The following publication M. Long and E. Chung, "Multi-Agent Reinforcement Learning for Cooperative Transit Signal Priority to Promote Headway Adherence," in IEEE Transactions on Intelligent Transportation Systems, vol. 26, no. 3, pp. 3588-3602, March 2025 is available at https://doi.org/10.1109/TITS.2025.3533603.

Multi-Agent Reinforcement Learning for Cooperative Transit Signal Priority to Promote Headway Adherence

Meng Long, Edward Chung

***Abstract*—Headway regularity is an essential indicator of transit reliability, directly influencing passenger waiting time and transit service quality. In this paper, we employ multi-agent reinforcement learning (MARL) to develop a Cooperative Transit signal priority strategy with Variable phase for Headway adherence (CTVH) under a multi-intersection network. Each signalized intersection is controlled by an RL agent, which determines the next step's signal, adapting to real-time traffic dynamics of transits and non-transits and promoting transit headway adherence. The proposed approach considers four critical aspects, i.e., complicated states with multiple conflicting bus requests, rational actions constrained by domain knowledge, comprehensive rewards balancing buses and cars, and a collaborative training scheme among agents. They are correspondingly addressed by proper state representation with estimated bus headway deviations, irrational actions masking, reward functions formulated by general traffic queue and transit headway deviation, and appropriate MARL approach with synchronous action processing. Our method also takes into account the phase transition loss by setting yellow and all-red time. Simulation results compared with the coordinated fixed-time signal (CFT) and bus holding (BH) strategy verify the merits of the proposed method in terms of improvements in transit headway adherence and influence on general traffic. Based on the results, we further discuss the BH method's limitations due to bus bay length and various holding lines and the CTVH method's benefits in the three-intersection environment and the entire-line network. The proposed method has a promising application in practice to improve transit reliability.**

***Index Terms*—Transit signal priority; traffic signal control; multi-agent reinforcement learning; arterial road; headway adherence.**

I. Introduction

RELIABILITY is a crucial attribute in evaluating the quality of transit service and determining the travel experience of transit users. A basic definition of bus reliability is the stability of the transit performance over time, such as travel time, waiting time, boarding time, seat availability, and so on [1], [2]. Hence, it is directly related to the attractiveness of public transit to current and potential riders [3]. Besides transit users, transit reliability is also significant to transit operators as operational costs are tied to transit service levels.

Headway adherence is one element significantly affecting bus reliability [4]. The variability of traffic states and passenger demand always leads to headway instability [5]. Once the headway between the target bus and the bus in front increases, there will be more people waiting at bus stations, and then the longer boarding time caused by increasing waiting passengers would increase bus dwell time and enlarge the headway more. Then, the headway between the target bus and the bus behind decreases, and the fewer waiting passengers and less boarding time would cause the bus behind to run faster and finally catch the target bus after a period of time. This phenomenon is well-known as bus bunching [6]. In this situation, more transit users experience crowded buses and wait for a long time, deteriorating travel comfort [7].

