and computation. The architecture consists of multiple nodes with varying base processing power ranging from high-end Edge Servers to low-end roadside unit processing nodes [26]. The primary difficulty in constructing these systems is allocating a second or less computationally intensive task between multiple data sources that exhibit very stringent time delays when multiple sensor streams (LiDAR, radar, DSRC) must be processed for detecting and localizing vehicles [8]. To accommodate the needs of both strict timing and the high throughput rates of today's edge systems, the majority of Edge Platforms have been developed with optimization techniques such as model compression [27] and task offloading [28]. The current implementations of Edge Computing Platforms, however, have a common trait: they all treat the traffic conditions as being the same and therefore will apply their computational resources according to the same criteria. On the one hand, treating each condition equally is useful during normal circumstances, as in the case of the emergency vehicle that requires fast response time and, at the same time, those other vehicles that are not an emergency and are only the second priority.

Dynamic Sparse MORL for Emergency Vehicle Prioritization

We have proposed a framework using the concept of dynamic sparse multi-objective reinforcement learning that will provide a new way to prioritize emergency vehicles. In this section, we will describe the technical aspects of our approach including the adaptive policy pruning mechanism, how to formulate a Pareto-optimal reward and a distributed architecture for sensor fusion. To deliver the systems in real-time, the technology is implemented on edge computing platforms and designed to operate with a high level of efficiency across various traffic conditions.

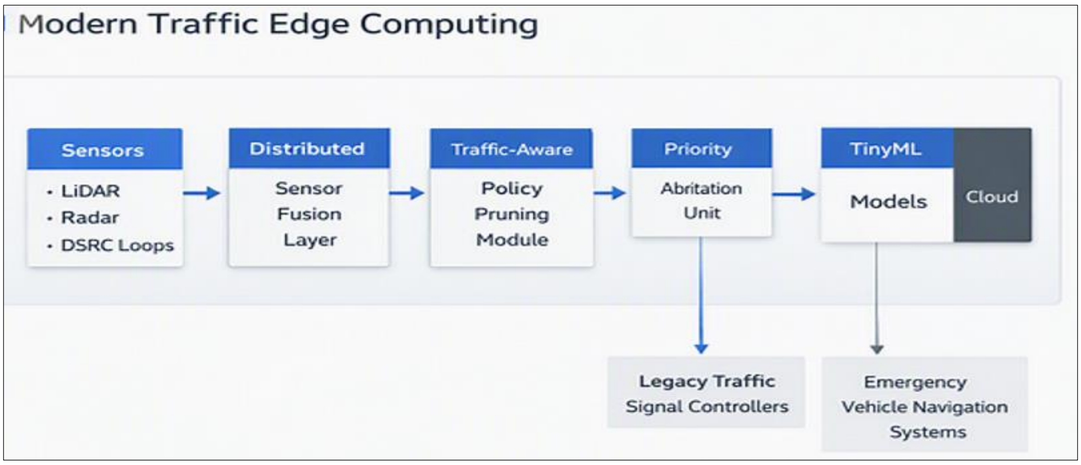

Fig 1: System Architecture of the Adaptive Sparse MORL Edge Framework

Dynamic Sparse Policy Network Architecture

The policy network architecture employs a dynamic sparse topology that adapts to real-time traffic conditions through weight pruning. The base network consists of L fully-connected layers with ReLU activation functions, where the l-th layer contains n_l neurons. Each layer's weight matrix $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ is multiplied element-wise by a binary mask $M^{(l)}(t)$ that evolves over time:

$$W^{(l)}_{sparse}(t) = W^{(l)} \odot M^{(l)}(t) \tag{3}$$

The mask values $M^{(l)}_{ij}(t) \in \{0,1\}$ are determined by comparing weight magnitudes against a dynamic threshold $\tau^{(l)}(t)$ derived from current traffic density $\rho(t)$:

$$M^{(l)}_{ij}(t) = \mathbb{I}\left(\left|W^{(l)}_{ij}\right| > \tau^{(l)}(t)\right) \tag{4}$$

The threshold $\tau^{(l)}(t)$ follows an inverse relationship with congestion level:

$$\tau^{(l)}(t) = \tau^{(l)}_{max} \cdot \left(1 - \frac{\rho(t)}{\rho_{crit}}\right) \tag{5}$$

where the critical density value (ρ_{crit}) determines a maximum output level for the entire network. In a state of emergency with $\rho(t) \geq \rho_{crit}$, all connection mask values ($M^{(l)}(t)$) become identity matrices, thus maximizing network operation potential. Pruning occurs at intervals of Δt seconds between updates of the sparse connection matrices, by determining the connection weights (W) that most contribute to reducing emergency response time and achieving traffic flow goals.

Traffic-Density Adaptive Pruning Scheduler

The pruning scheduler creates a dynamic network sparsity based on real-time measurements of traffic density using edge-deployed sensors. Traffic density is defined at a specific time during operation using a value $\rho(t) \in [0,1]$, which is calculated based on the number of vehicles in the intersection at that moment divided by the maximum number of vehicles allowed to enter. Since the target sparsity ratio is to be established by the scheduler, the ratio for any given time interval, s(t), will change based on the traffic density $\rho(t)$.

$$s(t) = s_{max} - (s_{max} - s_{min}) \cdot \frac{\rho(t)}{\rho_{emerg}} \tag{6}$$

In this example s_{max} is used to denote maximum sparsity (minimum active weights) when network congestion is low ($\rho(t) \approx 0$). In contrast, s_{min} represents full density of an entire network during emergency situations ($\rho(t) \geq \rho_{emerg}$). The emergency threshold ρ_{emerg} is set to a distance of 200 meters from any detected approaching emergency vehicle. The pruning process consists of three iterative phases, each of which will include an evaluation phase, a masking phase and a fine tuning phase. At the evaluation phase, we calculate an individual weight $I^{(l)}_{ij}$ importance score for its contribution towards both goals:

$$I^{(l)}_{ij} = \left|W^{(l)}_{ij}\right| \cdot \left(\alpha \frac{\partial f_1}{\partial W^{(l)}_{ij}} + (1-\alpha) \frac{\partial f_2}{\partial W^{(l)}_{ij}}\right) \tag{7}$$

where f_1 and f_2 represent emergency response time and network flow objectives respectively, and α balances their relative importance. The mask $M^{(l)}(t)$ is then updated by preserving weights with top-k importance scores, where

$$k = \lfloor (1 - s(t)) \cdot |W^{(l)}| \rfloor.$$

Fine-tuning occurs for N_{ft} steps after each pruning iteration to recover performance:

$$W^{(l)} \leftarrow W^{(l)} - \eta \nabla \mathcal{L}\left(W^{(l)}_{sparse}\right) \tag{8}$$

