Contents lists available at ScienceDirect

Multimodal Transportation

journal homepage: www.elsevier.com/locate/multra

Full Length Article

Multimodal adaptive traffic signal control: A decentralized multiagent reinforcement learning approach

Kareem Othman[a,b,*], Xiaoyu Wang[a], Amer Shalaby[a], Baher Abdulhai[a]

[a] *Civil engineering department, University of Toronto, Toronto, Ontario, Canada*
[b] *Public works department, Faculty of engineering, Cairo University, Giza, Egypt*

ARTICLE INFO

Keywords:
Adaptive traffic signal
Transit signal priority
Deep reinforcement learning
Multiagent reinforcement learning
Multimodal traffic signal control

ABSTRACT

Public transit is considered a compelling alternative to the car, renowned for its affordability and sustainability, given that a single transit vehicle can accommodate a substantially higher number of passengers compared to regular passenger vehicles. In urban areas, a significant portion of the travel time spent by street-running transit vehicles is consumed waiting at traffic signals. Thus, transit signal priority (TSP) strategies have evolved over the years to give preference to transit vehicles at signalized intersections. Traffic signals are usually optimized for the general vehicular traffic flow, with TSP logic subsequently inserted as an add-on to modify the underlying signal timing plans, thereby granting priority to transit vehicles. However, one major issue associated with the implementation of TSP is its negative impact on the surrounding traffic, creating a conflict between prioritizing passenger vehicles versus transit vehicles. This paper proposes a novel decentralized multimodal multiagent reinforcement learning signal controller that simultaneously optimizes the total person delays for both traffic and transit. The controller, named embedding communicated Multi-Agent Reinforcement Learning for Integrated Network-Multi Modal (eMARLIN-MM), consists of two components: the encoder that is responsible for transforming the observations into latent space and the executor that serves as the Q-network making timing decisions. eMARLIN-MM establishes communication between the control agents by sharing information between neighboring intersections. eMARLIN-MM was tested in a simulation model of five intersections in North York, Ontario, Canada. The results show that eMARLIN-MM can substantially reduce the total person delays by 54 % to 66 % compared to pre-timed signals at different levels of bus occupancy, outperforming the independent Deep Q-Networks (DQN) agents. eMARLIN-MM also outperforms eMARLIN which does not incorporate buses and bus passengers in the signal timing optimization process.

1. Introduction

The level of mobility increases in all cities across the world (Consultancy, 2002; Orski, 2000; Hu et al., 2023). Over the past few decades, there has been a significant increase in the usage of passenger cars, light trucks, and vans. Meanwhile, public transit in many cities has experienced a decline in both ridership and service quality prior to the COVID-19 pandemic (Othman, 2021; Hu et al., 2023). For instance, transit systems in several major urban areas in the United States, including Atlanta, Miami, and Los Angeles, experienced a notable decline in ridership exceeding 7.5 % over a span of four years from 2014 to 2018 (O'Toole, 2018). Similarly, in Canada, ridership data reveal similar trends in numerous cities such as Toronto, Calgary, Vancouver, Montreal, Ottawa,

[*] Corresponding author: Civil engineering department, University of Toronto, 35St George St, Toronto, ON M5S 1A4, Canada.
E-mail addresses: kareem.othman@mail.utoronto.ca, karemmohamed1993@cu.edu.eg (K. Othman).

https://doi.org/10.1016/j.multra.2025.100190
Received 6 May 2024; Received in revised form 9 July 2024; Accepted 13 August 2024

Saskatoon, and Halifax (Diab et al., 2020). Ridership studies have shown that transit travelling speed is one of the key determinants of transit ridership (Litman, 2020). In general, transit speed is one of the most concerning issues facing transit agencies. A century ago, transit seemed fast when compared with the only alternative transportation mode available, which was walking. Nowadays, transit is not faster than it was in 1918. In general, most transit modes are much slower than driving and in many cases transit is slower than cycling (O'Toole, 2018). According to the American Public Transportation Association, the average transit (it should be noted that the term "transit" is used throughout the paper to refer to bus transit) running speed is around 20 km/h (the American Public Transportation Association, 2018). Transit routes in many high demand corridors experience much lower running speeds. On the other hand, the average driving speed is 45 km/h or higher in most cities in the US. As a result, the average work trip time in the US is 50 min by transit and 25 min by car. In Canada, transit has similar operating speeds. For example, buses have an average operating speed of 19 km/h (min 11 km/h- max 28 km/h), while the average driving speed is around 48 km/h (Sweet et al., 2015). As a result, transit speed inferiority is bound to exacerbate congestion and dependence on the car as well as excessive fuel consumption and environmental pollution. The feasibility of alleviating these challenges through the construction of new transportation infrastructure is often hindered by financial constraints, spatial constraints, and apprehensions regarding environmental impact and sustainability. Consequently, an alternative approach to augmenting the capacity of urban transportation networks involves leveraging technology aimed at optimizing the efficiency of existing infrastructure for both transit users (to increase the ridership) and the general traffic.

1.1. Related work

Optimizing traffic signals leads to reduced delays for drivers within urban networks. The predominant types of signal controls currently adopted across the globe are fixed-time and actuated traffic signals. Fixed-time controllers rely on historical traffic data to optimize signal timing offline, maintaining a static configuration once deployed (Webster, 1958; Gartner et al., 1990; Robertson, 1969; Trafficware, 2015). Conversely, actuated signal controllers exhibit greater responsiveness to traffic dynamics by integrating sensor feedback (El-Tantawy et al., 2013). Pedestrians can be integrated into actuated control schemes through the activation of a button, thereby influencing traffic signal priorities. Moreover, actuated control systems facilitate the implementation of Transit Signal Priority (TSP) mechanisms (Sims, 1979), which enhance the efficiency of public transit operations. Nonetheless, they do not explicitly target delay optimization. On the other hand, Adaptive traffic signal controllers (ATSC) represent a more sophisticated approach capable of outperforming traditional control mechanisms, especially in stochastic operating environments. Their heightened efficacy stems from their continual adjustment of signal timings to optimize their objectives (Evans, 2006). Among the notable ATSC systems such as SCOOT (Hunt et al., 1981), SCATS (Sims, 1979), PRODYN (Henry et al., 1984), OPAC (Gartner, 1983), UTOPIA (Mauro and Di Taranto, 1989), and RHODES (Head et al., 1992), optimization occurs through the utilization of internal models of the operating environment. However, these models often exhibit simplifications and may not remain current with real-time conditions, while the optimization algorithms frequently rely on heuristic and suboptimal methods. Given the stochastic nature of traffic patterns and driver behavior, precise traffic modeling presents a formidable challenge. Realistic models tend to be more complex and thus harder to manage, posing a trade-off between complexity and practicality in controller design. Additionally, traffic control entails a sequential decision-making process wherein each decision influences future traffic states. This inherent complexity precludes the straightforward computation of an optimal solution. Nevertheless, the emergence of Reinforcement Learning (RL) has ushered in significant advancements in this domain (Abdulhai and Kattan, 2003; Abdulhai et al., 2003; Bazzan, 2009; Chen and Cheng, 2010; El-Tantawy et al., 2013; Genders and Razavi, 2016; Gao et al., 2017; Gong et al., 2019). RL algorithms excel in addressing such complex problems, as they can iteratively learn optimal control strategies through interaction with the real-world environment and self-assessment of their performance (Sutton and Barto, 2018).

Conventional RL methods are typically formulated in a tabular structure, primarily suited for addressing problems with relatively small state-action spaces. However, in RL applications where the number of state-action combinations is exceedingly large, certain cells within the Q-table (which is a data structure used to store the values of action-state pairs which evaluate the quality of taking certain actions in specific states) may remain unvisited or underexplored during training. Consequently, the controller may encounter unknown states necessitating exploration. To address such challenges, RL with function approximation, such as Deep Q-Networks (DQN), proves to be more suitable for handling extensive state-action space problems (Xu et al., 2014). The incorporation of function approximators, like neural networks, facilitates enhanced interpolation and generalization capabilities. To elaborate, tabular RL approaches exhibit high sensitivity to the dimensionality of the state space, as they rely on discretization techniques for continuous state variables, presenting them in tabular form. Furthermore, they are susceptible to the curse of dimensionality phenomenon in environments with large state spaces. Recent advancements in RL with function approximation techniques have paved the way for leveraging RL methods in continuous and extensive state spaces (Mnih et al., 2013). A subset of RL with function approximation, known as Deep Reinforcement Learning (DRL), has recently reached maturity. DRL demonstrates the capacity to tackle large state space problems effectively and achieve superior performance compared to other RL techniques employing function approximation. In studies such as Genders and Razavi (2016) and Gao et al. (2017), the street surface is discretized into small cells, forming a matrix comprising positions and speeds of vehicles approaching the intersection to optimize traffic signals for a single intersection. RL techniques have been frequently used in the literature for optimizing traffic signal controls with the primary aim of enhancing traffic operations. El-Tantawy and Abdulhai (2010) developed a single-agent RL algorithm using tabular Q-learning. They investigated the appropriate state model for different traffic conditions. Three models were developed, each with a different state representation. The models were tested on a typical multiphase intersection to minimize the vehicle delay and were compared to the pre-timed control strategy as a benchmark. Pol and Oliehoek (2016) developed a single-agent DRL ATSC algorithm that minimizes the vehicle delay and waiting time for a hypothetical case study. They focused on a simple case study during which the action an agent takes is a choice between two different configurations (as there are only two phases) of traffic light settings. Shabestary and Abdulhai (2018) developed a single-agent DRL ATSC algorithm (DQTSC) that maximizes the overall intersection throughput using DQN which is a combination of the Q-learning algorithm and deep convolutional network. Li et al. (2020) developed a single-agent DRL algorithm that employs DQN for a hypothetical case study. Three different traffic conditions were tested in a simulation including uniform, nonuniform distributions, and sudden changes in traffic directions. In general, a large number of studies adopted RL for developing ATSC that improve traffic operations at the intersection level (Dujardin et al., 2011; Shabestary and Abdulhai, 2019; Genders and Razavi, 2020, 2022; Miletić et al., 2020; Gong et al., 2020, 2022; Li et al., 2020; Wang et al., 2023a, 2023b; Yazdani et al., 2023; Ducrocq and Farhi, 2023; Bouktif et al., 2023; Kolat et al., 2023; Abdulhai et al., 2003; Gao et al., 2017). While these studies show promising results, they do not consider public transit or the impact on transit users.

There are multiple sources of transit delays such as traffic signals, the number of bus stops along the bus route, the spacing between these stops and traffic volumes. In general, traffic signals have always been recognized as a major source of delays for public transit. It is estimated that traffic signals account for 10 to 20 % of the total transit travel time (Levinson and Mentor, 2003), and in dense urban areas these percentages can reach up to 35 % (Levinson and Mentor, 2003; Hu et al., 2023). Transit Signal Priority (TSP) has been widely implemented to facilitate the movement of transit vehicles. TSP prioritizes the movement of transit vehicles at intersections by modifying the original signal plan such as the start and end times of signal phases, and the sequence of phases. The green extension and red truncation are the most commonly implemented forms of active TSP. The green extension strategy extends the green phase duration to provide extra time for transit vehicles to clear the intersection. The red truncation strategy, also known as early green, reduces the red phase duration to reduce the time left for the return of the next green interval, thus reducing transit signal delay. The application of the green extension and red truncation strategies in Toronto and Melbourne streetcars provides 6 to 8 % reduction in transit travel time (Currie and Shalaby, 2008; Smith et al., 2005). However, one major issue associated with the implementation of TSP is its impact on the surrounding traffic (Shalaby et al., 2006). Although TSP can generally improve transit performance, it increases the traffic delays on the cross street. For example, if the traffic demands on both the mainline and on side streets are high, TSP may degrade the overall performance of the signal performance (Schultz et al., 2020; Shalaby et al., 2021). A study by Skabardonis and Christofa (2011) showed that TSP can deteriorate the intersection LOS by up to two levels (e.g., from LOS C to LOS E). While the application of RL for TSP seems feasible and promising (similar to ATSC), RL has been rarely used in the context of TSP. Hu et al. (2023) developed a two-way TSP algorithm for optimizing transit reliability and speed at a single intersection. This study proposed a dual-objective two-way TSP algorithm (D2 TSP) that optimizes both transit delays and reliability (headway variability) using deep reinforcement learning (DRL). The DRL models are enhanced with a prioritization heuristic, which chooses actions that optimize the overall performance of buses operating in opposite directions. Long et al adopted the QMIX RL algorithm to improve both transit travel time and headway regularity at three intersections for a hypothetical case study. Long et al. (2022) used single agent RL to develop a TSP system based on the extended Dueling Double Deep Q-learning with invalid action masking (eD3QNI) to reduce the overall travel time for transit users. Similarly, the studies by Shen et al. (2023), Chen et al., 2020 adopted single-agent RL to develop a TSP system that reduces transit travel times through the intersection area. While these studies show promising results for improving the performance of buses, they do not optimize for or consider the impact on the general traffic.

1.2. Research gap, objectives, and contribution

It is evident from the literature that previous studies are either transit-oriented or traffic-oriented. However, it is also evident that optimizing for one mode will deteriorate the performance of the other. In addition, in real life, the transportation network is shared between both transit and traffic, which necessitates the need for multimodal ATSC that optimizes the conditions for both transit and traffic. The term multimodal will be used in this paper referring to traffic and transit. In general, studies that focus on multimodal traffic signal controls are rare such as Dujardin et al. (2011) and Christofa et al. (2013) studies which employed mixed integer linear programming to minimize the overall person delay at the intersection. However, the primary focus of the linear programming model was placed on the explicit formulation and representation of system performance (model of the system), typically involving assumptions that simplify the dynamic traffic environment. These assumptions, such as deterministic traffic flow, may overlook or oversimplify the stochastic nature of the transportation system. On the other hand, Shabestary and Abdulhai (2019) developed a multimodal single agent, model-free DRL ATSC algorithm (MiND) that maximizes the overall person throughput rather than vehicles, regardless of what mode they are on. Similarly, Yu et al. (2023) used RL for the purpose of optimizing traffic signals for both transit and traffic for a hypothetical case.

From the above review, it is evident that RL is a promising method that can be used to optimize traffic signals for both transit and traffic. However, the above literature review has revealed the following gaps:

- Need for multimodal multiagent RL ATSC models: RL has been frequently used to optimize traffic signals for the general traffic. On the other hand, the application of RL for optimizing traffic signals for improving transit operations is rare. In addition, studies that apply RL for multimodal objectives are scarce and were developed to optimize only for one single intersection (single agent RL). This highlights the need for further studies that use RL to optimize traffic signals for both transit and traffic across multiple intersections.
- Vehicle-based vs. Person-based RL ATSC models: Vehicle-based RL models do not differentiate between the different types of vehicles (cars, buses) and deal with them similarly. Vehicle-based RL models are the most common type in the literature which focus on optimizing traffic signals for the general traffic (minimizing the overall delays of the general vehicular traffic at the intersection). On the other hand, person-based models assign weights to the different types of vehicles based on the occupancy

of the different vehicle types. These models, which generally focus on maximizing the overall person throughput through the intersection, are rare in the literature. While both person-based and vehicle-based RL models have been developed in the literature, none of the previous studies compare the performance of these two types of models. However, it is important to evaluate the impacts of these two types of models and compare their performance in order to understand their benefits and limitations.

- Evaluating the impact of the bus occupancy level (i.e. bus load) on the efficiency of both Vehicle-based and Person-based models: While bus occupancy cannot affect vehicle-based RL models, it will affect the performance of person-based RL models. Thus, it is important to evaluate the performance of person-based models under different bus occupancy levels in comparison to vehicle-based models.
- Independent vs. coordinated agents: There are two different methods for modeling ATSCs as a multiagent reinforcement learning (MARL) problem. In the first, and most commonly method used for ATSC, each RL agent focuses on optimizing its own intersection without sharing any information with surrounding intersections. However, the performance of the intersection is usually affected by its neighboring intersections (Wang et al., 2023a, 2023b). As a result, the second type of MARL models focuses on coordinating different agents by sharing information between neighboring intersections. This type of MARL model is known as coordinated MARL and it is more complex as it usually requires higher computational power when compared to independent MARL. As a result, it is important to evaluate and compare the impacts of these two types of MARL models to understand their benefits, limitations, and performance.