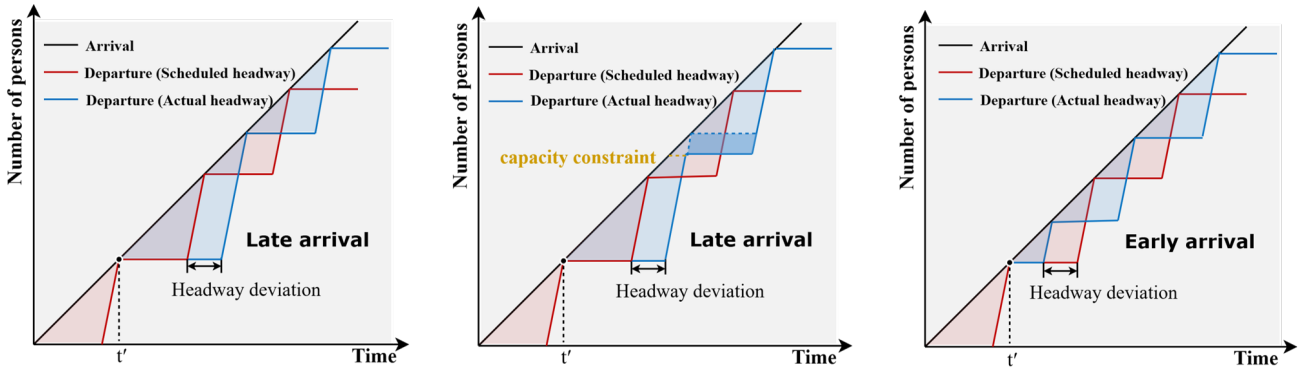
To better understand the effects of headway adherence on the passenger waiting phenomenon, the arrival and departure of passengers at a stop are shown in Fig. 1 under late arrival and early arrival. Scheduled headway is always set evenly, but actual headway can be uneven due to many factors, such as traffic conditions and passenger demand. In Fig. 1a, the black line denotes passenger arrivals, and the red and blue lines represent actual and expected passenger departures, respectively. One bus arrives late after time t', so there are more passengers waiting for this late bus. The blue area is the cumulative waiting time when one bus comes late, while the red area is the cumulative waiting time when buses arrive on schedule. The first blue triangle area after time t' is much bigger than the red, showing a much longer waiting time for passengers. If there is a passenger capacity constraint for buses like in Fig. 1b, the first bus after time t' does not have enough capacity for all waiting passengers, then the rest who fail to board are required to wait for the second bus. Assume that the second bus has enough capacity for all the waiting people when it arrives at the stop. Compared to the blue area of Fig. 1a and 1b, we can find that capacity constraint causes waiting time worse, and the dark blue area of Fig. 1b shows the increased waiting time under capacity constraint compared to no capacity constraint.

In contrast, Fig. 1c shows the case that the first bus after time t' arrives early, but the following buses arrive according to the scheduled headway, so the blue area is small. Fewer people arrive for the early arrival situation, so it is not easy to overpass bus capacity, which is simpler than the late arrival situation. However, as the figure shows, the bus company needs to run

Manuscript received December 12 2023; revised June 24 2024 and November 23 2024; accepted December 29 2024. The work of Meng Long was supported in part by the Foundation Project of Chongqing Normal University under Grant 24XWB040.

Meng Long is with the National Center for Applied Mathematics, Chongqing Normal University, Chongqing 401331, China (e-mail:longmeng@cqnu.edu.cn)

Edward Chung is with the Department of Electrical and Electronic, The Hong Kong Polytechnic University, Hong Kong 999077, China (e-mail: edward.cs.chung@polyu.edu.hk)



(a) When the bus arrives late without capacity constraint.

(b) When the bus arrives late with capacity constraint.

(c) When the bus arrives early.

Fig. 1. The cumulative passenger waiting time under different conditions.

one more bus to carry the same number of passengers, so the early arrival situation will increase operational costs. In addition, if the first bus arrives early in reality, it is more likely that the following bus will come with a large headway (larger than scheduled headway), which will cause a situation like Fig. 1a. Then, the cumulative waiting time would increase a lot.

Therefore, non-adherence to headway has a significant negative influence on transit users and transit operators. To even out headway and also solve bus bunching problems, a variety of strategies have been developed, i.e., real-time information dissemination [8], bus holding strategy [9]–[14], speed control [13], [15]–[17], transit signal priority (TSP) [17]–[20], etc. Real-time information dissemination is a policy approach to changing passenger boarding behavior by providing in-vehicle congestion information, further mitigating bus bunching. This is an indirect control measure and highly relies on passenger assumption compared to other methods. The bus holding strategy is to hold early arrival buses at stations for a while to enlarge headway. Conversely, there also are ways to shorten headway for late buses, like stop-skipping [21] and bus insertion [22]; however, the former would deteriorate the experience of passengers waiting at skipped bus stops, and the latter would increase the costs of bus operators as bus insertion requires many standby buses positioned at suitable places. Speed control uses an adaptive control scheme to accelerate the late buses and decelerate the early buses, but it must always trace and control all buses, which is more complicated than TSP strategies. TSP strategies can modify traffic signals according to bus states, which are usually used to decrease bus delays and improve schedule adherence [23]. In addition, they are also effective for headway adherence, e.g., prioritizing buses arriving late with large headways and not prioritizing buses arriving early with small headways, though research on this is very limited.