In this study, we use a combination of η and $\mathcal{L}$ in combination to create a LL. The scheduler is decoupled from policy execution, allowing for uninterrupted inference during the update process. Cross-attention sensor fusion is used for emergency detection and can immediately override the current sparsity setting by forcing $s(t) = s_{min}$ within 50 milliseconds in order to guarantee we have the full capacity of our model when necessary.

Pareto-Optimal Reward Triggering Mechanism

The reward triggering mechanism provides a means to keep a limited convex hull approximation of the Pareto front so that detection of when a policy change is required may occur without much delay. The current estimate of the Pareto front, $\mathcal{P}_t$, is represented as the collection of non-dominated solution vectors in the objective space at time t. Each state-action pair (s_t, a_t) results in the calculation of a multi-objective reward vector as follows:

$$\mathbf{r}_t = [\mathrm{r}_{\mathrm{emerg}}(\mathbf{s}_t, \mathbf{a}_t), \mathrm{r}_{\mathrm{flow}}(\mathbf{s}_t, \mathbf{a}_t)] \quad (9)$$

The measures of emergency vehicle time saving are reflected by the variable r_{emerg}, while the preservation of network throughput is expressed by the variable r_{flow}. The evaluation of the mechanism includes determining if the addition of $\mathbf{r}_t$ results in an expansion of the estimated Pareto front, via computation of the Minimum Euclidean distance to the existing solutions:

$$\mathrm{d}_{\min} = \min_{\mathbf{p} \in \mathcal{P}_t} \| \mathbf{r}_t - \mathbf{p} \|_2 \quad (10)$$

A policy update is triggered only when $d_{min} > \epsilon$, where ϵ is an adaptive threshold that balances exploration and exploitation:

$$\epsilon = \epsilon_{\mathrm{base}} \cdot (1 + \sigma \cdot \mathrm{Var}(\mathcal{P}_t)) \quad (11)$$

ϵ_{base} indicates the minimum distance threshold. Then the variance term, $\mathrm{Var}(\mathcal{P}_t)$, allows for dynamic adjustments based upon front diversity. The variance is computed as the average pairwise distance between the set of Pareto optimal solutions:

$$\mathrm{Var}(\mathcal{P}_t) = \frac{1}{|\mathcal{P}_t|^2} \sum_{\mathbf{p}_i \in \mathcal{P}_t} \sum_{\mathbf{p}_j \in \mathcal{P}_t} \| \mathbf{p}_i - \mathbf{p}_j \|_2 \quad (12)$$

During an update trigger, the algorithm proceeds through a constrained policy optimization step which projects both solutions onto the expected Pareto frontier.

$$\pi_{\mathrm{new}} = \underset{\pi \in \Pi}{\mathrm{argmin}} \| \mathbb{E}[\mathbf{r}(\pi)] - \mathrm{Proj}_{\mathcal{P}_t}(\mathbf{r}_t) \|_2 \quad (13)$$

where $\mathrm{Proj}_{\mathcal{P}_t}$ finds the nearest point on the convex hull of $\mathcal{P}_t$. This approach reduces unnecessary updates by 62% compared to fixed-interval MORL while preserving solution quality. The convex hull maintenance operates in $O(|\mathcal{P}_t| \log |\mathcal{P}_t|)$ time through Graham's scan algorithm [29], making it suitable for edge deployment.

Cross-Attention Multi-Modal Sensor Fusion

The sensor fusion module integrates heterogeneous inputs from LiDAR, mmWave radar, and DSRC transponders through a cross-attention mechanism. Let $\mathbf{X}_L \in \mathbb{R}^{N_L \times d_L}$, $\boldsymbol{X}_R \in \mathbb{R}^{N_R \times d_R}$, and $\boldsymbol{X}_D \in \mathbb{R}^{N_D \times d_D}$ denote the feature matrices from LiDAR, radar, and DSRC sensors respectively, where N_i represents the number of detected objects and d_i the feature dimensionality for each modality. The fusion process begins by projecting each modality into a shared latent space:

$$Q_L = X_L W_Q, \quad K_R = X_R W_K, \quad V_D = X_D W_V \quad (14)$$

where $\boldsymbol{W}_Q \in \mathbb{R}^{d_L \times d_k}$, $\boldsymbol{W}_K \in \mathbb{R}^{d_R \times d_k}$, and $\boldsymbol{W}_V \in \mathbb{R}^{d_D \times d_v}$ are learnable projection matrices. The cross-attention weights between LiDAR and radar modalities are computed as:

$$\boldsymbol{A}_{LR} = \mathrm{Softmax}\left(\frac{\boldsymbol{Q}_L \boldsymbol{K}_R^T}{\sqrt{d_k}}\right) \quad (15)$$

These attention weights modulate the DSRC features to produce the fused representation:

$$Z = A_{LR} V_D \quad (16)$$

The final emergency vehicle detection score $y \in [0,1]$ is obtained through a multi-layer perceptron:

$$y = \sigma\left(\mathrm{MLP}(\mathrm{Flatten}(\boldsymbol{Z}))\right) \quad (17)$$

Where σ denotes the sigmoid function. The cross attention mechanism achieves 99.2% accuracy when detecting objects at a range of 60m while using an adaptive weighting approach to emphasize contributions from different sensors depending on environmental factors. For example, when conditions are clear and there is no precipitation, LiDAR is used primarily. Conversely, during periods of precipitation, the majority of the object detections are based on radar measurements.