The distinctive characteristics of each mode of transportation pose considerable challenges in designing a real-time adaptive signal controller capable of: (1) integrating both transit and non-transit vehicles simultaneously, (2) harnessing contemporary pervasive data on vehicles' speed, position, and occupancy, (3) optimizing for multiple traffic signals, and (4) making decisions at a fine temporal resolution, consistent with the modern standards in state-of-the-art adaptive traffic signal control, typically operating on a second-by-second basis. In this study, we present a novel multiagent deep learning-based multimodal signal control system aimed at addressing the aforementioned challenges. Our approach builds on Wang et al. (2023a, 2023b) work, specifically the use of the eMARLIN, which leverages multiagent deep reinforcement learning techniques. eMARLIN is distinguished by its model-free nature, self-learning capability, and real-time adaptability to the dynamics of the environment or the transportation network. In this paper, we extend the eMARLIN method and platform to explicitly incorporate transit and optimize for both modes: general traffic and transit. We focus on optimizing person throughput, regardless of the mode of transportation they utilize. We refer to the system as eMARLIN multimodal (eMARLIN-MM) in the remainder of the paper. Through this methodology, vehicles with higher occupancy, such as transit vehicles (buses), inherently exert influence on signal timing in their favor, given their role in serving a greater number of individuals. Consequently, the overall outcome ensures equitable optimization of delay for all stakeholders involved. In addition, this paper focuses on providing a comprehensive analysis of the performance of different MARL logics of ATSCs as follows:

- Analyze the impacts of multimodal RL models (that optimize traffic signals to simultaneously improve transit and traffic performance) on the performance of traffic, transit, and the entire corridor (transit + traffic) in both single agent (for a single intersection) and independent and coordinated multiagent (for multiple intersections) settings.
- Test and compare the impact of two different MARL techniques: the first is independent agents (IQL) and the second is coordinated agents (eMARLIN-MM) that share information across neighboring intersections.
- Evaluate and compare the performance of vehicle-based and person-based RL models (single and multiagent) to understand the impacts, benefits, and limitations of each method.
- Evaluate the impact of different levels of bus occupancy on the performance of the different RL models.

Our method represents a fully decentralized controller wherein each intersection communicates exclusively with its immediate preceding and succeeding neighbors. Notably, within the realm of Reinforcement Learning (RL)-based ATSC, only a limited number of approaches fall under this category. For instance, MARLIN (El-Tantawy et al., 2013) and its variations coordinate agents by exchanging their local observations and actions with immediate neighbors. To ensure learning stability, each agent in MARLIN maintains a set of policy estimators for its neighbors, which significantly increases the complexity of the models.

The reminder of the paper is organized as follows: section two discusses the Markov Decision Process and its applications to RL problems. The third section discusses the methodology followed in this paper and shows the proposed MARL framework, the design of the RL agents, and the experimental setup. Then, the fourth section presents the results of the proposed ATSC framework on the performance of traffic and transit while comparing these results to the traditional methods and to other baseline RL methods from the literature. Section five focuses mainly on analyzing the impact of vehicle occupancy on the performance of the different methods tested in this study. Finally, the conclusion section focuses on highlighting the main outcomes of this study.

2. Markov decision process

A Markov Decision Process (MDP) is a mathematical framework used to model decision-making in situations where outcomes are random and partly under the control of a decision-maker. MDPs provide a formal and widely used framework for modeling and solving sequential decision-making problems under uncertainty. In an MDP, the decision-making process evolves over a sequence of discrete time steps. At each time step, the decision-maker observes the current state of the system, selects an action from a set of possible actions, and then receives a reward and transitions to a new state based on a probability distribution determined by the current state and the chosen action. Crucially, the transition probabilities satisfy the Markov property, meaning that the future state depends only on the current state and the action taken, and not on the history of previous states and actions. The key components of an MDP are:

1. **States (*S*):** The possible situations or configurations that the system can be in The decision-maker makes decisions based on the current state of the system.
2. **Actions (*A*):** The available choices that the decision-maker can make in each state. The set of actions available may vary depending on the state.
3. **Transition probabilities (*T*):** The probabilities of transitioning from one state to another given a particular action. These probabilities capture the stochastic nature of the system's dynamics.
4. **Rewards (*R*):** The immediate numerical feedback that the decision-maker receives after taking an action in a particular state. The goal of the decision-maker is typically to maximize the cumulative reward over time using the discount factor $\gamma \,\epsilon\, [0,\ 1]$.
5. **Policy (π):** A strategy that specifies which action to take in each state. The policy can be deterministic (mapping each state to a single action) or stochastic (mapping each state to a distribution over actions).

The primary objective in solving an MDP is to find an optimal policy that maximizes the expected cumulative reward over time. The policy function π maps states to actions. It can be deterministic, where for each state s, it specifies exactly one action a, denoted as $\pi(\mathrm{s}) = a$. Alternatively, it can be stochastic, where for each state s, it provides a probability distribution over actions, denoted as $\pi(\mathrm{a|s})$, which represents the probability of taking action a in state s. The state-action value function, often denoted as Q(s,a), represents the expected cumulative reward obtained by starting in state s, taking action a, and then following a specific policy thereafter. Mathematically, it is expressed using the following equation.

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \epsilon\ S} T\left(s'|s,a\right) \sum_{a' \epsilon\ A} \pi\left(\mathrm{a}'|s'\right) Q^{\pi}\left(s',a'\right) \tag{1}$$

If the dynamics of the environment in a MDP are perfectly known, finding the optimal policy becomes a problem of dynamic programming. Dynamic programming algorithms, such as value iteration or policy iteration, can be used to find the optimal policy in such cases (Sutton and Barto, 2018; Szepesvári, 2022; Correll et al., 2022; Russell and Norvig, 2016). On the other hand, Model-free RL methods are approaches to learning optimal policies in reinforcement learning settings without explicitly modeling the dynamics of the environment. Unlike model-based methods, which require the knowledge of transition probabilities and rewards, model-free methods learn directly from experience gained through interactions with the environment. Model-free methods are particularly useful in scenarios where the environment's dynamics are unknown, complex, or difficult to model accurately such as the transportation network. These methods are versatile and applicable to a wide range of problems, including those with high-dimensional state or action spaces. Value-based methods are used in model-free methods to estimate the value (or expected return) of being in a particular state and taking a particular action. Q-learning is one of the most popular value-based methods that focuses on finding the optimal policy by fitting the Q-function using temporal-difference learning. It involves updating the action-value function based on observed transitions and rewards, using the Bellman equation (Bellman, 1956) shown in Eq. (2). Nevertheless, the optimal Q-function remains either unknown or suboptimal until the agent undergoes training to convergence. Traditional Q-learning methods typically depict Q-values in a discrete or tabular format, known as the Q-table. The agent updates the Q-value for the observed state-action pair using the Q-learning update rule shown in Eq. (3) using α as the learning rate. In this paradigm, the agent must visit all state-action pairs a sufficient number of times during the training phase to converge towards optimal values. However, this approach becomes computationally inefficient as the size of the state space and the number of possible actions expand. Tabular Q-learning becomes impractical when dealing with large or continuous state spaces. Storing Q-values for every possible state-action pair requires exponential memory, making it infeasible for high-dimensional state spaces. In addition, tabular Q-learning may struggle to learn in environments with sparse rewards, where the agent receives feedback only occasionally. Since Q-values are updated based on observed rewards, sparse rewards can result in slow learning or the failure to discover optimal policies. As a result, function approximation techniques such as deep neural networks (DNN) are used to approximate the Q-function allowing for more efficient representation of Q-values in high-dimensional or continuous state spaces (Mnih et al., 2015). The deep neural Q-network is usually trained through stochastic gradient descent to minimize the squared temporal difference (TD) loss.

$$Q(s,a) = \mathbb{E}_{s' \in S}\left[r_t + \gamma \max_{\mathrm{a}'} Q\left(s_{t+1},a'\right)|s,a\right] \tag{2}$$

$$Q'(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q\left(s',a'\right) - Q(s,a)\right] \tag{3}$$

3. Methodology

An MDP can be employed to optimize a system that consists of multiple intersections. The computational complexity associated with solving an MDP increases exponentially with the system's scale. When considering large-scale networks, modeling the entire traffic network comprehensively and training a single centralized RL agent becomes impractical. To facilitate a more manageable representation, we make the assumption that the collective local observations at intersections can encapsulate the dynamics of the entire system, thereby ensuring full observability of the system. Consequently, we formalize the problem of multi-agent ATSC as decentralized Markov Decision Processes (DEC-MDPs). DEC-MDPs extend the conventional MDP framework to accommodate scenarios where multiple agents within a single system take control over the system components. These agents act collaboratively and synchronously influence the system's dynamics, thereby establishing inter-agent dependencies. Formally, a DEC-MDP is defined as a tuple, where the state of the system is represented as a joint aggregation of the local observation components. Thus, the problem

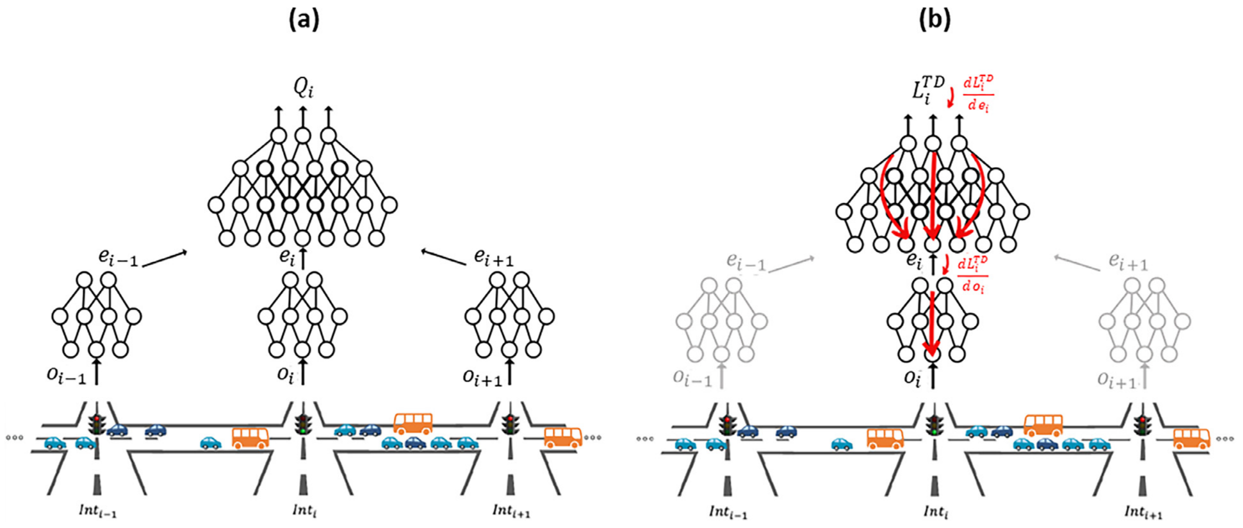


Fig. 1. Illustration of the eMARLIN-MM structure from the perspective of one intersection (agent i) showing: (a) the forward pass and how different agents send their encoded observations $e_{i\ \pm 1}$ (not raw observations $o_{i\pm 1}$) to agent i to make actions (b) the backward pass showing the backpropagation pass and how the weights of the executor and the local Encoder (only for agent i) are updated.

is transformed into a stochastic game, wherein each agent assumes control over a distinct intersection and independently learns its policy with the constraint that every agent can observe information from its direct neighbors. In other words, agent i does not gain access to the global state of the system S but can only observe a subset of the system based on its succeeding and preceding neighbors. Furthermore, within the context of multiple agents learning within a shared environment, the policies of individual agents evolve over the course of training, thereby introducing non-stationarity in dynamics to neighboring agents and exacerbating the challenges associated with distributed agent learning. Relying solely on local observations O_i leads to a scenario where agents greedily compete within this nonstationary environment, preventing the possibility of any coordination. As a result, it becomes imperative to ensure sufficient information sharing between neighboring intersections so that the communication between these agents facilitates coordination among the different agents in the system.

3.1. Proposed eMARLIN-MM framework

We propose a multimodal extension of the novel approach termed embedding communicated Multi-Agent Reinforcement Learning for Integrated Networks - eMARLIN (Wang et al., 2023a, 2023b). We call the multimodal version eMARLIN-MM. Our framework comprises two primary modules: an encoder and an executor. Each agent's encoder transforms its respective raw observations into a latent space, denoted as the "observation embedding." Subsequently, agents disseminate their own embeddings to neighboring agents (the immediate succeeding and preceding neighbors), rather than transmitting raw observations. They then gather and concatenate embedding vectors from neighboring agents, incorporating them as input to the executor module. The executor module serves as a Q-network, estimating the Q-value associated with each potential action and making decisions accordingly. Each agent jointly trains its encoder and executor modules using the Deep Q-Network (DQN) algorithm in an end-to-end fashion. This approach entails the regulation of the encoder module by the gradient derived from the downstream task, wherein the embeddings from neighboring agents are integrated. Consequently, the embeddings in eMARLIN-MM not only serve as a representation of raw observations but also encapsulate additional implicit information derived from the peculiar design of the training process. Notably, the executor module treats the embeddings received from neighboring agents as constant inputs, thereby ensuring that these inputs do not influence the agent's encoders. This design choice alleviates pressure on both the communication and computational systems. A visual representation of the eMARLIN-MM method applied is presented in Fig. 1 showing the structure of one agent. In this setup, each agent independently trains its own Q-network. The output of the network corresponds to the Q-value associated with various actions, while the inputs comprise the agent's observation O_i and the communicated signal from neighboring agents e_j, where $j \epsilon N_i$.

As mentioned earlier, the network architecture comprises two primary components: an encoder and an executor. The encoder serves as the initial component, and it is a learned neural network mapping local raw observations $o_i\ \epsilon\ O_i$ onto a latent space ε_i that has a lower dimension than the raw observations O_i. For each agent, this function (the encoder) maps the local observation O_i to a latent space ε_i, thereby generating the corresponding embedding e_i, denoted as the observation embedding of the agent. The architecture of the Encoder is specifically designed to avoid the transmission of heavy raw sensed data, such as images or videos. This design contributes to the lightweight nature of the eMARLIN-MM approach. The Executor module is the second component, and it serves as the Q-network responsible for generating final decisions within the eMARLIN-MM framework. The Executor is a neural network in which the inputs consist of the embedding derived from the agent's observation, concatenated with the embedding vectors communicated from neighboring agents $e_i|e_j,\ \ j \in N_i$. Thus, the executor makes decisions based not only on its observation space but rather over the collective embedding space encompassing the agent and its neighboring agents. This setup bears some similarity to the methodology employed in MARLIN (El-Tantawy et al., 2013), where Q-learning is conducted based on observations from the agent and its neighbors. However, in contrast to MARLIN, our approach does not necessitate the estimation of neighbor policies; instead, it solely relies on neighbor embeddings to provide contextual information. This distinction renders our method considerably more efficient than a deep learning version of MARLIN in terms of both complexity and convergence rate. The inference process, shown in Fig. 1(a), involves the forward pass of the network. Each agent concurrently evaluates its observation and generates its observation embedding. It is important to note that each agent learns a distinct embedding mapping based on the network's topology and its specific location within it. Subsequently, the embedding is transmitted; each agent sends its embedding to all of its neighboring agents. Finally, each agent aggregates all available embedding vectors into a single vector and forwards it to its executor network to evaluate the Q-values of the available actions. During the training phase, gradients are backpropagated through the network and the TD loss is computed and differentiated with respect to both the agent's observation and the embedding vectors from neighboring agents as shown in Eq. (4). In this setup, the embedding vectors from neighboring agents are treated as constant inputs, influencing both the Q-values (decisions) and the loss function. Consequently, the Executor module learns to incorporate these embeddings into its decision-making process. Importantly, the calculation (gradient) of agent i does not affect the embedding of its neighboring agent $j \in N_i$. As a result, there is no need for the transmission of gradients to the neighboring agents. For agent i, only the weights of its own Executor and the local Encoder are updated during training, as illustrated in Fig. 1(b). This design is driven by two key intentions:

1. **Reduction of the Needed Communication Bandwidth:** By avoiding the transmission of gradient signals, the communication bandwidth required is minimized. This facilitates distributed learning among agents, as the need for exchanging gradients is obviated.
2. **Local Training Signal Regulation:** By exclusively regulating Encoders based on local training signals, a portion of the policy information is captured. Consequently, embedding vectors can convey this pertinent information to coordinating agents. This ensures that each agent maintains a degree of independency in its learning process while still benefiting from the information gleaned from neighboring agents.

$$\nabla L_i\left(o_i,\ e_j\right) = \frac{dL_i}{do_i} = \frac{\partial L_i}{\partial e_i}\frac{\partial e_i}{\partial o_i} + \sum_{j\in N_i}\frac{\partial L_i}{\partial e_j} \tag{4}$$

3.2. Design of the agents

1- ***State (observations):*** The local information collected or sensed at each intersection (agent i) includes seven distinct components:
   a- The number of cars in each lane of each intersection approach.
   b- The speeds of the cars travelling in each lane of each intersection approach.
   c- The number of buses in each lane of each intersection approach.
   d- The speeds of the buses travelling in each lane of each intersection approach.
   e- The occupancy of the buses traveling in each lane of each intersection approach.
   f- The index of the current signal phase.
   g- The elapsed duration of the current phase, relative to the minimum/maximum permissible green time.
2- ***Action:*** At every time step, each agent (i) selects its own action independently $a_i \in A_i$. The agent can select one of the possible following actions:
   a- Extend: In this action, the agent i chooses to extend the current phase for one additional time step.
   b- Switch: In this action, the agent i chooses to switch or change the current phase to one of the permitted subsequent phases.

The permissible phase transitions are contingent upon the agent's present phase. For instance, if from a particular phase the agent has the option to transition to one of two potential phases, then there will be three available actions from that phase:

a- Extend: Continue the current phase.
b- Switch to First Possible Phase: change to the first possible subsequent phase.
c- Switch to Second Possible Phase: change to the second possible subsequent phase.

In addition, phase durations are bounded by minimum and maximum time limits. When the phase duration falls below the minimum threshold, the agent is restricted to the Extend action only. Conversely, when the phase duration reaches the maximum threshold, the agent is limited to Switch actions exclusively. Upon selecting a Switch action during a green phase, the traffic light transitions into a yellow phase, followed by a red phase.

In traffic signal control systems, phasing schemes refer to the timing patterns used to allocate green time to different movements at intersections. There are three main types of phasing schemes: constrained variable (CVPS), variable (VPS), and fixed (FPS) phasing schemes (Chia et al., 2017). CVPS is a phasing scheme wherein the permissible phase transitions are dependent upon the source phase. VPS, which is considered a special case of the CVPS, assumes that all the phase transitions are allowed (i.e., all phases except the current phase are permissible as the next phase). In contrast, if only one possible transition is permitted from each phase, it is termed an FPS. In practice, an FPS dictates a predetermined phase order, with the agent primarily controlling the duration of each phase (Chu et al., 2019). In recent ATSC literature, VPS is a commonly chosen phasing scheme. However, using VPS raises the risk of certain turning movements at the intersection being starved of green time, i.e., not being served any green time in a cycle

(Chia et al., 2017; Chu et al., 2019; Chen et al., 2020). In this study, we employ the CVPS phasing schemes that ensure all major movements within a cycle are served. In this case, each intersection has a CVPS which defines the allowed phase sequence and transitions. Further details about the phasing schemes adopted at the different intersections are provided in the case study subsection.

3- Reward: In our case, the agent's objective is to maximize its cumulative reward over time, thus necessitating a reward function that aligns with the problem's objective. Given our goal of minimizing intersection person delays (or alternatively maximizing the person throughput at the intersection), we opt to define the reward function as negative of the delay on all approaches of the intersection. In other words, the cumulative delay at the intersection is computed as the summation of negative delays incurred by individual vehicles, whether regular or transit. The delay experienced by each vehicle is a function of the number of passengers it carries (occupancy). If a vehicle's speed falls below a certain threshold, its cumulative delay increases by the number of passengers onboard. It should be noted that the reward signal in real applications also comes from sensor readings, same as the input observations. In some cases, precise vehicle speeds cannot be obtained from the detection, but only a categorization of moving/stopped. Here, we can only use queue lengths to approximate the delay time. The use of a fixed threshold in delay calculations serves several practical purposes in our study. Firstly, in real-world applications, sensor data often provides imperfect or incomplete information, particularly regarding precise vehicle speeds. This limitation necessitates alternative metrics, such as queue lengths, to approximate delay times accurately. By employing a fixed threshold, we can consistently interpret these imperfect data inputs into actionable delay estimates, ensuring robustness and reliability in our calculations. However, it should be mentioned that the potential advantages of dynamic speed thresholds that adjust based on real-time traffic conditions, road types, and intersection characteristics. Such an approach could theoretically enhance accuracy by tailoring delay calculations more closely to current environmental factors affecting traffic flow. This adaptability may be particularly beneficial in scenarios where precise speed data is reliably available, allowing for more nuanced delay assessments that reflect immediate changes in traffic dynamics. However, generally, sensor readings may not always provide detailed speed information but rather categorical data on vehicle movement (moving/stopped). Thus, the use of queue lengths as a proxy for delay remains a practical approach. This method aligns with the operational constraints and data availability typical in many real-world traffic monitoring systems. Thus, it can be concluded that while dynamic speed thresholds offer theoretical advantages in certain contexts, the decision to use a fixed threshold in our delay calculations is rooted in the necessity to interpret sensor data reliably and effectively under real-world conditions where complete speed data may not always be accessible. The main advantage of this pragmatic approach is the balance between accuracy and feasibility in real-life conditions. Thus, it can be stated that A fixed threshold provides a simple and consistent criterion for defining when a vehicle is considered to be in a queue. Moreover, this consistency is crucial for maintaining uniformity in delay calculations across different traffic scenarios. In many real-world applications, sensor data may not provide precise speed measurements due to limitations in detection technology. Often, sensors can only categorize vehicles as either moving or stopped. By using a fixed threshold, we can work within these data limitations and still produce reliable estimates of queue lengths and delays. Thus, implementing a fixed threshold is straightforward and computationally efficient. This simplicity ensures that the system can operate in real-time without requiring complex calculations or extensive data processing. On the other side, implementing dynamic speed thresholds that adjust based on real-time traffic conditions, road types, and intersection characteristics might have some advantages. Dynamic thresholds can potentially enhance the accuracy of delay calculations by tailoring the threshold to the specific context. For example, dynamic thresholds can adapt to varying traffic densities and flow rates, providing a more responsive measure of when vehicles are queuing. For instance, during peak hours, the threshold can be adjusted to reflect higher congestion levels. Different road types (e.g., highways, arterial roads) and intersection configurations may require different thresholds to accurately capture queuing behavior. A dynamic system can account for these variations and adjust accordingly. However, implementing dynamic thresholds requires real-time data on traffic conditions and road characteristics, as well as more complex algorithms to adjust the thresholds. This approach may increase computational overhead and necessitate more sophisticated data processing capabilities. Thus, the adopted fixed threshold system aligns with the operational constraints and data availability typical in many real-world traffic monitoring systems.

For transit vehicles, we refrain from increasing the delay for transit vehicles during boarding and alighting at transit stops. The controller should only be penalized for delays attributable to its actions related to the traffic lights, and not due to dwelling at the bus stop. The adopted reward function is shown in Eq. (5). This reward function focuses on minimizing the overall person delay at the intersection and it is called the person-based mode. It should be noted that bus occupancy levels vary from one bus route to another and they typically vary among different buses serving the same route. Thus, it is important to train the RL agent on different levels of bus occupancies so that the agent is able to deal with variabilities in the occupancy level. As a result, the bus occupancy was assumed to follow a uniform distribution with two boundaries for the minimum bus occupancy (=1 person/bus) and the maximum bus occupancy (= bus capacity = 51 person/bus) ($X \sim Uniform\ (1, 51)$). It should be noted that the uniform distribution was selected for modeling the bus occupancy to make sure that the different levels of bus occupancy are equally visited by the eMARLIN-MM agents during the training phase, thus ensuring a robust model that can select appropriate decisions at the different bus occupancy levels. In the real-world field (at the implementation phase of this method), bus occupancy can be measured in real-time using Automated Passenger Counter (APC) technology which is widely used in the City of Toronto. It should be noted that the accuracy of APCs in estimating the bus occupancy has been improving over the years. In addition, the COVID-19 pandemic has accelerated this progress, as transit agencies needed to monitor and report bus occupancies in real time. Consequently, APC data has gradually become available in real time and with high accuracy. For the passenger vehicle occupancy, it was assumed at 1.25 person/vehicle which is the average occupancy of this type of vehicle in the City of Toronto in the AM peak period according to the

Transportation Association of Canada (2016) report. It should be noted that bus occupancy was assumed to follow a distribution while a fixed value was assumed for passenger vehicle occupancy because bus occupancy can be estimated in real-time using APC technology; however, there is no similar technology for passenger vehicles to automatically capture the occupancy. As such, it is hard to estimate passenger vehicle occupancy in real-time and thus a fixed value of 1.25 was assumed. In addition, passenger vehicle occupancy generally has a small value in comparison to bus occupancy, and thus it is generally more important to have exact information about the bus occupancy levels in comparison to vehicle occupancy. Thus, the investment required to obtain such data in real-time may not be economically justified. The resources needed for widespread deployment of advanced occupancy detection technologies might outweigh the benefits, especially when considering the relative impact of vehicle occupancy compared bus occupancy. Besides the person-based reward function shown in Eq. (5), a second reward function that focuses on minimizing the vehicle delay is tested without considering vehicle occupancy, which is the common approach in the literature that we depart from. In this case, it is called the vehicle-based model, and it deals with buses and cars similarly assuming that they both have the same weights regardless of their occupancy levels as shown in Eq. (6).

$$r_i = -D_{buses} - D_{cars} = -\sum_{x=1}^{X} O_{bus\,x}\, d_{bus_x} - \sum_{y=1}^{Y} O_{car\,y}\, d_{car\,y} \tag{5}$$

$$r_i = -D_{buses} - D_{cars} = -\sum_{x=1}^{X} d_{bus_x} - \sum_{y=1}^{Y} d_{car\,y} \tag{6}$$

Where: r_i is the reward value, D_{buses} is the total person-delays experienced by bus passengers for all buses in the intersection approaches, D_{cars} is the total person-delays experienced by car passengers for cars in all intersection approaches, $O_{bus\,x}$ is the occupancy of bus x in the intersection, d_{bus_x} is the delay of bus x and as actions are taken on a second-by-second basis this value equals to 1 when the bus speed is below the defined threshold and 0 when the bus speed is higher. $O_{car\,y}$ is the occupancy of car y in the intersection area, $d_{car\,y}$ is the delay of car y and as actions are taken on a second-by-second basis this value equals 1 when the bus speed is below the predefined threshold and 0 when the bus speed is higher.

4- Communication between the agents: the agents communicate by sharing information among neighboring intersections. Communication in our framework is restricted to a fixed-length information vector that agent i can receive from its neighboring agents $j \in N_i$ every single time step. Unlike the other MARL methods such as MARLIN (El-Tantawy et al., 2013), where observations and actions of neighbors are communicated, the proposed communication technique in this study does not impose constraints on the type of information shared between the agents. As long as this communication is between neighboring agents and maintains a fixed length, any relevant information can be shared among neighboring agents.

It should be noted that the adopted communication framework employs a fixed-length vector representation to facilitate efficient transmission of information among agents while conserving bandwidth to achieve computational efficiency. This approach involves mapping raw observations into an embedding space where each observation is represented by a vector of fixed dimensions. The choice of a low-dimensional embedding space aims to achieve a balance between information preservation and communication efficiency. Ideally, this embedding space should retain essential information from the original high-dimensional raw observation space. However, if too much information is lost during the embedding process, typically indicated by a reduction in performance metrics, adjustments can be made by increasing the dimensionality of the embedding space. Thus, the dimensionality of the embedding space was treated as a hyperparameter that requires tuning to optimize the trade-off between information sharing and communication efficiency. Increasing the dimensionality allows agents to preserve more detailed and nuanced information within their embedding vectors, potentially enhancing their ability to share richer and more informative messages; however, it increases the required computational power. It is important to note that while a fixed-length vector limits the amount of information that can be directly transmitted in each communication instance, the effectiveness of our approach hinges on the embedding's ability to encapsulate relevant features from the raw observations. This design choice ensures a balance between computational efficiency and the need for agents to exchange meaningful and actionable information. In addition, it should be noted that in the event neighboring agents fail to communicate, the performance of the eMARLIN-MM method would reduce to a performance similar to the independent agents (iDQN), losing the advantage and the benefits achieved from the communication between the agents.

In this study, it was assumed that the collective local observations at intersections can encapsulate the dynamics of the entire system. This assumption is grounded in several key considerations. Firstly, each intersection operates within a localized environment where the majority of interactions influencing traffic flow occur within a limited spatial and temporal vicinity (El-Tantawy et al., 2013; Wang et al., 2023a, 2023b). Thus, by aggregating local observations, it is possible to capture the essential dynamics affecting both immediate traffic and interactions with neighboring intersections. Secondly, the proposed eMARLIN-MM framework incorporates communication between control agents, enabling the sharing of pertinent information among neighboring intersections. This inter-agent communication ensures that local decisions are informed by the broader context, thereby bridging the gap between localized observations and system-wide dynamics.

5- Hyperparameters: the main hyperparameters used in this study are summarized in Table 1. It should be noted that in this study we propose a multimodal extension of the novel approach termed embedding communicated Multi-Agent Reinforcement Learning for Integrated Networks - eMARLIN (Wang et al., 2023a, 2023b). Thus, further details about the hyper parameters can be found in Wang et al. (2023a, 2023b) study.

Table 1
Summary of the main hyperparameters used in this study.

Exploration: epsilon-greedy		
Initial epsilon		1
Final epsilon		0.01
Training pipeline		
DQN target network update frequency		2k
Batch size		256
Buffer size		2M
Neural Networks		
eMARLIN-MM Encoder	Layers for flatten observations	64, 64
	CNN for transit observations	one layer with shape [1, obs_dim], 8 filters Where: obs_dim= transit observations dimension
Embedding size		32
eMARLIN-MM Executor	Layers sizes	64, 32

3.3. Experimental setup and case study

In this study, Aimsun Next was used to develop a simulation model for the area around the Yonge and Steeles intersection which is one of the major intersections in Toronto, Canada. Specifically, this intersection is located at the boundary between the City of Toronto and York Region. Fig. 2 shows the signalized intersections in the study area which are used for testing the proposed eMARLIN-MM algorithm. This area was selected as it experiences heavy traffic and carries high volumes of transit passengers every weekday (TTC, 2020). Due to its location along the northern boundary of the City of Toronto, this simulation area around the Yonge-Steeles intersection is utilized by both the Toronto Transit Commission (TTC) and York Region Transit (YRT) bus routes. The simulation model starts at Finch Station in the south which serves as a transit hub for all the YRT routes. In addition, this area is considered critical to the TTC as it serves major bus routes. As a result, in 2019, the TTC launched the RapidTO program, primarily focused on improving transit service performance along five of the busiest corridors serving both transit and traffic commuters. Steeles Avenue West was one of these corridors including the area modeled in this study. As a result, the use of an intelligent traffic signal controller such as eMARLIN-MM can be an innovative strategy which benefits both transit and traffic. While the RapidTO highlighted Steeles Avenue West as one major corridor where both transit and traffic suffer from major delays, no changes were made in the corridor to solve this issue and reduce the delays in this area. Fig. 3 shows the different transit routes that service the simulation area for both transit agencies (the TTC and YRT) and their direction. The simulated area includes eight signalized intersections that can be classified into two categories. The first category, called major-major, includes intersections of two major arterials; the Yonge and Steeles intersection is the only intersection in this study that belongs to this category. The second category is called major-minor intersections, representing the intersection between a major arterial and a minor road; the remaining four intersections belong to this category. In addition, the distances between any two signalized intersections in this model vary between 150 and 450 m. The city's signal timing plans, obtained from the city of Toronto, adhere to the standard NEMA phasing diagram with semi-actuated control (Holm et al., 2007). Under these plans, major through phases have fixed durations and cannot be skipped, whereas minor through phases and all phases involving a protected left turn are callable and extendable via loop detectors. When evaluating eMARLIN-MM, five intersections along Yonge Street (a north–south corridor) are controlled by the eMARLIN-MM agents, while the remaining three intersections follow the city's signal timing plans as shown in Fig. 4. The phasing scheme employed by the RL agents is a CVPS, wherein a phase transition is permitted only if it aligns with the possibilities outlined in the city plan. The model was developed for the AM peak period starting from 6 to 10 am. The demand is calibrated through several steps. Initially, the 2016 Transportation Tomorrow Survey (TTS) data (Data Management Group, 2018), comprising origin–destination trips between different traffic analysis zones, is utilized to generate demand for a larger hybrid mesoscopic-microscopic model of the Greater Toronto and Hamilton Area (GTHA). Subsequently, the TTS traffic analysis zones intersecting our smaller subnetwork are kept as origins and destinations. Each incoming link into the subnetwork is assigned a new origin, while each outgoing link is assigned a new destination. Then, vehicle counts are accumulated in 15-min intervals from a simulation of the larger GTHA network. The resulting origin–destination counts are further adjusted using techniques outlined in the 'Integrating Macro, Meso, Micro, and Hybrid Simulations' tool in Aimsun (Aimsun, 2022). Then, this adjustment process utilizes turning movement counts at the subnetwork intersections, which are publicly available from the City of Toronto and accumulated in 15-minute intervals to calibrate the model to be in line with the real-world traffic demand pattern and volumes in 2019 (the pre-COVID conditions). The turning movement counts of the final calibrated demand closely align with the city's turning movement counts, with a coefficient of determination (R^2) equal to 0.933.