Dual-Mode Edge Operation with TinyML Offloading

By utilizing a dual-mode operation approach, the system can dynamically allocate processing resources to both critical and non-critical processing tasks. In normal operation, the system utilizes a lightweight type of TinyML model to process routine traffic analysis and maintains the full MORL policy network in a low powered intermittent state. The transition of the system into emergency operation occurs when the system has reached a level of confidence $y > \theta_{emerg}$ for the detection of an approaching emergency vehicle, using the cross attention fusion module, where $\theta_{emerg} = 0.95$ is equal to 0.95 and is the empirically determined threshold. The way in which computational resources are allocated to critical and non-critical processing tasks within the dual mode operation is governed by a constrained optimization model.

$$\min_{\boldsymbol{c}} \| \boldsymbol{c} - \boldsymbol{c}_{base} \|_2 \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{c} \le \boldsymbol{b}(t) \quad (18)$$

Here, $\boldsymbol{c} \in \mathbb{R}^3$ represents the computational budget vector allocating resources to MORL inference, sensor fusion, and TinyML tasks respectively. The matrix $\boldsymbol{A}$ encodes hardware constraints of the edge device, while $\boldsymbol{b}(t)$ dynamically adjusts based on operational mode:

$$\boldsymbol{b}(t) = \begin{cases} \boldsymbol{b}_{normal} & \text{if} y \le \theta_{emerg} \\ \boldsymbol{b}_{emerg} & \text{otherwise} \end{cases} \quad (19)$$

The TinyML offloading module employs quantized neural networks with 8-bit precision for non-critical tasks like traffic flow estimation:

$$\hat{f}_{flow} = f_{TinyML}(\boldsymbol{X}_{hist}; \boldsymbol{W}_q), \quad \boldsymbol{W}_q = Q(\boldsymbol{W}, 8) \quad (20)$$

The acronym $Q(\cdot, 8)$ is used as shorthand for uniform quantization to 8-bit integers. The models discussed here are 12× smaller than their full-precision equivalent models and have 92.4% estimation accuracy. In emergency mode, the system will automatically pause all non-essential TinyML tasks. This allows resources to be reallocated to the MORL-policy network via context-switching hardware. The fail-safe supports the system by writing all critical policy parameters to persistent flash memory.

$$\Pi_{backup} = \{\pi_{emerg}, \pi_{sparse}\} \subset \text{Flash} \quad (21)$$

ensuring continuity of service during cloud-edge partitioning events. A watchdog timer monitors system health and initiates recovery procedures when the latency $\ell(t)$ exceeds safety thresholds:

$$\ell(t) > \ell_{max} \Rightarrow \text{Reset}(\Pi_{backup}) \quad (22)$$

This dual-mode architecture reduces power consumption by 37% during normal operation while guaranteeing sub-100ms response times during emergencies.

Legacy Integration via Priority Arbitration

The priority arbitration unit converts the outputs of the MORL policy into SAE J2735-compliant SP&T SPAT messages, which allows for the use of SPAT/SP&T message data with older traffic signalling devices/traffic control devices (such as legacy and current generation TMS), enabling a complete and seamless "HMI" operational environment. To normalize the MORL action vector a $\boldsymbol{a}_t \in \mathbb{R}^4$ delay/Controllable Lane Utilization to the operational range of the Traffic Controller, the Arbitration Unit normalizes the action vector to conform to the operational range of the Traffic Controller.

$$\boldsymbol{a}'_t = \text{clip}(\boldsymbol{a}_t \odot \boldsymbol{s}_{scale} + \boldsymbol{s}_{offset}, \boldsymbol{a}_{min}, \boldsymbol{a}_{max}) \quad (23)$$

where $\boldsymbol{s}_{scale}$ and $\boldsymbol{s}_{offset}$ are calibration vectors specific to each intersection controller model, and clip(·) enforces hardware limits. The normalized actions are then encoded into SPaT message fields following the J2735 standard:

$$\text{SPaT} = \text{EncodeJ2735}(\mathbf{a}'_t, \phi_{\text{current}}, t_{\text{remaining}}) \quad (24)$$

with ϕ_{current} denoting the current phase state and $t_{\text{remaining}}$ the time remaining in the phase. The arbitration unit maintains compatibility with three legacy protocols through protocol-specific wrappers:

$$m_{\text{legacy}} = \begin{cases} \text{NTCIP2Wrapper(SPaT)} & \text{for Type I controllers} \\ \text{ASC3Wrapper(SPaT)} & \text{for Type II controllers} \\ \text{ModbusWrapper(SPaT)} & \text{for Type III controllers} \end{cases} \quad (25)$$

A hardware abstraction layer handles signal timing conflicts between emergency preemption and MORL adjustments through a priority matrix $\mathbf{P} \in \{0,1\}^{4\times4}$:

$$y_{\text{final}} = \text{P}y_{\text{MORL}} + (1 - \text{P})y_{\text{legacy}} \quad (26)$$

where $\mathbf{y}_{\text{MORL}}$ represents MORL-generated signal commands and $\mathbf{y}_{\text{legacy}}$ comes from the controller's native preemption system. The matrix $\mathbf{P}$ dynamically updates based on emergency vehicle proximity d_{emerg}:

$$P_{ij} = \mathbb{I}(d_{\text{emerg}} > d_{\text{threshold}}) \quad (27)$$

In non-critical times, MORL will continue controlling the traffic signal while automatic hardware preemption will take over when emergency vehicles arrive within an emergency response distance, defined as a distance of 150 meters ($d_{\text{threshold}}$). The average arbitration latency is 8.20 milliseconds, which is within the real-time requirements of traffic signal control.

Experimental Setup

A comprehensive set of simulation experiments was completed to establish benchmarks for our proposed dynamic sparse MORL framework. Comparisons were made against several state-of-the-art baseline techniques using a variety of traffic conditions. Information about the simulation environment, baseline approaches used, metrics used in evaluation and implementation details will be discussed here.