3.4. Public transit modeling

As mentioned earlier, the simulation area is located at the boundary between the City of Toronto and York Region and accommodates both TTC and YRT bus routes. Specifically, this area is served by four high frequency TTC bus routes and a total of 16 YRT

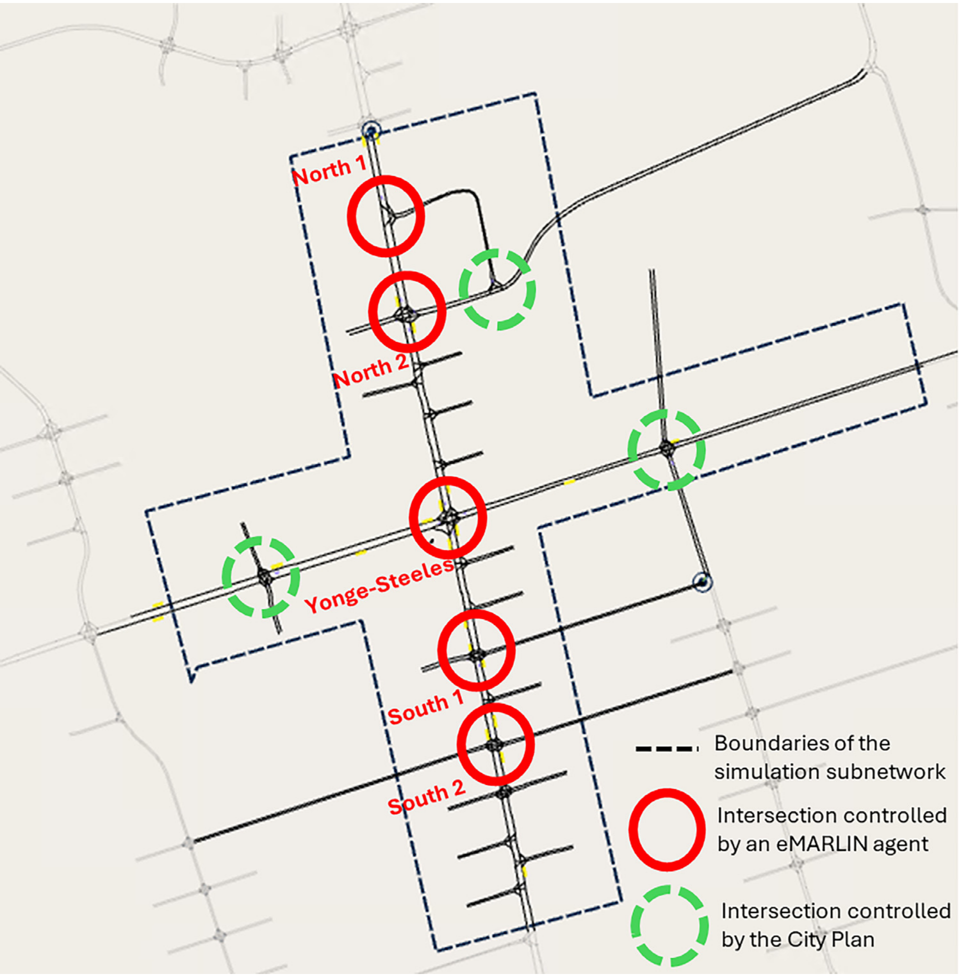

North 1
North 2
Yonge-Steeles
South 1
South 2
Boundaries of the simulation subnetwork
Intersection controlled by an eMARLIN agent
Intersection controlled by the City Plan

Fig. 2. Simulation area subnetwork as a part of the bigger network model showing the different intersections and differentiating the intersections controlled by the eMARLIN-MM agents and the intersections that follows the City of Toronto NEMA plan.

bus routes (ranging between low and high frequency). These routes were modeled in the simulation model based on their published schedules and headways by both the TTC and YRT. Table 2 summarizes the headways of the different TTC and YRT routes. For YRT, some routes operate solely in the PM peak period while others operate in the AM peak period. Since this study is focused on the AM peak period, only routes that operate in the AM peak were considered in the simulation model. The table shows that the combined headway of all TTC routes is 1.9 min, while the combined headway for YRT routes is 1.8 min. In addition, it should be noted that both the TTC and YRT bus routes share the section from Finch station south of the model to the Yonge and Steeles intersection. As a result, the combined headway of both TTC and YRT in this section is 0.92 min. These combined headways clearly show the large number of buses that utilize this area in the AM peak period and thus the need for some level of priority that facilitates the movements of these buses at the intersection based on their occupancy levels, highlighting the importance of using intelligent traffic signal controllers such as eMARLIN-MM.

For the dwell times, information about the exact dwell times of the different bus routes was not accessible. Thus, it is assumed that the dwell time at the bus stops in the model follows a normal distribution with an average value of 20 s and a standard deviation of 6.8 s ($X \sim N\ (20,\ 6.8)$). This distribution was adopted from earlier studies conducted in the City of Toronto for predicting the bus dwell and traveling times (Shalaby and Farhan, 2004; Milkovits, 2008; Miller et al., 2018). For the occupancy, it was assumed that passenger vehicle occupancy is 1.25 person/vehicle which is the average occupancy of this type of vehicle in the City of Toronto in the AM peak period according to the Transportation Association of Canada (2016) report. For buses, the occupancy level should vary across different bus routes and different locations. However, this data was not accessible. Thus, it is important to train the RL agent on different levels of bus occupancies so that the agent is able to deal with variabilities in the occupancy level. As a result, the bus occupancy was assumed to follow a uniform distribution with two boundaries for the minimum bus occupancy (=1 person/bus) and the maximum bus occupancy (= bus capacity = 51 person/bus) ($X\ \sim\ Uniform\ (1, 51)$). The bus occupancy changes at every bus stop by generating a random number and following the uniform distribution defined for the bus occupancy ($X\ \sim\ Uniform\ (1, 51)$).

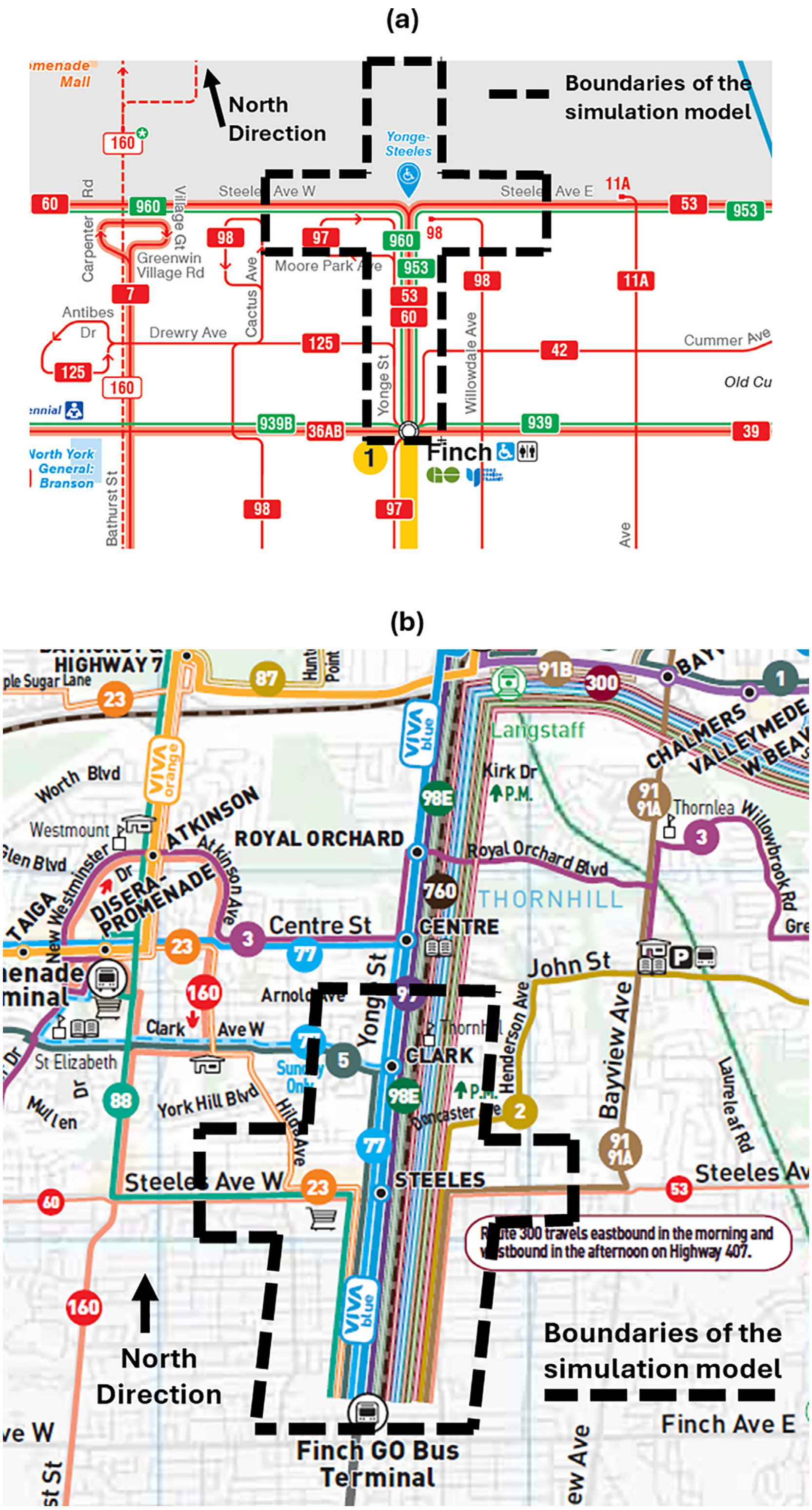


Fig. 3. Different transit routes operating in the simulation area for: (a) the TTC (it should be noted that the northern part of the model does not appear in the figure as this area is not served by the TTC) and (b) YRT.

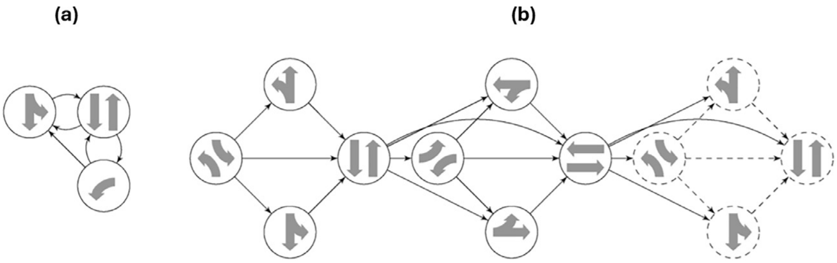


Fig. 4. CVPS used across the different intersections in the simulation area: (a) represent the phasing scheme adopted for the three-legged intersection (North 2) in the major-minor category. (b) represent the phasing scheme used for Yonge-Steeles intersection. The leftmost four phases are duplicated on the right (represented by dashed nodes) to enhance clarity in graph depiction. The phasing scheme for the remaining intersections (North 1, South1, and South 2) are similar except for the East-West phase for the minor road which becomes a permitted phase that needs to be called (indicating that this phase can be skipped). It should be noted that both left and right turns are permitted on the through movement for the different schemes.

Table 2
Summary of the headways of both the TTC and YRT routes in the working days AM peak period.

(a) TTC Routes

Route	Headway (min)	Frequency (bus/h)
53	6	10
60	8	7.5
953	11	5.45
960	7	8.57
Combined headway =1.9 min		Combined Freq= 31.6

(b) YRT Routes

Route	Headway (min)	Frequency (bus/hr)
88	16	3.75
23	21	2.85
5	22	2.72
77	15	4
VIVA Blue	7	8.57
99	32	1.875
98 E	PM PEAK only	–
760	Weekends	–
300	15	4
301	PM PEAK only	–
302	PM PEAK only	–
303	PM PEAK only	–
304	PM PEAK only	–
305	PM PEAK only	–
2	21	2.85
91	22	2.72
Combined headway =1.8 min		Combined Freq= 33.37

(c) Combined (TTC+YRT)

Combined Frequency (bus/h)	64.97
Combined Headway (min)	0.924

It should be noted that the uniform distribution was selected to model the bus occupancy rate to make sure that the different levels of bus occupancy are equally visited by the eMARLIN-MM agents during the training phase and thus have a robust model that can select the appropriate decision at the different bus occupancy levels. In the field (during the implementation of this method), bus occupancy can be estimated in real-time using Automated Passenger Counter (APC) technology which is widely used in the City of Toronto. For passenger vehicle occupancy, it was assumed that passenger vehicle occupancy is 1.25 person/vehicle which is the average occupancy of this type of vehicle in the City of Toronto in the AM peak period according to the Transportation Association of Canada (2016) report. It should be noted that bus occupancy was assumed to follow a distribution while a fixed value was assumed for passenger vehicle occupancy because bus occupancy can be estimated in real-time using APC technology; however, there is no similar technology for passenger vehicles to automatically capture the occupancy. Thus, it is hard to estimate passenger vehicle occupancy in real-time and thus a fixed value of 1.25 was assumed. In addition, passenger vehicle occupancy generally has a small value in comparison to bus occupancy and thus it is generally more important to have exact information about the bus occupancy levels in comparison to vehicle occupancy.

3.5. Tested scenarios

In this study, four different types of MARL methods were tested and compared. In addition, the performance of these MARL methods is benchmarked against the baseline (i.e. the existing signal timing plan) and traditional techniques. Overall, the impacts of the following traffic signal controllers on (traffic, transit, and the entire corridor) were tested and compared:

- City plan: this traffic signal controller follows the standard dual-ring NEMA scheme that is used by the city.
- Unconditional TSP (Unc-TSP): this controller can extend the green phase to facilitate the bus movement through the intersection. This is the most common TSP controller used in the City of Toronto (Hu, 2022). If a bus is detected in the control area, the controller extends the green phase by 2 s and checks whether the bus has cleared the intersection or not to switch to the next phase. If the bus cannot clear the intersection during this 2 s period, an additional 2 s of green extension is provided and so on until the bus clears the intersection or the phase reaches the maximum allowable green phase. It should be noted that this controller cannot deal with different buses from different branches but only considers and provides priority to the first bus that enters the intersection control area and ignores any other bus that enters the intersection area during this priority call.
- Independent Deep Q-Network (iDQN) Vehicle-Based controller (iDQN-VB): the controller in this case consists of separate DQN agents (one DQN agent for each intersection) and these agents do not share any information or observations. In addition, the agents were modeled to maximize vehicular traffic performance. In other words, the reward function is set to minimize the total vehicle delay using Eq. (6), and in this case buses were treated similar to passenger vehicles (ignoring the bus occupancy).
- iDQN Person-Based controller (iDQN-PB): it is similar to the (iDQN-VB) except for the objective or the reward function. In this case, the main objective is to maximize the person throughput through the intersection using reward function (5) which focuses on minimizing the overall person delay at the intersection.
- eMARLIN-MM-VB controller: this is the eMARLIN-MM controller discussed earlier in detail, and in this case neighboring agents share information to maximize the total vehicle throughput through the intersection (ignoring the occupancy of the buses) using the reward function presented in Eq. (6).
- eMARLIN-MM-PB controller: similar to the eMARLIN-MM-VB with one exception which is the objective of maximizing the total person throughput using the reward function shown in Eq. (5).

4. Analysis and results

This section focuses on comparing the performance of the vehicle-based and person-based eMARLIN-MM methods with the other methods mentioned earlier (iDQN, pre-timed, and unconditional TSP). Firstly, the learning stability of the different MARL methods are examined in Fig. 5 which depicts the average global reward over the different learning episodes or iterations. The curves clearly show that the four methods achieved convergence. In addition, the methods were separated for person-based in Fig. 5(a) and vehicle-based methods in Fig. 5(b) because the two reward functions are different and thus the scale needed to cover the range of average reward values varies between the two methods. Secondly, to compare the performance of the different methods, Figs. 6 and 7 were developed to show the person and vehicle delays by mode at the different intersections. In addition, the figures show the overall person and vehicle delays along the entire corridor (encompassing the five intersections). The results show that the MARL methods (both iDQN and eMARLIN-MM) improve the conditions for both transit and traffic when compared to the two benchmarks (pre-timed and unconditional TSP) regardless of the logic used for optimization (vehicle-based or person-based). In addition, both iDQN and eMARLIN-MM can reduce the total person and vehicle delays for the different types of intersections (major-major and major-minor). However, the biggest impact can be observed at the major intersection of Yonge and Steeles. For this intersection, both iDQN and eMARLIN-MM can achieve 40–60 % improvement in the total person and vehicle delays for the general traffic across the different intersections. Vehicle-based MARL methods achieve the lowest person and vehicle delays for the general traffic as shown in Figs. 6(a) and 7(a) while person-based MARL methods achieve the lowest person and vehicle delays for public transit as shown in Figs. 6(b) and 7(b). Finally, for the total delays (transit + traffic), vehicle-based methods achieve the lowest total delays while person-based methods achieve the lowest total person delays as summarized in Figs. 6(c) and 7(c). This is because of the differences in the reward functions adopted for each method as the reward function used for vehicle-based methods focuses on minimizing the total vehicle (traffic + transit) delays while the reward function adopted for person-based methods focuses on minimizing the total person delays (for both traffic and transit commuters based on their occupancy levels).

Furthermore, the results show that eMARLIN-MM is achieving better performance for traffic, transit, and the entire corridor when compared with iDQN for both the vehicle-based and person-based methods. To better understand the impact of the different methods, Table 3 was developed to show the percentage change in the total person and vehicle delays of the different tested methods relative to the pre-timed case at all intersections. For the entire corridor (transit + traffic for the five intersections), vehicle-based methods achieve the lowest total vehicle delays with −47 and −54 % reductions achieved by iDQN and eMARLIN-MM compared to pre-timed signal plan. On the other hand, person-based methods achieve the lowest total person delays with −46 and −51 % reduction achieved by iDQN and eMARLIN-MM compared to pre-timed signal plan. It is clear that eMARLIN-MM outperforms the iDQN method achieving lower delays for both the vehicle-based and person-based methods. While the previous discussion focused on the total delays for the entire corridor, the same results hold for the five intersections showing that eMARLIN-MM outperforms the iDQN method for the

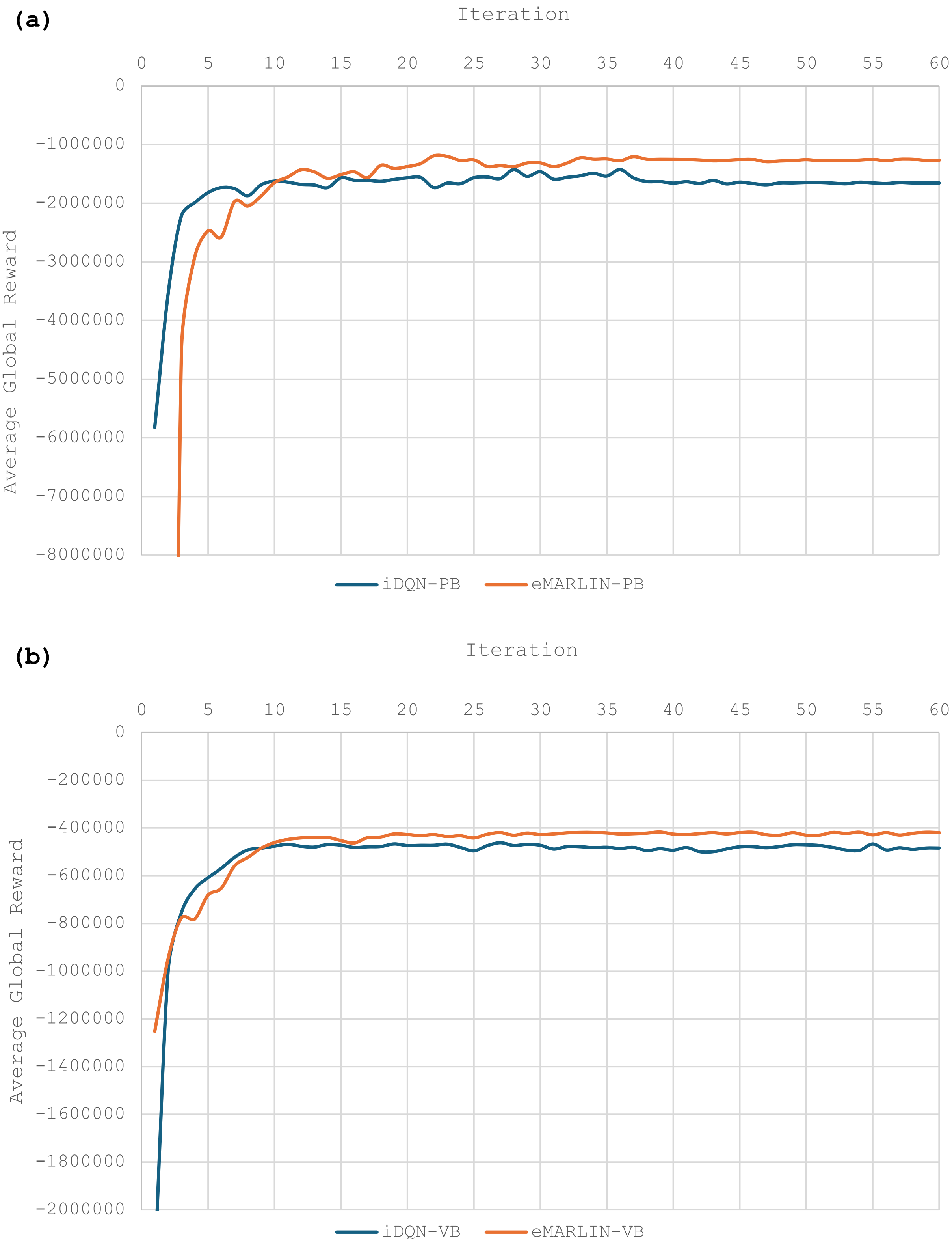


Fig. 5. Learning curves for the different methods tested: (a) for person-based methods and (b) for vehicle-based methods.

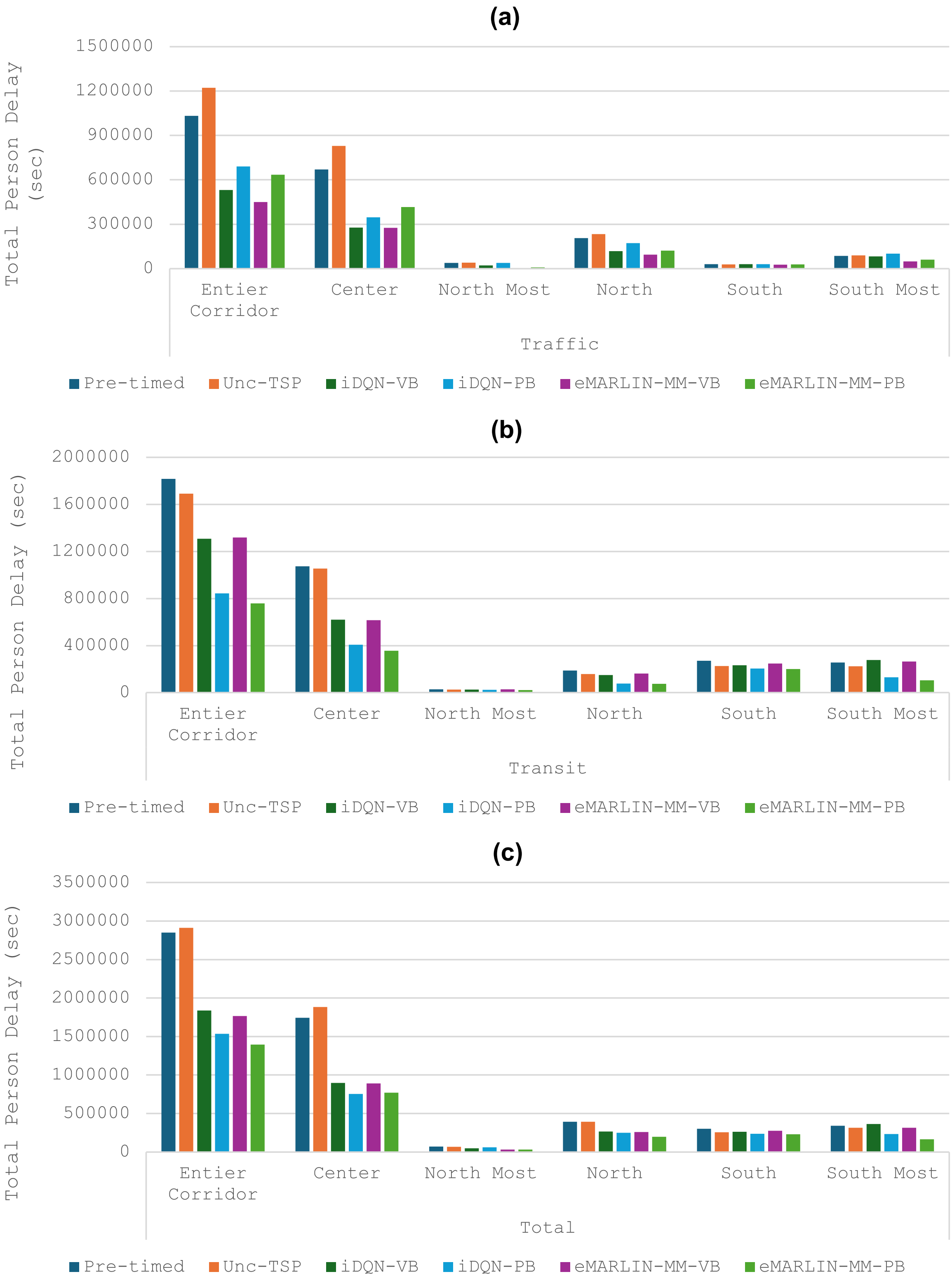


Fig. 6. Total person delay across the different intersections for: (a) the general traffic, (b) transit users, (c) overall for the two modes (transit + traffic).

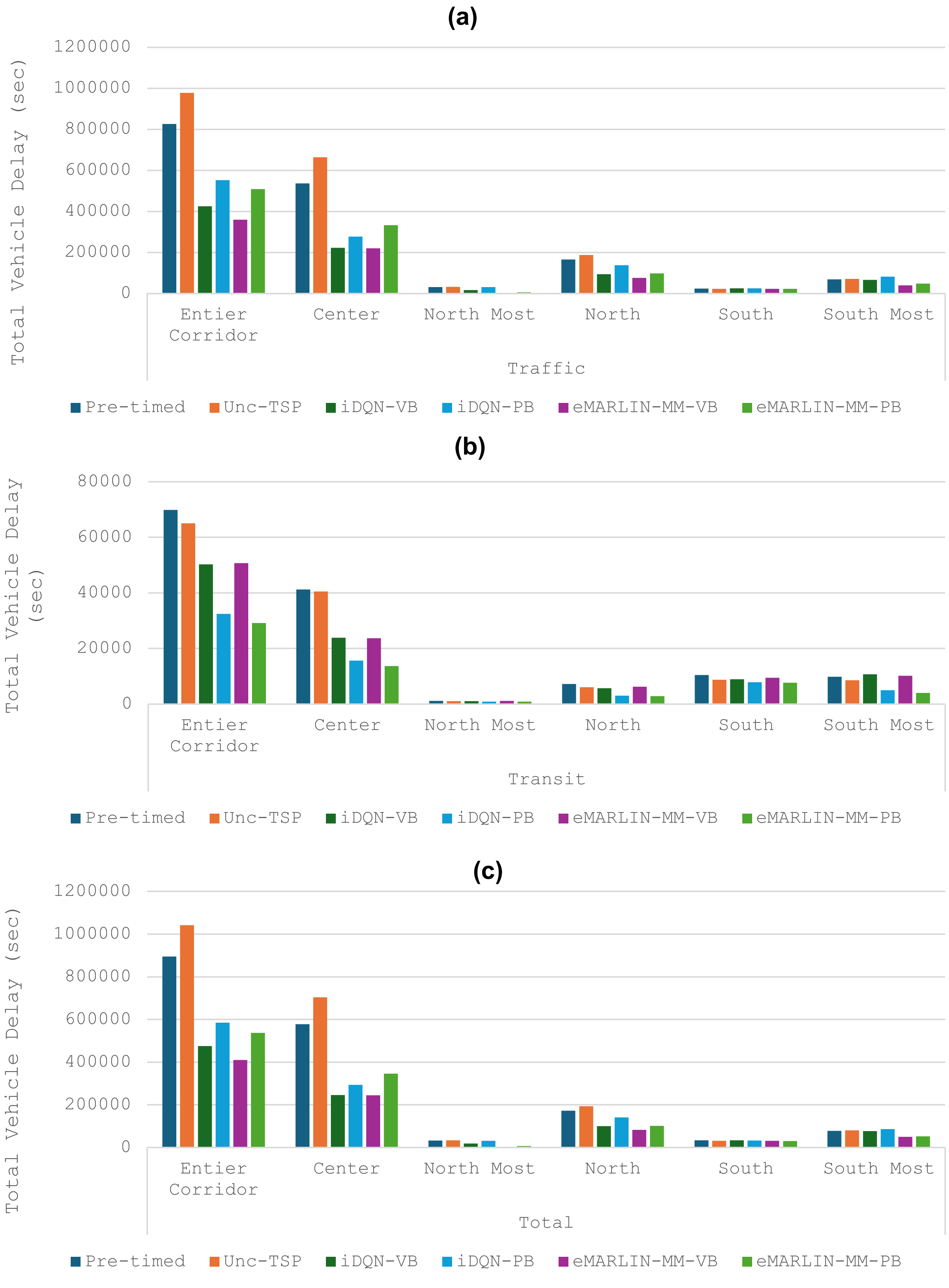


Fig. 7. Total vehicle delay across the different intersections for: (a) the general traffic, (b) transit, (c) overall for the two modes (transit + traffic).

Table 3
Percentage of change: (a) in the total person (traffic + transit) and (b) total vehicle (traffic + transit) delays for the different methods tested across the different intersections compared to the pre-timed scenario (when the bus occupancy follows a uniform distribution).