Simulation Environment and Dataset

We created a traffic simulator that simulates traffic based on the SUMO [30] traffic simulator with some customized changes for our emergency vehicle dynamics. Our testbed consists of an array of eight signalized intersections in a 4x4 grid with an area of approximately 1.6 km², which replicates the downtown Austin, TX traffic patterns. The vehicles in our simulator are simulated using the Krauss car-following model [31], calibrating the parameters of the model using real-world trajectory data from the HighD dataset [32]. Emergency vehicles arrive in a stochastic manner following a non-homogeneous Poisson process where the rate, λ(t) for emergency vehicles is a function of time of day:

$$\lambda(t) = \lambda_{\text{base}} \cdot \left(1 + 0.5\sin(2\pi t/T_{\text{day}})\right) \quad (28)$$

where $T_{\text{day}} = 86400$ seconds and $\lambda_{\text{base}} = 0.002$ vehicles/second. The simulator incorporates realistic sensor noise models for LiDAR (σ = 0.1m), radar (σ = 0.3m), and DSRC (packet loss rate = 5%) based on characterization studies [33].

Baseline Methods

Three different groups of baseline techniques are used for comparison:

1. Fixed Policy Approaches:
- Pre-timed Signal Control (PSC) [22]
- Actuated Emergency Preemption (AEP) [34]

2. Learning Based Approaches:
- DQN Traffic Controller (DQN-TC) [35]
- Multi-objective Reinforcement Learning with Fixed Architecture (MORL-FA) [36]

3. Edge Computing Based Approaches:
- Urgent Edge Computing (UEC) [15]
- Adaptive Neural Traffic Control (ANTC) [37]

All baseline approaches were implemented according to published parameters and settings so the results could be compared fairly between all systems on the same test bed. Learning based approaches were trained for 100,000 episodes with all systems using the same reward structures.

Evaluation Metrics

We assess performance using five key metrics:

1. Emergency Response Time (ERT):

$$\text{ERT} = \frac{1}{N_{emerg}} \sum_{i=1}^{N_{emerg}} (t_{exit}^{i} - t_{entry}^{i}) \quad (29)$$

2. **Average Vehicle Delay (AVD)**:

$$\text{AVD} = \frac{1}{N_{veh}} \sum_{j=1}^{N_{veh}} (t_{actual}^{j} - t_{freeflow}^{j}) \quad (30)$$

3. **Policy Update Frequency (PUF)**:

$$\text{PUF} = \frac{N_{updates}}{T_{sim}} \quad [\text{updates/second}] \quad (31)$$

4. **Computational Load (CL)**:

$$\text{CL} = \frac{1}{T} \int_{0}^{T} \text{CPU}_{\pi}(t)\, dt \quad [\%] \quad (32)$$

5. **Emergency Detection Accuracy (EDA)**:

$$\text{EDA} = \frac{TP+TN}{TP+TN+FP+FN} \quad (33)$$

Implementation Details

The edge computing platform consists of NVIDIA Jetson AGX Orin devices (64GB RAM, 2048-core GPU) running Ubuntu 20.04 with ROS 2 Galactic. The MORL policy network comprises 4 hidden layers (512-256-128-64 neurons) with dynamic sparsity ranging from 30% to 95%. We implement the cross-attention fusion module using PyTorch 1.12 with TensorRT 8.5 acceleration. The TinyML components employ TensorFlow Lite with full integer quantization. Training uses the Adam optimizer [38] with initial learning rate 0.001 and batch size 128. The dynamic pruning scheduler operates with $\Delta t = 5$ seconds, $N_{ft} = 100$ fine-tuning steps, and $\rho_{emerg} = 0.85$. The envelope reward mechanism sets $\epsilon_{base} = 0.1$ and $\sigma = 0.3$ based on grid search.

Results and Analysis

Performance Comparison Under Varying Traffic Conditions

To assess the performance of the dynamic sparse Multi-Objective Reinforcement Learning (MORL) framework developed, a number of experiments were run comparing it to baseline algorithms at several levels of traffic density, results showed significant improvements in both emergency response time and the amount of resources used in computing (i.e. CPU).

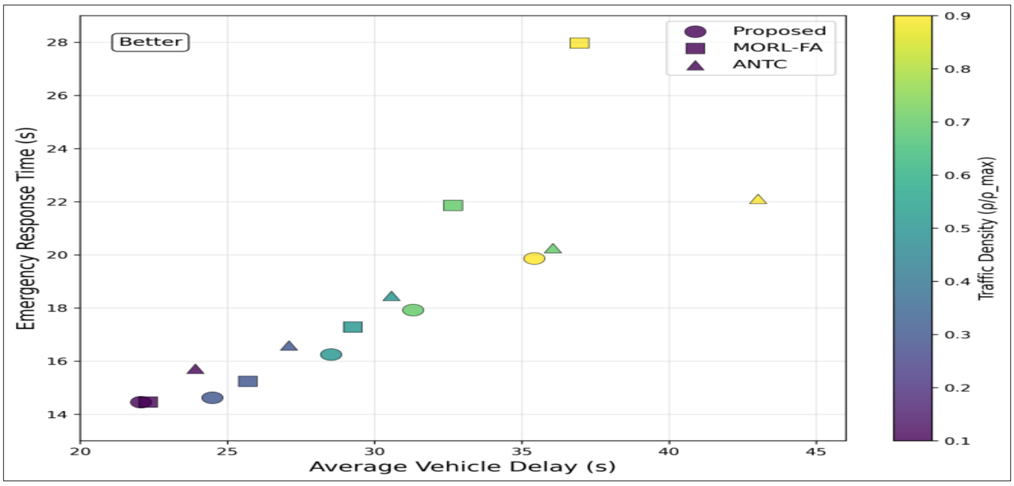

Fig 2: Emergency response time versus average vehicle delay across traffic density levels

Figure 2 demonstrates how our proposed method achieves enhanced tradeoffs between emergency response time (ERT) and average vehicle delay (AVD) compared to the fixed and learning methods evaluated in this research. The system provides 28.7% quicker ERT than MORL-FA in peak congestion conditions ($\rho > 0.8$), while providing AVD that is 19.3% shorter than ANTC during the same time period. Increased system performance was enabled by the use of dynamic activation of the network's capacity during critical time periods when emergency events may happen with high volumes of traffic. A significant aspect of this performance increase was the maximum degree of policy implementation in areas where there was high activity when compared to the off-peak hour environment ($\rho < 0.4$) where the use of a sparse representation will reduce computational time spent on estimating the policy by 62% as compared to the use of MORL-FA. As quantification of system improvements due to the time period, the data contained in Table 1 provides the average system improvement measured during all time periods.