(a)

Method	% Change in Person Delay					
	Entire corridor	Center	North Most	North	South	South Most
Pre-timed (City Plan)	Reference					
iDQN-PB	−46	−57	−10	−37	−21	−32
iDQN-VB	−35	−48	−29	−32	−12	6
eMARLIN-MM-PB	−51	−56	−57	−50	−24	−51
eMARLIN-MM-VB	−38	−49	−55	−35	−9	−8
Unc-TSP	2	8	−8	−4	−15	−8

(b)

Method	% Change in the Vehicle Delay					
	Entire corridor	Center	North Most	North	South	South Most
Pre-timed (City Plan)	Reference					
iDQN-PB	−35	−49	−3	−18	−4	11
iDQN-VB	−47	−57	−44	−42	−2	−1
eMARLIN-MM-PB	−40	−40	−79	−42	−10	−33
eMARLIN-MM-VB	−54	−58	−93	−52	−8	−36
Unc-TSP	16	22	2	12	−8	2

different intersections for both the vehicle-based and person-based delays at the different types of intersections (major-major or major-minor).

Finally, the results of the unconditional TSP show that this method results in increasing the delay for passenger cars (5–24 % increase when compared to pre-timed signal plan) especially at the center intersection (Yonge and Steeles) resulting in 23 % increase. On the other hand, this method can improve transit performance only in major-minor intersections (6–16 % decrease when compared to pre-timed signal plan) with minor impact on major-major intersections (only 1.7 % improvement). For the major-major intersection, transit is heavily utilizing the four approaches to service both the TTC and YRT buses. Thus, providing unconditional priority to once approach, while ignoring the other approaches, results in additional delays for buses travelling in the other approach and thus the impact of the unconditional TSP on transit service is minor for this intersection. In addition, unconditional priority results in additional delays for the general traffic using the other approach which is serving high traffic demand and thus results in major increases in the delays for this approach. On the other hand, for the major-minor intersections, transit travels in one direction and thus giving priority to this direction would reduce transit delays but at the cost of additional delays to traffic on the other approach. As a result, the overall impact of this method (on both transit + traffic) is limited resulting in an increase in the total vehicle delay of up to 16 % for the entire corridor (five intersections) and 2 % increase in the total person delays. For this major-major intersection, this method results in 8 % and 22 % increase in the total person and vehicle delays (transit and traffic), indicating that it is making the conditions worse than pre-timed signal when considering person or vehicle delays, and thus it is not recommended to implement unconditional TSP methods in intersections between two major roads.

5. Impact of bus occupancy on the efficiency of the different methods

While the previous discussion focused on comparing the different methods at various levels of traffic demand represented by the intersecting roads major-major (for intersections serving heavy traffic demands from all four approaches) and major-minor (for intersections serving heavy demand in one direction and low demand in the other), it is also important to analyze and understand the performance of the different methods at different levels of bus occupancy or transit demand. It should be noted that the previous analysis was based on the assumption that bus occupancy follows a uniform distribution that varies at each bus stop. In this section, the performance of the different methods is tested at two levels of bus occupancy: low and high (the two extremes). The bus occupancy is assumed to be a constant value that does not vary along the bus route. For the low bus occupancy case, it is assumed that the occupancy of the bus is identical to the occupancy of the passenger vehicle= 1.25 passenger/ bus. On the other hand, the second extreme is assuming that all the buses are running full. In this case, it is assumed that the bus occupancy equals 51 passenger/ bus, which is the capacity of the standard 12-m buses used by both the TTC and YRT. Tables 4 and 5 show the total person and vehicle delays (transit+ traffic) for the different methods tested across the different intersections compared to the pre-timed scenario for the low and high occupancy scenarios.

For the case of low occupancy, the results show that both vehicle-based and person-based methods converge and achieve the same results for both the iDQN and eMARLIN-MM methods. In addition, the changes in the total person delays and vehicle person delays are similar across the different intersections for the different methods. In this case, the reward function used for person-based methods transforms to the vehicle-based method because this reward function assigns weights to the different modes based on their occupancy as shown in Eq. (5). For the case of low occupancy, both buses and passenger vehicles have the same occupancy levels

Table 4
Percentage of change: (a) in the total person (traffic + transit) and (b) total vehicle (traffic + transit) delays for the different methods tested across the different intersections compared to the pre-timed scenario (when the bus occupancy = 1.25 passenger/bus = passenger car occupancy).

(a)

Method	% Change in Person Delay					
	Entire corridor	Center	North Most	North	South	South Most
Pre-timed (City Plan)	Reference					
iDQN-PB	−47	−57	−44	−42	−2	−1
iDQN-VB	−47	−57	−44	−42	−2	−1
eMARLIN-MM-PB	−54	−58	−93	−52	−8	−36
eMARLIN-MM-VB	−54	−58	−93	−52	−8	−36
Unc-TSP	16	22	2	12	−8	2

(b)

Method	% Change in the Vehicle Delay					
	Entire corridor	Center	North Most	North	South	South Most
Pre-timed (City Plan)	Reference					
iDQN-PB	−47	−57	−44	−42	−2	−1
iDQN-VB	−47	−57	−44	−42	−2	−1
eMARLIN-MM-PB	−54	−58	−93	−52	−8	−36
eMARLIN-MM-VB	−54	−58	−93	−52	−8	−36
Unc-TSP	16	22	2	12	−8	2

Table 5
Percentage of change: (a) in the total person (traffic + transit) and (b) total vehicle (traffic + transit) delays for the different methods tested across the different intersections compared to the pre-timed scenario (when the bus occupancy = 51 passenger/bus = bus capacity).

(a)

Method	% Change in Person Delay					
	Entire corridor	Center	North Most	North	South	South Most
Pre-timed (City Plan)	Reference					
iDQN-PB	−54	−51	−33	−39	−52	−53
iDQN-VB	−32	−47	−23	−29	−14	8
eMARLIN-MM-PB	−66	−68	−80	−38	−62	−72
eMARLIN-MM-VB	−34	−47	−39	−27	−8	−3
Unc-TSP	−1	4	−3	−5	−16	−10

(b)

Method	% Change in the Vehicle Delay					
	Entire corridor	Center	North Most	North	South	South Most
Pre-timed (City Plan)	Reference					
iDQN-PB	−19	−8	18	33	88	−52
iDQN-VB	−47	−57	−44	−42	−2	−2
eMARLIN-MM-PB	−28	−16	−92	−3	−22	−64
eMARLIN-MM-VB	−54	−58	−93	−52	−7	−37
Unc-TSP	16	22	2	12	−8	2

(O_{both}) and thus Eq. (5) transforms to Eq. (7). In this case, person-based methods focus on optimizing for a constant which is the (O_{both}) multiplied by the total vehicle delay which is the same as optimizing for the total vehicle delay equation shown in Eq. (6). Thus, in this case, the two reward functions used for person-based and vehicle-based methods are identical and the two methods will achieve the same performance. In addition, the changes in both the person and vehicle delays are similar for the different methods when compared to the pre-timed signal plan. In this case, the total person delays for transit, traffic and both are simply the multiplication of 1.25 by the vehicle delays for the different modes and the overall for the summation of the two modes. As a result, the percentage changes in the delays when compared to the pre-timed case are identical for the different methods for both the person and vehicle delays. Moreover, the results show that eMARLIN-MM can achieve 54 % reduction in the delays for the entire corridor, while iDQN can achieve 47 % reduction. These results show that eMARLIN-MM outperforms iDQN and achieves lower delays.

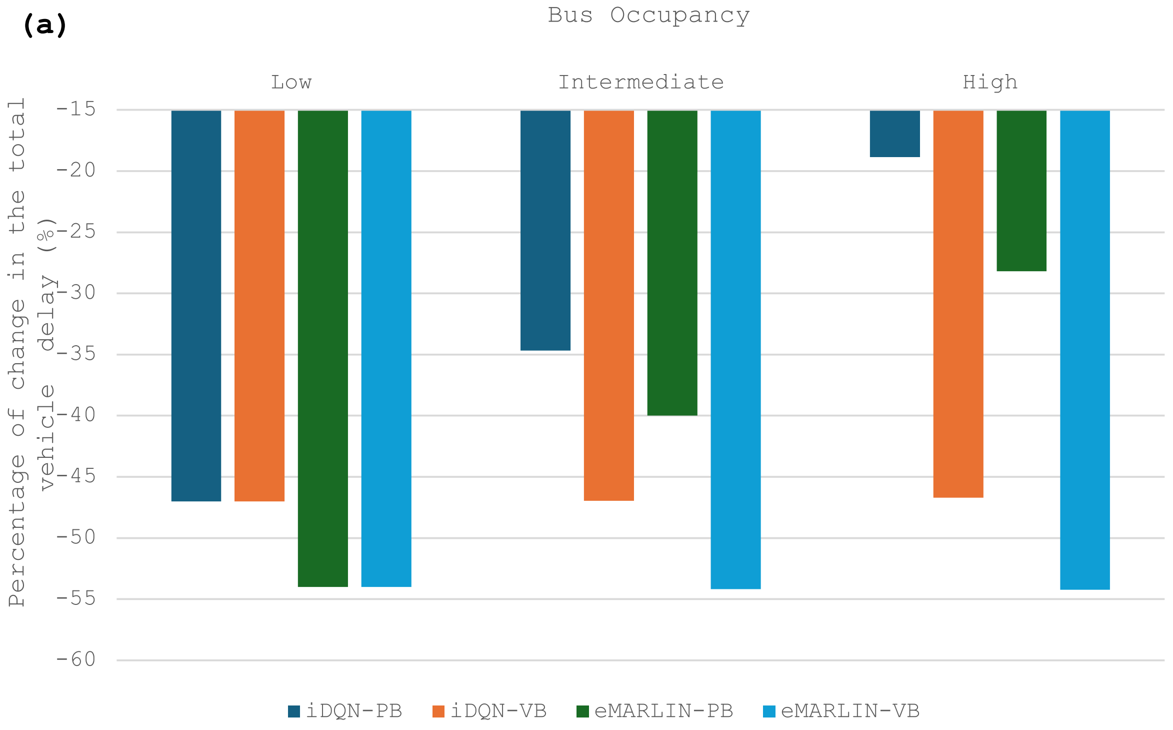
$$r_i = -\sum_{x=1}^{X} O_{both}\, d_{bus_x} - \sum_{y=1}^{Y} O_{both}\, d_{car\,y} = O_{both}\left[-\sum_{x=1}^{X} d_{bus_x} - \sum_{y=1}^{Y} d_{car\,y}\right] \tag{7}$$

For the case of high occupancy, the results show that iDQN methods can achieve −54 to −32 % reduction in the total person delays and −19 % to −47 % reduction in the vehicle delays. On the other hand, eMARLIN-MM methods achieve −34 to −66 % reduction in the total person delays and −25 to −54 % reduction in the total vehicle delays. These results show that the use of person-based methods can significantly minimize the total person delays (−54 and −66 % for iDQN and eMARLIN-MM) when compared to the vehicle-based methods (−32 % and −34 %) as a result it is highly recommended to use person-based methods especially when the occupancy of the buses is high to maximize the person throughout from the intersection. In addition, the results show that eMARLIN-MM consistently outperforms the iDQN method and thus it is recommended to use eMARLIN-MM to achieve the best performance by coordinating the different agents.

Finally, to evaluate the performance of the different methods at the different levels of bus occupancy, Fig. 8 was developed to show the percentage change in the total vehicle and person delays for the different methods for the entire corridor (traffic + transit) compared to the pre-timed scenario. It should be noted that the figure shows three levels of occupancy: low, intermediate, and high. The low and high bus occupancies are the same as defined earlier while the intermediate level represents the case of a constant bus occupancy of 26 passengers/bus. The results show that the performance of the vehicle-based methods, for the total vehicle delay, remains consistent across the different levels of bus occupancy as shown in Fig. 8(a) because this method is not affected by the bus occupancy (percentage change in the total vehicle delays = −54 and −47 % for the eMARLIN-MM and iDQN methods). On the other hand, the ability of the person-based methods to achieve lower vehicle delays decreases with the increase in the bus occupancy rates. In addition, eMARLIN outperforms iDQN across the different levels of bus occupancies and for both vehicle-based and person-based methods. For the total person delays shown in Fig. 3(b), the results show that the ability of the vehicle-based methods to achieve better total person delays decreases with the increase in the bus occupancy moving from (−54 % and −47 % for eMARLIN-MM and iDQN at low bus occupancies to −34 % and −32 % at high bus occupancies). On the other hand, the percentage change in the total person delays for person-based methods remains consistent from low to intermediate bus occupancy levels (−47 % and −53 % for iDQN and eMARLIN-MM) because in this case the agent might provide priority to buses that has low levels of occupancy and thus reduce transit delays while increasing passenger vehicle delays and the overall increase and decrease in the delays for transit and traffic offset each other, resulting in constant performance in this region (from low to intermediate bus occupancies). However, from intermediate to high bus occupancy levels, person-based methods can achieve substantial improvements in the total person delays jumping from −47 % and −52 % for iDQN and eMARLIN-MM at intermediate bus occupancy to −55 % and −66 % for iDQN and eMARLIN-MM at high bus occupancy levels. Thus, while person-based methods (when compared to vehicle-based methods) achieve higher vehicle delays with the increase in the bus occupancy, they achieve substantially lower person delays indicating that it prioritizes buses with higher occupancy rates at the expense of traffic to maximize the person throughput in the network. Finally, these results clearly show that eMARLIN-MM outperforms iDQN across the different bus occupancy levels and for the two methods tested (person-based and vehicle-based).

Finally, in order to capture the relationship between the level of occupancy and the performance of the different RL algorithms, the performance (in terms of the percentage of change in the total person travel times compared to the case of fixed traffic signals) of the four RL algorithms was tested at seven different levels of bus occupancy as follows: 1.25, 9, 18, 26, 34, 42, and 51 passengers. Fig. 9 was developed to visualize the relationship between the level of bus occupancy in the x-axis and the percentage of change in the total vehicle (Fig. 9-a) or person (Fig. 9-b) delays in the y-axis. Fig. 9-a clearly shows that vehicle delays for vehicle-based methods remain consistent across the different levels of bus occupancy indicating that this method is not sensitive to the variability in the bus occupancy. On the other hand, vehicle delays for person-based methods decrease with the increase in the bus occupancy, indicating that these methods might achieve higher vehicle delays to achieve another goal. In addition, at low bus occupancy levels, both vehicle-based and person-based methods achieve the same performance. Furthermore, the figure shows the superiority of the eMARLIN methods over the independent methods (iDQN) across the different scenarios. Person delays for the different RL methods across the different bus occupancy levels are presented in Fig. 9-b. This figure shows that the efficiency of vehicle-based methods in improving person delays decreases with the increase in the bus occupancy levels. On the other hand, the figure shows the superiority of person-based methods in improving person delays across the different levels of bus occupancy. The figure shows that the percentage change in the total person delays for person-based methods remains consistent from low to intermediate bus occupancy levels because in this case the agent might provide priority to buses that has low levels of occupancy and thus reduce transit delays while increasing passenger vehicle delays and the overall increase and decrease in the delays for transit and traffic offset each other, resulting in constant performance in this region (from low to intermediate bus occupancies). However, from intermediate to high bus occupancy levels, person-based methods can achieve substantial improvements in the total person delays. Thus, it can be stated that while person-based methods (when compared to vehicle-based methods) achieve higher vehicle delays with the increase in the bus occupancy, as they achieve substantially lower person delays indicating that it prioritizes buses with higher occupancy rates at the expense of traffic to maximize the person throughput in the network. Finally, these results clearly show that eMARLIN-MM outperforms iDQN across the different bus occupancy levels and for the two methods tested (person-based and vehicle-based). In addition, the results show that both vehicle-based and person-based methods achieve the same performance at low levels of bus occupancy.