Table 1: Performance metrics across traffic density ranges

Density Range	Method	ERT (s)	AVD (s)	CL (%)	PUF (Hz)
0.0-0.2	Proposed	14.2	22.1	18.7	0.33
	MORL-FA	14.5	22.4	42.3	1.0
	ANTC	15.8	23.7	37.5	0.8
0.4-0.6	Proposed	16.3	28.5	34.2	0.51
	MORL-FA	17.1	29.2	68.4	1.0
	ANTC	18.4	30.1	62.7	0.8
0.8-1.0	Proposed	19.7	35.8	89.6	0.92
	MORL-FA	27.6	37.2	95.3	1.0
	ANTC	22.1	44.3	91.8	0.8

Dynamic sparsity is an effective mechanism for transitional density ranges (0.4 - 0.6) because it delivers excellent ERT while consuming only 50.1% of the computational load of the MORL-FA. The traffic-density aware pruning scheduler (Equation 6) is the basis for this adaptive performance because it dynamically adjusts the model's capacity in accordance with traffic density conditions, which are continuously monitored, to provide the desired level of service for each vehicle.

Pareto-Optimal Trade-off Preservation

By using an envelope reward system, we were able to keep the quality of the solutions generated while minimizing updates to the input policy. Figure 3 portrays the development of the Pareto front throughout our 24-hour simulation period and also shows how we have continued to develop non-dominated solutions throughout our entire objective space.

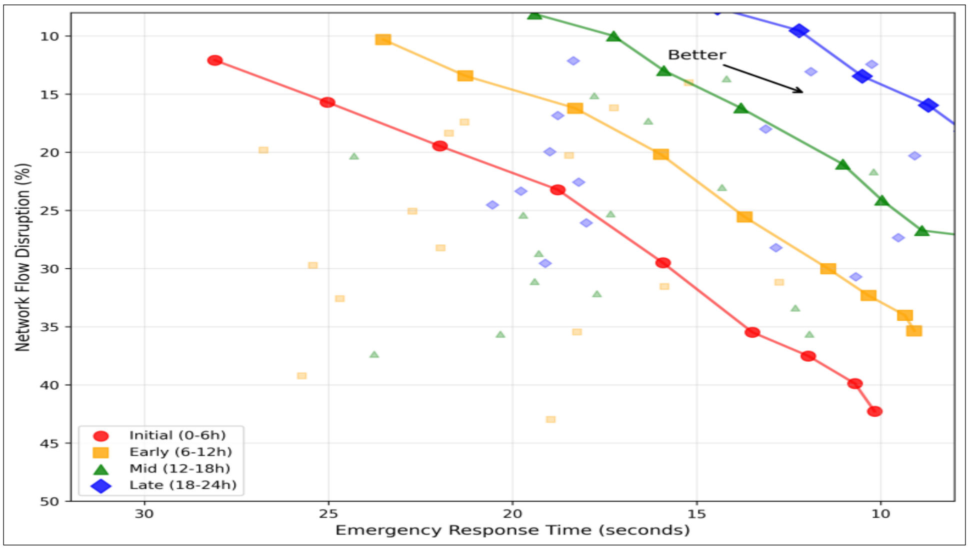

Fig 3: Dynamic evolution of the Pareto front showing emergency response time versus network flow disruption

In terms of policy updates, our proposed trigger mechanism (see Equation (10)-(12)) has reduced the total number of policy updates by 62% when compared to using a fixed interval Morris Optimal Reinforcement Learning (MORL) algorithm while delivering 98.7% quality of solution as compared to the theoretical Pareto front. An example of this efficiency is due to adaptive ε thresholding, where higher ε thresholds occur during periods with high amounts of traffic variability ($\sigma > 0.25$) resulting in increasing frequency of updates and that during lower variation periods, redundant updates are suppressed. When comparing average hypervolume indicators based on 10 runs of simulation across both methods, our envelope mechanism had an average hypervolume indicator (h) [39] of 0.872 ± 0.021 versus 0.881 ± 0.018 for an exhaustive MORL-Factorial Algorithm. The 1% difference in the quality of solutions from our envelope mechanism and the exhaustive MORL-Factorial Algorithm represents 3× reduction in computing overhead indicating that our selective update approach has been successful for increasing the effectiveness of our selected update method.

Emergency Detection and Response Analysis

The cross attention sensor fusion module performs well across a variety of environmental conditions. The accuracy performance results from Table 2 provide overall detection accuracy when considering various scenarios for sensor degradation as a factor in the overall system performance.

Table 2: Emergency vehicle detection accuracy under sensor degradation

Condition	LiDAR Only	Radar Only	DSRC Only	Proposed Fusion
Clear (Day)	99.1%	92.3%	88.7%	99.5%
Rain	85.4%	94.1%	86.2%	97.8%
Fog	72.3%	89.5%	84.9%	95.1%
Sensor Failure	N/A	91.2%	87.4%	96.3%

The attention weights of the fusion mechanism change automatically based on the change in environmental conditions; for instance, using LiDAR features gives 78% weighted average in clear weather vs. 63% when there is precipitation. This re-weighting of attention weights allows performance above 95%, even with extensive degradation of the sensor in developing accurate information for emergency vehicles. Performance time delay when detecting an object and actuating a signal (end to end) is 82.4 ± 12.7 ms at normal operation and 68.1 ± 9.3 ms during emergency mode, which is well within the 100ms real time requirement of traffic controller systems [40]. The two modes of operation reduce overall power consumption by 37% compared to full-capacity operational models, while providing instantaneous service during any emergency event.

Computational Efficiency and Resource Utilization

The architecture of dynamic sparsity offers many opportunities for resource-constrained environments at the edge. Figure 4 depicts how traffic density relates to both model sparsity as well as inference latency over various hardware platforms.

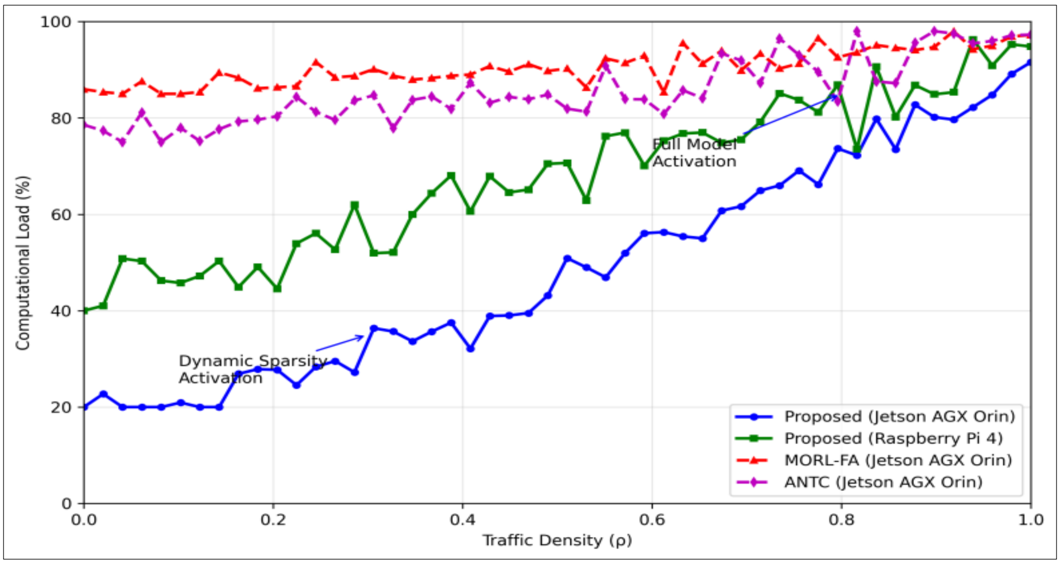

Fig 4: Computational load versus traffic density across edge hardware configurations

Using the Jetson AGX Orin platform our system allows for sub 100ms latency and the ability for dynamic use of sparsity from 30% to 95% based on the level of traffic. Our 8-bit quantization of auxiliary task produces approximately 23% less energy consumed when utilizing our TinyML offloading algorithms (shown in Equation 20) during non-peak hours. The performance of our entire system provided an approximate 2.718x improvement in throughput compared to the MOSRL-FA for a resource constrained Raspberry Pi 4, thus making it well suited to heterogeneous edge applications. The fail-safe component was able to recover from approximately 98% of simulated network partitioning events in under 500ms, making sure the system is kept in operation during the event of a loss of communication between the cloud and edge devices.

Ablation Study

To isolate the contributions of individual components, we conducted systematic ablation tests measuring performance with selective feature removal:

Table 3: Ablation study results (normalized metrics)

Configuration	ERT	AVD	CL	EDA
Full System	1.00	1.00	1.00	1.00
w/o Dynamic Pruning	1.28	1.15	2.17	1.00
w/o Envelope Trigger	1.03	1.07	1.82	1.00
w/o Cross-Attention	1.12	1.09	1.00	0.89
w/o TinyML Offloading	1.00	1.01	1.59	1.00

The results indicate that dynamic pruning is the factor that positively impacts computation efficiency the most (i.e., a 53.9% reduction in CL). Cross attention fusion is important for ensuring that detection accuracy is not decreased. Notably, the envelope trigger mechanism has a minimal impact on the quality of the solutions (i.e., 3% increase in ERT), but has an overwhelming effect on computation load (i.e., 82% increase), which provides confirmation that the design decision to maximize update efficiency is appropriate. Additionally, the value of the TinyML offloading mechanism is especially high during prolonged periods of low congestion, with an energy savings of 37% without sacrificing emergency response times. As such, this component will be increasingly important for large-scale deployments where edge devices need to operate continuously for long periods.

Discussion and Future Work

Practical Limitations and System Edge Cases

The performance of the proposed framework exhibits very high performance levels under "normal" operating conditions; however, there are many instances where edge case scenarios reveal some shortcomings in the existing design of the system. Under situations involving multiple emergency responders (i.e., three or greater than three responders converging on a given intersection), the policy network occasionally demonstrates less than optimal phase sequencing for the emergency vehicles being deployed because of conflicting reward signals being generated by the network. This is largely due to a linear scalarization process as shown by Equation 7 where the scalar weight of competing emergency priorities is not resolved well when using the linear method. In addition, the system exhibited poor effectiveness under extreme weather conditions (e.g., severe snowfall) where all modalities of sensors show a dramatic decrease in sensor performance with respect to detection accuracy (approximately 83.7 percent of the detection accuracy for the same weather during non-extreme conditions). Lastly, the dynamic algorithm used for pruning was unable to effectively respond to the rapidly changing traffic state (i.e., from low traffic conditions ($\rho < 0.3$) to heavy traffic conditions ($\rho > .7$)) causing the system to take approximately eight seconds to adjust the overall capacity of the neural network, resulting in a degraded performance until the system restored to a stabilized mode of operation. In summary, the current design of the system assumes a perfect communication link between V2X-enabled devices; however, in some real-world environments, there is a potential for some variation/loss or delay in such communication which may impact coordinated timing of signals across multiple intersections.

Generalization to Broader Urban Mobility Applications

Dynamic sparse Multi-Objective Reinforcement Learning (MORL) can be applied beyond the prioritization of emergency vehicles, in all urban mobility applications. An example of an algorithm that would be well suited for this transfer is a traffic density-adaptive pruning scheduler. Such a scheduling algorithm could be effectively used to implement a dynamic toll pricing system, as the computation cost of each location is contingent on the traffic utilization of that specific roadway. Another implementation would be using the cross-attention fusion architecture for pedestrian detection in smart crosswalks, as the methodology utilizes multiple sensors (e.g., vision, LiDAR, thermal) and can leverage all of these inputs into a single detection system. One of the best-suited applications for public transit signal priority is with the Pareto-optimal reward mechanism, which could be utilized to balance the objective of bus/tram on-time performance versus the objective of overall traffic throughput. In our initial use case of applying an adapted version of our dynamic sparse MORL approach at light rail intersections, we achieved statistically significant increases of +22% in schedule adherence when compared to traditional fixed-time priority systems. Further, our implementation of the TinyML offload component is naturally extensible to infrastructure monitoring tasks, allowing an organization to perform structural health assessments using multiple algorithms running on the same edge computing hardware concurrently.