These results clearly show the importance of bus occupancy data for the efficient implementation of the person-based approach. In general, failure to communicate bus occupancy data would result in deteriorating the performance of the person-based methods, and the RL agent would consider buses as passenger vehicles similar to vehicle-based methods. In other words, without bus occupancy data, the system's performance will degrade and revert to a less efficient vehicle-based approach, where decisions are made based on the number of vehicles rather than the number of passengers. Therefore, bus occupancy data are critical for the efficient implementation of this method. It should also be noted that bus occupancy information is currently available in real-time during bus operations in the City of Toronto, given that most of the TTC buses are equipped with automated passenger counters that are able to get information



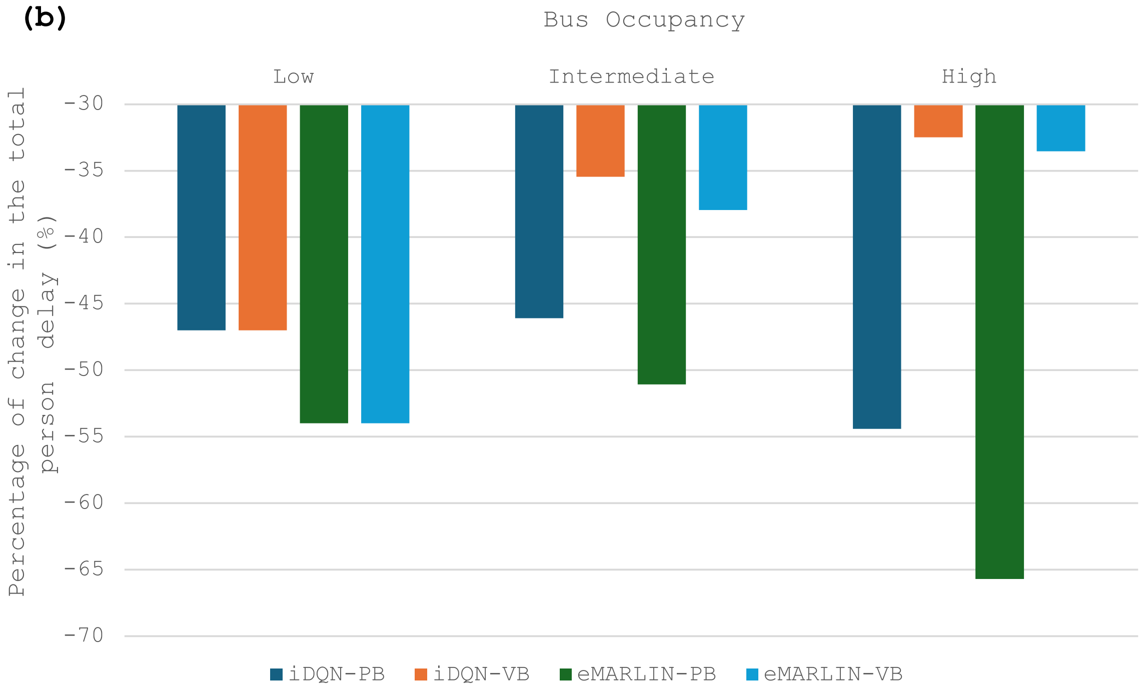


Fig. 8. Percentage of change: (a) in the total vehicle (traffic + transit) and (b) total person (traffic + transit) delays for the different methods tested for the entire corridor at the different levels of bus occupancies.

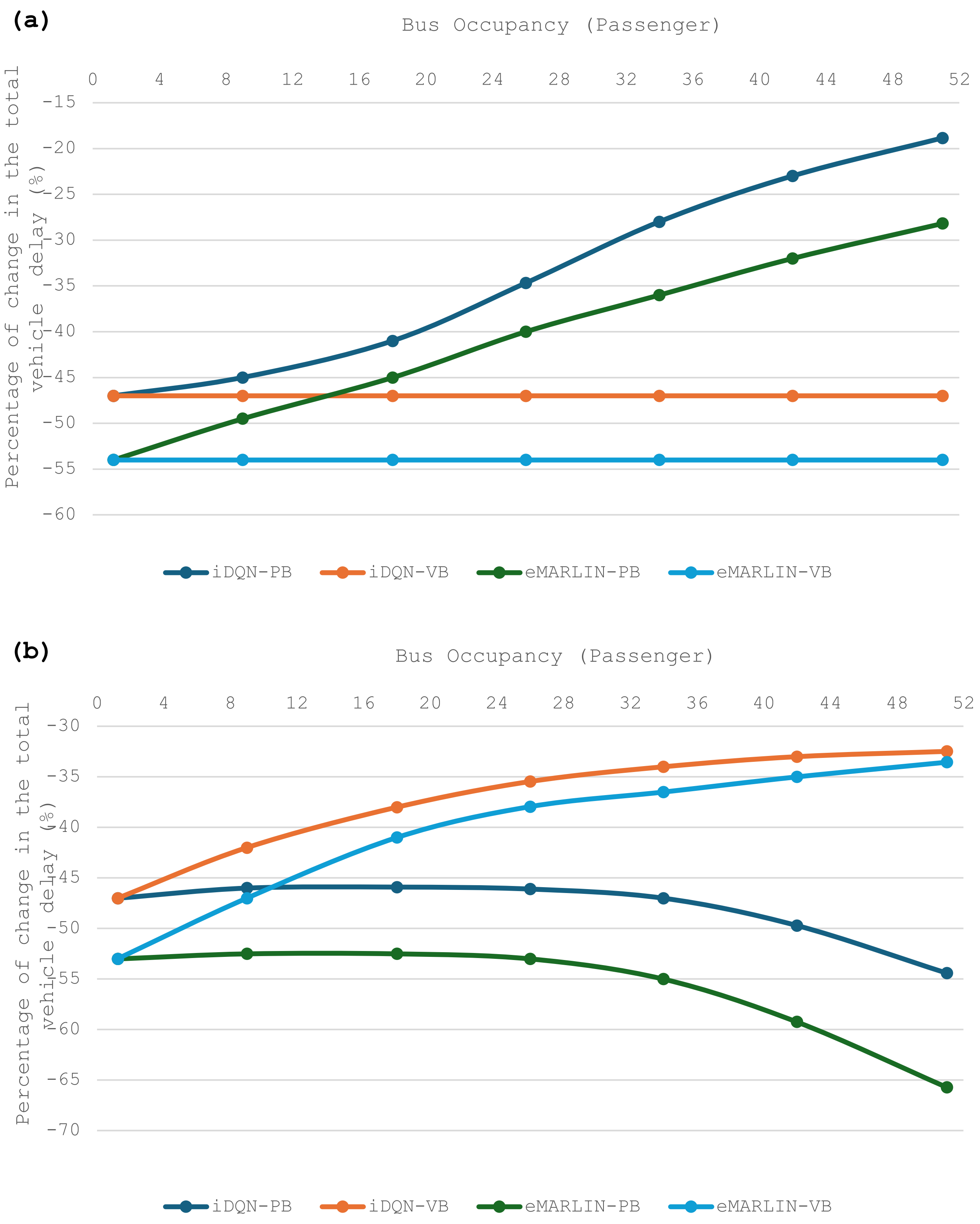


Fig. 9. Percentage of change: (a) in the total vehicle (traffic + transit) and (b) total person (traffic + transit) delays for the different methods tested for the entire corridor at the different bus occupancy levels.

Table 6
Resulting *P*-values of the *t*-test conducted between vehicle and person delays the different algorithms and the pre-timed signals at the different levels of bus occupancy: (a) *P*-values resulting from comparing the person delays and (b)) *P*-values resulting from comparing the vehicle delays.

(a)

Method	Bus Occupancy						
	1.25	9	18	26	34	42	51
eMARLIN-PB	0.000115	0.000103	9.3E−05	8.67E−05	8.32E−05	8.07E−05	7.95E−05
eMARLIN-VB	0.000115	0.000113	0.000112	0.000113	0.000115	0.000122	0.000133
iDQN-PB	0.00013	0.000115	0.0001	9.28E−05	8.93E−05	8.56E−05	8.21E−05
iDQN-VB	0.00013	0.000128	0.000128	0.00013	0.000135	0.000145	0.000161

(b)

Method	Bus Occupancy						
	1.25	9	18	26	34	42	51
eMARLIN-PB	0.000115	0.000115	0.000115	0.000115	0.000115	0.000115	0.000115
eMARLIN-VB	0.000115	0.00011	0.0001	8.48E−05	6.85E−05	5.63E−05	4.61E−05
iDQN-PB	0.000132	0.000132	0.000132	0.000132	0.000132	0.000132	0.000132
iDQN-VB	0.000132	0.000121	0.00011	9.78E−05	8.81E−05	7.83E−05	6.89E−05

about the bus occupancy levels in real-time. In the scenario of the absence of real-time bus occupancy information, the RL agents can rely on historical data to estimate the expected bus occupancy rate based on the time of the day. In addition, the agents can be built with in-built AI-based bus occupancy prediction models that can estimate the bus occupancy rates based on several factors such as the time of the day.

Finally, the *t*-test was used to test the statistical significance of the different methods in comparison to the fixed pre-timed signals. It should be noted that every scenario was tested in 30 simulation replications and the results of these replications were used to conduct the *t*-test and evaluate whether the delays of the developed methods are significantly different from the delays of the pre-timed signals. In addition, a confidence interval of 95 % was used for the *t*-test. The results of the *t*-test are presented in Table 6. The results show that the delays (both vehicle and person delays) of the four different RL methods are significantly different from the delays (both vehicle and person delays) of the pre-timed traffic signals across the different levels of bus occupancy, indicating that the different RL methods produce significantly lower delays when compared to the pre-timed traffic signals.

6. Practical implementation

The integration of the eMARLIN-MM system into adaptive traffic signal control systems presents a significant advancement in urban traffic management. Traditional traffic signal control systems typically rely on pre-set timing plans that are periodically updated based on historical traffic data. These systems, however, lack the flexibility to adapt in real-time to the dynamic nature of transportation networks. The eMARLIN-MM system, on the other hand, continuously learns and adapts to changing traffic and transit conditions by optimizing signal timings through the feedback received from the environment. This adaptability can lead to substantial improvements in person flow efficiency, reduction in congestion, and minimization of travel time and vehicle emissions. Implementing the eMARLIN-MM ATSC system in real-world traffic networks presents significant challenges, particularly due to the difficulty of deploying these systems online without prior training. In the initial training phase, the eMARLIN-MM agents are prone to making errors as they explore the action space, which makes direct application to real traffic networks impractical. Consequently, it is essential to initially train these systems in a simulated microscopic environment (offline training) before deploying them in the real world.

In general, one of the foremost practical implications is the scalability and generalization of RL-based traffic control systems. The effectiveness of these systems in managing traffic flow at the trained intersections has been demonstrated in controlled environments. However, scaling up to more complex intersections and dynamic traffic environments presents substantial challenges. In a real-world scenario, intersections vary widely in terms of geometry, traffic volume, and patterns. An RL system must generalize well across these diverse settings to be effective. For the eMARLIN-MM ATSC system, it scales nearly linearly with the number of intersections in the network because each agent focuses only on its immediate neighborhood. However, to manage large networks (for example with over 100 intersections), special techniques such as the intersection layout parameterization and agent parameter sharing are needed. Intersection layout parameterization standardizes the observation representation for intersections with varying layouts, ensuring that the system can handle different intersection configurations with the same algorithm. Agent parameter sharing allows agents with identical observation representations to share their policy parameters, effectively reducing the computational load and enhancing scalability. By combining these techniques, eMARLIN-MM should be able to create a single learnable agent whose policy can be broadcast to all intersections, thereby simplifying the scaling process and improving efficiency.

The ability to generalize to complex intersection layouts is vital for the practical application of RL-based traffic control systems. The eMARLIN-MM system achieves this through its flexible observation representation, which scales linearly with the number of incoming lanes. This flexibility allows for the customization of eMARLIN-MM agents with minimal effort, accommodating various intersection configurations without significant modifications to the underlying algorithm. This generalizability ensures that eMARLIN-MM can be deployed across diverse urban environments, handling a wide range of intersection complexities effectively. Similarly, adapting to dynamic traffic demands is another critical aspect for the practical implementation of RL-based traffic control systems. Current implementations of eMARLIN-MM rely solely on real-time observations as inputs, which limits the agents' ability to distinguish between different traffic demand patterns. Thus, the use of historical data can help the eMARLIN-MM algorithm to address this limitation. For example, the use of long short-term memory (LSTM) or the Transformer methods in the eMARLIN-MM algorithm can be beneficial to help the agents memorize historical data and handle different demand patterns. Thus, future work on eMARLIN-LSTM-MM and eMARLIN-Transformer-MM can focus on addressing this limitation by incorporating historical-observation augmentation. By including historical data in the agent's input, eMARLIN-LSTM-MM and eMARLIN-Transformer-MM can improve the agents' ability to recognize and adapt to various traffic demand patterns. This enhancement should enable the system to generalize more effectively to dynamic traffic conditions, ensuring more robust and responsive traffic management.

The deployment of RL-based traffic control systems such as eMARLIN necessitates substantial investments in hardware and infrastructure. The system relies on an extensive network of sensors, cameras, and communication devices to gather real-time traffic data and implement control actions. These components must be robust, durable, and capable of withstanding varying environmental conditions. Regular maintenance and calibration are essential to ensure accurate data collection and system performance. Additionally, the integration of eMARLIN-MM with existing traffic management infrastructure requires significant upgrades or replacements of traffic signals, controllers, and communication networks. These infrastructure changes involve considerable financial and logistical efforts, which must be accounted for in the planning and implementation phases. However, with the current infrastructure in the City of Toronto and the City's investment in RL-based ATSC, the implementation of eMARLIN-MM should be feasible. The City of Toronto employs the Advanced Traffic Management System (ATMS), a sophisticated technological platform designed for remote monitoring and control of traffic signals. This computerized system enables transportation administrators to effectively manage the city's transportation network by promptly adjusting signal timings in response to dynamic traffic conditions. ATMS gathers real-time traffic information from a variety of sources including traffic sensors, cameras, and other data inputs. This comprehensive data enables transportation managers to make informed decisions regarding traffic signal timing and other aspects of traffic management. The system currently utilizes the Adaptive Traffic Control Systems to achieve this functionality. ATSC, implemented across Toronto, utilizes a network of sensors, cameras, and data sources to continuously monitor traffic conditions. Its key strength lies in its ability to dynamically adapt to changes in traffic flow. For instance, in response to sudden increases in traffic due to special events or road closures, the system can swiftly adjust signal timings to minimize congestion and reduce delays. Furthermore, by optimizing signal timings to reduce stops and delays, ATSC enhances vehicle travel times, thereby improving the efficiency of person and goods movements. Within Toronto's ATSC framework, two main technologies are employed: Split Cycle Offset Optimization Technique (SCOOT) and Signal Coordination and Timing (SCAT). These technologies contribute to the system's ability to effectively manage traffic flow and optimize transportation efficiency throughout the city (Humber College, 2024). In addition, given the availability of the required infrastructure, the City of Toronto is investing in more advanced RL-based ATSC systems (Habibinia, 2024). Thus, the current infrastructure technology in the City of Toronto can support the implementation of the eMARLIN-MM system.