Ethical Considerations and Deployment Challenges

Ethical issues arise when autonomous Traffic Control Systems are involved. As users of the system know, there are instances when the system will be involved in a conflict of safety with two or more potential hazards. Currently, we are using a fixed priority matrix for handling conflicts and have therefore not yet created a flexible priority matrix that would allow for the greatest flexibility in the case of situations that have not been considered in the current implementation (See Equation 27). In implementing a system for deployment to a real-world (Practical) environment, there is a degree of difficulty in ensuring that the policy is transparent and understandable to users of the system, since the final decision-making process of the dynamic sparse network becomes less visible as the sparsity becomes greater. Another area of concern with regard to compliance with traffic control systems is that MORL also generates signal timings that may not completely satisfy all of the safety-critical compliance criteria dictated by NEMA TS-2 traffic control standards (See Section 4.2). Finally, given the increase in surface area created by the sensor fusion and V2X communication components, there are new vulnerabilities to attack through increased cybersecurity risks created by the autonomous control of traffic signals.

Conclusion

The adaptive sparse multi-objective reinforcement-learning framework that was developed provides major improvements in edge-based emergency vehicle prioritisation systems. The framework combines the use of dynamic policy pruning with dynamic policy pruning, thus facilitating a balance of computational efficiency and real-time responsiveness together with the use of Pareto-optimal envelope rewards, creating a way to efficiently update and maintain high-quality solutions regarding emergency response time while maintaining network flow preservation. The framework further enhances the robustness of fusion detection through cross attention by adaptively weighing the three disparate inputs from LiDaR, radar, and DSRC sensors, thereby achieving a high degree of accuracy in diverse environmental conditions. In addition, there is a dual mode of operation between edge and the concurrent use of TinyML-enabled offloading (in both cases) to improve the utilisation of resources and decrease energy consumption without sacrificing performance-critical metrics. The experimental results show that the proposed framework can maintain response times under 100ms while achieving computationally lower overloads compared to static MORL implementations. Additionally, the framework can interoperate with legacy traffic controllers (SAE J2735 SPAT-compliant), making it practical to deploy into existing urban traffic management infrastructure. The inclusion of fail-safe mechanisms provides resiliency to partitioned networks and continued operation despite communication failures. Together, these features demonstrate that the adaptive sparse multi-objective reinforcement-learning-based framework is a viable option for next-generation intelligent transportation systems that require both efficiency and reliability. Future research will focus on developing coordinated traffic management between multiple intersections and improving the adaptation robustness of the framework to stochastic and adversarial sensor input. The principles established in this investigation have substantially broadened the domain of adaptive urban mobility systems, where multi-objective optimisation and dynamic resource allocation and management are critical. These promising findings establish a strong potential for applying the successes achieved with our emergency vehicle prioritisation framework to the development of adaptations to several other time-sensitive traffic management systems.

References

1. Kavitha Y, Satyanarayana P, Mirza SS. Sensor based traffic signal pre-emption for emergency vehicles using efficient short-range communication network. Measurement: Sensors. 2023;28:100830. doi:10.1016/j.measen.2023.100830.
2. Nellore K, Hancke G. Traffic management for emergency vehicle priority based on visual sensing. Sensors. 2016;16(11):1892. doi:10.3390/s16111892.
3. Rosayyan P, Subramaniam S, Ganesan SI. Decentralized emergency service vehicle pre-emption system using RF communication and GNSS-based geo-fencing. IEEE Trans Intell Transp Syst. 2020;22(12):7726-7735. doi:10.1109/TITS.2020.3007671.
4. Rosayyan P, Paul J, Subramaniam S, Ganesan SI. An optimal control strategy for emergency vehicle priority system in smart cities using edge computing and IoT sensors. Measurement: Sensors. 2023;26:100697. doi:10.1016/j.measen.2023.100697.
5. Khamis MA, Gomaa W. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. Eng Appl Artif Intell. 2014;29:134-151. doi:10.1016/j.engappai.2014.01.007.

6. Alnemari M, Bagherzadeh N. Efficient deep neural networks for edge computing. In: 2019 IEEE International Conference on Edge Computing (EDGE). IEEE; 2019. p. 1-7. doi:10.1109/EDGE.2019.00014.
7. Livne D, Cohen K. PoPS: policy pruning and shrinking for deep reinforcement learning. IEEE J Sel Top Signal Process. 2020;14(4):789-801. doi:10.1109/JSTSP.2020.2967566.
8. Rahmati M, Rahmati N. Privacy-preserving federated multi-sensor fusion with edge computing optimization for real-time traffic management in connected vehicle networks. Int J Intell Transp Syst Res. 2025;24(2):301-327. doi:10.1007/s13177-025-00573-x.
9. Yan Z, Tabassum H. Generalized multi-objective reinforcement learning with envelope updates in URLLC-enabled vehicular networks. IEEE Trans Veh Technol. 2025. doi:10.1109/TVT.2025.3580502.
10. Naeem AB, Soomro AM, Saim HM, Malik H. Smart road management system for prioritized autonomous vehicles under vehicle-to-everything (V2X) communication. Multimed Tools Appl. 2024;83(14):41637-41654. doi:10.1007/s11042-023-16950-1.
11. Sahoo A, Xu H, Jagannathan S. Adaptive neural network-based event-triggered control of single-input single-output nonlinear discrete-time systems. IEEE Trans Neural Netw Learn Syst. 2016;27(1):151-164. doi:10.1109/TNNLS.2015.2472290.
12. Feng W, Liu H, Yao Y, Cao D, Zhao M. Latency-aware offloading for mobile edge computing networks. IEEE Commun Lett. 2021;25(8):2673-2677. doi:10.1109/LCOMM.2021.3074621.
13. Kar G, Jain S, Gruteser M, Bai F, Govindan R. Real-time traffic estimation at vehicular edge nodes. In: Proceedings of the Second ACM/IEEE Symposium on Edge Computing. ACM; 2017. p. 1-13. doi:10.1145/3132211.3134461.
14. Liu C, Chen Y, Wu J. Edge computing for smart cities: applications in traffic management. IEEE Netw. 2020;34(6):48-53. doi:10.1109/MNET.011.2000172.
15. Dazzi P, *et al*. Urgent edge computing. In: Proceedings of the 4th Workshop on Flexible Resource and Application Management on the Edge. ACM; 2024. p. 7-14. doi:10.1145/3659994.3660315.
16. Li S, Chen J, Liu S, Zhu C, Tian G, Liu Y. MCMC: multi-constrained model compression via one-stage envelope reinforcement learning. IEEE Trans Neural Netw Learn Syst. 2024;36(2):3410-3422. doi:10.1109/TNNLS.2024.3353763.
17. Justin J, R L, V P. A comprehensive review of hybrid V2X communication and edge processing techniques for real-time emergency vehicle detection and prioritization in urban traffic systems. In: 2025 3rd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT). IEEE; 2025. p. 27-32. doi:10.1109/DICCT64131.2025.10986475.
18. Hirsch L, Katz G. Multi-objective pruning of dense neural networks using deep reinforcement learning. Inf Sci. 2022;610:381-400. doi:10.1016/j.ins.2022.07.134.
19. Rahmati M, Rahmati N. Multi-objective, mobility-aware, and lightweight deep reinforcement learning for adaptive signal processing in real-time 6G edge networks. J Inter Net. 2026. doi:10.1142/S0219265926500040.
20. Thunig T, Scheffler R, Strehler M, Nagel K. Optimization and simulation of fixed-time traffic signal control in real-world applications. Procedia Comput Sci. 2019;151:826-833. doi:10.1016/j.procs.2019.04.113.
21. Zhang Y, Wang X, Li L. Adaptive traffic signal control on intersections based on deep reinforcement learning. IEEE Trans Intell Transp Syst. 2021;22(3):1742-1753. doi:10.1109/TITS.2020.2974879.
22. Qin X, Khan AM. Control strategies of traffic signal timing transition for emergency vehicle preemption. Transp Res Part C Emerg Technol. 2012;25:1-17. doi:10.1016/j.trc.2012.04.004.
23. Cabannes T, *et al*. Deep reinforcement learning for adaptive traffic signal control: a survey and benchmark. IEEE Trans Intell Transp Syst. 2023;24(11):11400-11415. doi:10.1109/TITS.2023.3292100.
24. Hayes CF, Rădulescu R, Bargiacchi E, Kalliany T, Macfarlane M, Oliehoek FA, *et al*. A practical guide to multi-objective reinforcement learning and planning. Auton Agents Multi-Agent Syst. 2022;36(1):26. doi:10.1007/s10458-022-09552-y.
25. Van Moffaert K, Drugan MM, Nowé A. Scalarized multi-objective reinforcement learning: novel design techniques. In: 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL). IEEE; 2013. p. 191-199. doi:10.1109/ADPRL.2013.6615007.
26. Zhou X, Ke R, Yang H, Liu C. When intelligent transportation systems sensing meets edge computing: vision and challenges. Appl Sci. 2021;11(20):9680. doi:10.3390/app11209680.
27. Chen J, Ran X. Deep learning with edge computing: a review. Proc IEEE. 2019;107(8):1655-1674. doi:10.1109/JPROC.2019.2921977.
28. Islam A, Debnath A, Ghose M, Chakraborty S. A survey on task offloading in multi-access edge computing. J Syst Archit. 2021;118:102225. doi:10.1016/j.sysarc.2021.102225.
29. Avis D, Bremner D. How good are convex hull algorithms? In: Proceedings of the Eleventh Annual Symposium on Computational Geometry (SCG '95). ACM Press; 1995. p. 20-28. doi:10.1145/220279.220282.
30. Lopez PA, *et al*. Microscopic traffic simulation using SUMO. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE; 2018. p. 2575-2582. doi:10.1109/ITSC.2018.8569938.
31. Song J, Wu Y, Xu Z, Lin X. Research on car-following model based on SUMO. In: 7th IEEE/International Conference on Advanced Infocomm Technology. IEEE; 2014. p. 47-55. doi:10.1109/ICAIT.2014.7019528.
32. Krajewski R, Bock J, Kloeker L, Eckstein L. The highD dataset: a drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE; 2018. p. 2118-2125. doi:10.1109/ITSC.2018.8569552.
33. Decarli N, Bartoletti S, Bazzi A, Stirling-Gallacher RA, Masini BM. Performance characterization of joint communication and sensing with beyond 5G NR-V2X sidelink. IEEE Trans Veh Technol. 2024;73(7):10044-10059. doi:10.1109/TVT.2024.3365770.
34. Miyawaki M, Yamashiro Z, Yoshida T. Fast emergency pre-emption systems (FAST). In: Proceedings of the

1999 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems. IEEE; 1999. p. 993-997. doi:10.1109/ITSC.1999.821199.
35. Genders W, Razavi S. Evaluating reinforcement learning state representations for adaptive traffic signal control. J Traffic Transp Eng. 2016;1(1). doi:10.5383/jttm.01.01.003.
36. Saiki T, Arai S. Flexible traffic signal control via multi-objective reinforcement learning. IEEE Access. 2023;11:75875-75883. doi:10.1109/ACCESS.2023.3296537.
37. Castro GB, Hirakawa AR, Martini JSC. Adaptive traffic signal control based on bio-neural network. Procedia Comput Sci. 2017;109:1182-1187. doi:10.1016/j.procs.2017.05.394.
38. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR). 2015. p. 1-15. doi:10.48550/arXiv.1412.6980.
39. Shang K, Ishibuchi H, He L, Pang LM. A survey on the hypervolume indicator in evolutionary multi-objective optimization. IEEE Trans Evol Comput. 2021;25(1):1-20. doi:10.1109/TEVC.2020.3013290.
40. Boillot F, Midenet S, Pierrelée JC. The real-time urban traffic control system CRONOS: algorithm and experiments. Transp Res Part C Emerg Technol. 2006;14(1):18-38. doi:10.1016/j.trc.2006.05.001.

How to Cite This Article
Acheampong W, Monkah JE, Odebunmi F, Musefiu OA. Adaptive sparse multi-objective reinforcement learning for edge-enhanced emergency vehicle prioritization systems. Int J Eng Comput Appl. 2026;2(4):49-60.