7. Conclusion

This study focused on developing a multimodal decentralized MARL algorithm, called eMARLIN-MM, to optimize vehicle or person throughput through the intersections. The proposed eMARLIN-MM method consists of two components: the encoder which is responsible for receiving observations and transforming them into latent space and the executor which receives the latent space and serves as the Q-network that is responsible for generating the final decisions for the eMARLIN-MM method. In this method, the agents are communicating by sharing information between direct neighboring intersections. Two different types of eMARLIN-MM were developed: vehicle-based (eMARLIN-MM-VB) which focuses on maximizing the total vehicle delay at the intersection and person-based (eMARLIN-MM-PB) which focuses on maximizing the total person delay at the intersection. In addition, the performance of the developed eMARLIN-MM methods was tested in a simulation model of a subnetwork in the City of Toronto consisting of five intersections categorized into two classes: major-major and major-minor. Furthermore, the performance of the eMARLIN-MM method was compared to the independent iDQN methods, the pre-timed traffic signal controller (that is currently used in the simulated area in absence of adaptive control), and unconditional TSP. Finally, the efficiency of the different methods is tested at different levels of bus occupancy levels. The results of this study can be summarized as follows:

- The proposed eMARLIN-MM method can achieve substantial improvement in the total delay at the intersection. eMARLIN-MM-VB for the case of uniformly distributed bus occupancy, when compared to the existing pre-timed signal plan, can achieve −54 % reduction in the total vehicle delays while eMARLIN-MM-PB can achieve −51 % reduction in the total person delays.
- For the entire corridor, vehicle-based methods achieve the lowest total vehicle delays with −47 and −57 % reductions in the total vehicle delays for iDQN and eMARLIN-MM compared to pre-timed signal plan. In addition, vehicle-based iDQN and eMARLIN-MM methods achieve −35 % and −38 % reduction in total person travel times. On the other hand, person-based methods achieve the lowest total person delays with −46 and −51 % reduction in the total person delays for iDQN and eMARLIN-MM compared to pre-timed signal plan. Moreover, person-based iDQN and eMARLIN-MM methods achieve −35 and −40 % reduction in the total vehicle travel times. Thus, the results show that eMARLIN-MM outperforms the iDQN method achieving lower delays for both the vehicle-based and person-based methods.
- Bus occupancy has no impact over the vehicle-based methods because these methods focus on optimizing the total vehicle delay regardless of the occupancies of the different types of vehicles. On the other hand, bus occupancy is an important factor that

affects the influence of person-based methods that assign weights to the vehicles based on their level of occupancy. In general, the ability of person-based methods to minimize the total person delays at the intersection increases with the increase in the level of bus occupancy –47 % and –54 % for iDQN and eMARLIN-MM at low bus occupancies to –54 % and –66 % for iDQN and eMARLIN-MM at high bus occupancies.

- At low bus occupancy rates (when both buses and passenger vehicles have the same level of occupancy), the results show that both vehicle-based and person-based methods converge and achieve the same results for both the iDQN and eMARLIN-MM methods. This is because the two reward functions become identical.
- Person-based methods adjust control in accordance to bus occupancy and hence provide priority based on bus load. When buses have low occupancy, the controller automatically reduces priority accordingly. Therefore, person-based methods seamlessly tackle the tradeoff between traffic and transit.

Future work can focus on considering other modes of transportation such as pedestrians. Pedestrians are widely impacted by traffic signals, particularly in downtown areas and business districts; however, they are rarely integrated into traffic signal control optimization problems. In addition, future work should consider including other transit performance metrics to the optimization problem such as simultaneously improving transit delays and headway regularity. Moreover, future studies should focus on coordinating ATSC with merging strategies such as the green light optimal speed advisory system (GLOSA) for the general traffic and dynamic bus lanes (DBLs) and driver advisory system (DAS) for public transit.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

CRediT authorship contribution statement

Kareem Othman: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xiaoyu Wang:** Writing – review & editing, Validation, Software, Methodology, Formal analysis. **Amer Shalaby:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Baher Abdulhai:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis.

References

Abdulhai, B., Kattan, L., 2003. Reinforcement learning: introduction to theory and potential for transport applications. Can. J. Civil Eng. 30 (6), 981–991.

Abdulhai, B., Pringle, R., Karakoulas, G.J., 2003. Reinforcement learning for true adaptive traffic signal control. J. Transport. Eng. 129 (3), 278–285.

Aimsun, 2022. Aimsun Next 22 User's Manual. Barcelona, Spain, 2022. https://docs.aimsun.com/next/22.0.1/. Accessed 20 Feb 2023.

Bazzan, A.L., 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. Auton. Agent. Multi. Agent. Syst. 18, 342–375.

Bellman, R., 1956. Dynamic programming and Lagrange multipliers. Proc. Natl. Acad. Sci. 42 (10), 767–769.

Bouktif, S., Cheniki, A., Ouni, A., El-Sayed, H., 2023. Deep reinforcement learning for traffic signal control with consistent state and reward design approach. Knowl. based Syst. 267, 110440.

Chen, B., Cheng, H.H., 2010. A review of the applications of agent technology in traffic and transportation systems. IEEE Trans. Intell. Transport. Syst. 11 (2), 485–497.

Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., Li, Z., 2020. Toward a thousand lights: decentralized deep reinforcement learning for large-scale traffic signal control. In: Proceedings of the AAAI Conference on Artificial Intelligence, 34, pp. 3414–3421.

Chia, I., Wu, X., Dhaliwal, S.S., Thai, J., Jia, X., 2017. Evaluation of actuated, coordinated, and adaptive signal control systems: a case study. J. Transport. Eng., Part A: Syst. 143 (9), 05017007.

Christofa, E., Papamichail, I., Skabardonis, A., 2013. Person-based traffic responsive signal control optimization. IEEE Trans. Intell. Transport. Syst. 14 (3), 1278–1289.

Chu, T., Wang, J., Codecà, L., Li, Z., 2019. Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Trans. Intell. Transport. Syst. 21 (3), 1086–1095.

Consultancy, W.C., 2002. Urban Transport Fact Book. URL http://www.publicpurpose.com/tfb-ix.htm.

Correll, N., Hayes, B., Heckman, C., Roncone, A., 2022. Introduction to Autonomous robots: mechanisms, sensors, actuators, and Algorithms. Mit Press.

Currie, G., Shalaby, A., 2008. Active transit signal priority for streetcars: experience in Melbourne, Australia, and Toronto, Canada. Transp. Res. Rec. 2042 (1), 41–49.

Data Management Group. 2018. Design and Conduct of the Survey. Toronto, 2018. http://dmg.utoronto.ca/pdf/tts/2016/2016TTS_Conduct.pdf.

Diab, E., Kasraian, D., Miller, E.J., Shalaby, A., 2020. The rise and fall of transit ridership across Canada: understanding the determinants. Transp. Policy (Oxf.) 96, 101–112.

Ducrocq, R., Farhi, N., 2023. Deep reinforcement Q-learning for intelligent traffic signal control with partial detection. Int. J. Intell. Transport. Syst. Res. 21 (1), 192–206.

Dujardin, Y., Boillot, F., Vanderpooten, D., Vinant, P., 2011. Multiobjective and multimodal adaptive traffic light control on single junctions. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1361–1368.

El-Tantawy, S., Abdulhai, B., 2010. An agent-based learning towards decentralized and coordinated traffic signal control. In: 13th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 665–670.

El-Tantawy, S., Abdulhai, B., Abdelgawad, H., 2013. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto. IEEE Trans. Intell. Transport. Syst. 14 (3), 1140–1150.

Evans, R., 2006. Transportation engineering. In: Using the Engineering Literature. CRC Press, pp. 536–568.

Gao, J., Shen, Y., Liu, J., Ito, M. and Shiratori, N., 2017. Adaptive traffic signal control: deep reinforcement learning algorithm with experience replay and target network. arXiv preprint arXiv:1705.02755.

Gartner, N.H., 1983. OPAC: a demand-responsive strategy for traffic signal control (No. 906).

Gartner, N.H., Assmann, S.F., Lasaga, F., Hous, D.L., 1990. MULTIBAND–a variable-bandwidth arterial progression scheme. Transp. Res. Rec. (1287).

Genders, W. and Razavi, S., 2016. Using a deep reinforcement learning agent for traffic signal control. arXiv preprint arXiv:1611.01142.

Genders, W., Razavi, S., 2020. Policy analysis of adaptive traffic signal control using reinforcement learning. J. Comput. Civil Eng. 34 (1), 04019046.

Gong, Y., Abdel-Aty, M., Cai, Q., Rahman, M.S., 2019. Decentralized network level adaptive signal control by multi-agent deep reinforcement learning. Transport. Res. Interdiscipl. Perspect. 1, 100020.

Gong, Y., Abdel-Aty, M., Yuan, J., Cai, Q., 2020. Multi-objective reinforcement learning approach for improving safety at intersections with adaptive traffic signal control. Accid. Anal. Prevent. 144, 105655.

Habibinia, M., 2024. Toronto's Traffic Jams Are infamous. Here's the New $20,000-an-intersection Solution the City Hopes Will Unsnarl them. Toronto Star.

Head, K.L., Mirchandani, P.B. and Sheppard, D., 1992. Hierarchical framework for real-time traffic control (No. 1360).

Henry, J.J., Farges, J.L., Tuffal, J., 1984. The PRODYN real time traffic algorithm. In: Control in Transportation Systems. Pergamon, pp. 305–310.

Holm, P., Tomich, D., Sloboden, J., Lowrance, C.F., 2007. Traffic Analysis Toolbox Volume IV: Guidelines for Applying CORSIM Microsimulation Modeling Software (No. FHWA-HOP-07-079). United States. Department of Transportation. Intelligent Transportation Systems Joint Program Office.

Hu, W., 2022. Adaptive Transit Signal Priority Algorithms for Optimizing Bus Reliability and Travel Time using Deep Reinforcement Learning. University of Toronto (Canada).

Hu, W.X., Ishihara, H., Chen, C., Shalaby, A., Abdulhai, B., 2023. Deep reinforcement learning two-way transit signal priority algorithm for optimizing headway adherence and speed. IEEE Trans. Intell. Transport. Syst. 24 (8), 7920–7931.

Humber College, 2024. Smart Solutions for Toronto's Traffic Woes: advancements in ATMS. Available at: https://appliedtechnology.humber.ca/assets/files/civil-engineering-show/2024/SmartSolutionsForTorontoTrafficAdvancementsInATMS.pdf.

Hunt, P.B., Robertson, D.I., Bretherton, R.D. and Winton, R.I., 1981. SCOOT-a traffic responsive method of coordinating signals (No. LR 1014 Monograph).

Kolat, M., Kővári, B., Bécsi, T., Aradi, S., 2023. Multi-agent reinforcement learning for traffic signal control: a cooperative approach. Sustainability 15 (4), 3479.

Levinson, H.S. and Mentor, U.I., 2003. Bus Rapid Transit on City Streets How Does It Work.

Li, D., Wu, J., Xu, M., Wang, Z., Hu, K., 2020. Adaptive traffic signal control model on intersections based on deep reinforcement learning. J. Adv. Transport. 2020, 1–14.

Litman, T., 2020. Evaluating Public Transit Benefits and Costs. Victoria Transport Policy Institute, Victoria, BC, Canada.

Long, M., Wang, R., Chen, J., Chung, E. and Oguchi, T., Cooperative Transit Signal Priority for the Arterial Road with Multi-Agent Reinforcement Learning to Improve Schedule Adherence. Available at SSRN 4472361.

Long, M., Zou, X., Zhou, Y., Chung, E., 2022. Deep reinforcement learning for transit signal priority in a connected environment. Transport. Res. Part C: Emerg. Technol. 142, 103814.

Mauro, V., Di Taranto, C., 1989. UTOPIA. In: Proc. IFAC/IFIP/IFORS Symp. Control, Comput., Commun. Transp., pp. 245–252 1989.

Miletić, M., Ivanjko, E., Gregurić, M., Kušić, K., 2022. A review of reinforcement learning applications in adaptive traffic signal control. IET Intell. Transport Syst. 16 (10), 1269–1285.

Miletić, M., Kušić, K., Gregurić, M., Ivanjko, E., 2020. State complexity reduction in reinforcement learning based adaptive traffic signal control. In: 2020 International Symposium ELMAR. IEEE, pp. 61–66.

Milkovits, M.N., 2008. Modeling the factors affecting bus stop dwell time: use of automatic passenger counting, automatic fare counting, and automatic vehicle location data. Transp. Res. Rec. 2072 (1), 125–130.

Miller, E., Vaughan, J., Yusuf, B. and Higuchi, S., 2018. Surface transit speed update report.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M., 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

O'Toole, R., 2018. Charting public transit's decline. Policy Anal. 853, 1–18.

Orski, K., 2000. Can Alternatives to Driving Reduce Auto Use? Innovat. Br. 11 (1).

Othman, K., 2021. Public acceptance and perception of autonomous vehicles: a comprehensive review. AI Ethics 1 (3), 355–387.

Pol, E., Oliehoek, F.A., 2016. Coordinated deep reinforcement learners for traffic light control. In: Proceedings of Learning, Inference and Control of Multi-agent Systems (at NIPS 2016), 8, pp. 21–38.

Robertson, D.I., 1969. TRANSYT: a traffic network study tool.

Russell, S.J., Norvig, P., 2016. Artificial Intelligence: a Modern Approach. Pearson.

Schultz, G.G., Sheffield, M.H., Bassett, D., Eggett, D.L., 2020. Impacts of Changing the Transit Signal Priority Requesting Threshold on Bus Performance and General Traffic: A Sensitivity Analysis (No. UT-20.06). Dept. of Transportation. Division of Research, Utah.

Shabestary, S.M.A., Abdulhai, B., 2018. Deep learning vs. discrete reinforcement learning for adaptive traffic signal control. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 286–293.

Shabestary, S.M.A., Abdulhai, B., 2022. Adaptive traffic signal control with deep reinforcement learning and high dimensional sensory inputs: case study and comprehensive sensitivity analyses. IEEE Trans. Intell. Transport. Syst. 23 (11), 20021–20035.

Shabestray, S.M.A., Abdulhai, B., 2019. Multimodal intelligent deep (mind) traffic signal controller. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 4532–4539.

Shalaby, A., Farhan, A., 2004. Prediction model of bus arrival and departure times using AVL and APC data. J. Public Transport. 7 (1), 41–61.

Shalaby, A., Hu, W.X., Corby, M., Wong, A., Zhou, D., 2021. Transit signal priority: research and practice review and future needs. Handbook of Public Transport Research. Edward Elgar Publishing, pp. 340–372.

Shalaby, A., Lee, J., Greenough, J., Hung, S., Bowie, M.D., 2006. Development, evaluation, and selection of advanced transit signal priority concept directions. J. Public Transport. 9 (5), 97–120.

Shen, W., Zou, L., Deng, R., Wu, H., Wu, J., 2023. A bus signal priority control method based on deep reinforcement learning. Appl. Sci. 13 (11), 6772.

Sims, A.G., 1979. The Sydney coordinated adaptive traffic system. Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems 1979.

Skabardonis, A., Christofa, E., 2011. Impact of transit signal priority on level of service at signalized intersections. Proc. Soc. Behav. Sci. 16, 612–619.

Smith, H.R., Hemily, B. and Ivanovic, M., 2005. Transit signal priority (TSP): a planning and implementation handbook.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT press.

Sweet, M., Harrison, C., Kanaroglou, P., 2015. Congestion Trends in the City of Toronto (2011-2014). McMaster Institute for Transportation and Logistics, Hamilton, Ontario, Canada.

Szepesvári, C., 2022. Algorithms for Reinforcement Learning. Springer nature.

the American Public Transportation Association, 2018. Public Transportation Fact Book. https://www.apta.com/wp-content/uploads/Resources/resources/statistics/Documents/FactBook/2018-APTA-Fact-Book.pdf.

Trafficware, 2015. Synchro. [Online]. Available: http://synchrogreen.com.

Transportation Association of Canada, 2016. Urban Transportation Indicators: Fifth Survey. TAC, Ottawa, ON 2016.

TTC, 2020. Bus Lane Implementation Plan. TTC Board, Toronto, Canada 2020.

Wang, X., Abdulhai, B. and Sanner, S., 2023a. A critical review of traffic signal control and a novel unified view of reinforcement learning and model predictive control approaches for adaptive traffic signal control. Handbook on Artificial Intelligence and Transport, pp. 482–532.

Wang, X., Taitler, A., Smirnov, I., Sanner, S., Abdulhai, B., 2023b. eMARLIN: distributed coordinated adaptive traffic signal control with topology-embedding propagation. Transp. Res. Rec., 03611981231184250.

Webster, F.V., 1958. Traffic signal settings (No. 39).

Xu, X., Zuo, L., Huang, Z., 2014. Reinforcement learning algorithms with function approximation: recent advances and applications. Inf. Sci. (N.Y.) 261, 1–31.

Yazdani, M., Sarvi, M., Bagloee, S.A., Nassir, N., Price, J., Parineh, H., 2023. Intelligent vehicle pedestrian light (IVPL): a deep reinforcement learning approach for traffic signal control. Transport. Res. part C: Emerg. Technol. 149, 103991.

Yu, J., Laharotte, P.A., Han, Y., Leclercq, L., 2023. Decentralized signal control for multi-modal traffic network: a deep reinforcement learning approach. Transport. Res. Part C: Emerg. Technol. 154, 104281.