This paper aims to propose an adaptive TSP strategy with multi-agent reinforcement learning to promote headway adherence. Its contributions can be concluded in the following three aspects:

- We devised a Cooperative Transit signal priority strategy with Variable phases for Headway adherence (CTVH) algorithm under a multi-intersection network, which includes a complex state representation that accounts for multiple conflicting buses, rational actions constrained by domain knowledge, and comprehensive rewards that balance transit and non-transit objectives. The CTVH algorithm distinguishes buses by distance to stop line and headway deviation and handles multiple bus priority requests by optimizing overall queue length and headway deviation.
- Considering actions of different intersections in the multi-intersection network are asynchronous due to yellow and all-red time, we implement synchronous processing in the CTVH algorithm by treating both yellow and all-red time as actions to enable a centralized training scheme and enhance cooperation among agents. We constrain their durations using the Invalid Action Masking algorithm, and this processing allows second-by-second decisions, facilitating a more responsive TSP strategy and flexible traffic signal control.
- To assess the performance of the CTVH algorithm, we designed a three-intersection arterial simulation experiment with high dynamic demand and complicated bus lines. To further verify the effectiveness of the CTVH algorithm on the improvement of passenger waiting time, we simulate a whole bus line of Xi'an across 22 signalized intersections. The results confirm that CTVH outperforms other baselines and has a high potential for practical applications.

The remainder of this paper is organized as follows. Section II introduces some related works about RL-based signal control and innovative transit control techniques. Section III describes the methodology of this work, including the problem statement and details of the proposed algorithm. Section IV presents the experiment. Its results are shown in Section V. Section VI discusses the limitations of the BH method and the benefits of the CTVH method. We conclude this study and list future works in Section VII.

II. Related Works

A. RL-based signal control

Traffic signal control (TSC) is crucial for optimizing vehicle flow through intersections, aiming to minimize congestion and travel times. Traditional methods like fixed-time control, actuated control, and adaptive control have been widely used. Fixed-time control relies on pre-set signal timings based on historical data, actuated control adjusts to real-time traffic using sensors, and adaptive systems like SCOOT and SCATS dynamically update timings based on live traffic data [24]. However, these methods often rely on predefined rules and thresholds, which may not be able to adapt to the dynamic and complex nature of traffic conditions. To address this limitation, researchers have turned to Reinforcement Learning (RL). The RL agents can learn to adjust the traffic signal timings and phases based on the current traffic conditions, with the goal of optimizing a specific objective, such as minimizing average vehicle delay or maximizing the throughput of the intersection [25].

Several RL algorithms have been explored for TSC, including value-based, policy-based, and actor-critic RL methods [26]. Early value-based RL approaches, like Q-learning [27], [28], showed promise in discrete actions but faced scalability issues. Its deep variant, Deep Q-Networks (DQN) [29], [30], combines with neural networks and has been extensively used to handle large state spaces and learn effective policies from raw traffic data. Double Dueling Deep Q-Network (D3QN) [31] enhances the traditional DQN by combining Double Q-learning to mitigate overestimation bias and the dueling architecture to separately estimate state value and advantage, leading to more reliable and efficient learning. Various policy-based RL algorithms have also been deployed in TSC, such as Policy-Gradient [32], and Proximal Policy Optimization (PPO) [33], [34]. Some other popular actor-critic methods have also been explored in TSC by researchers, like Actor-Critc [35], advantage actor-critic (A2C) [36], and Deep Deterministic Policy Gradient (DDPG) [24], which excel in continuous action space. Based on comparison results of the previous study [37], D3QN outperforms other RL methods in problems with limited and discrete actions in terms of less computation and efficient convergence.

MARL is another advancement, where multiple agents control different intersections and learn to cooperate, and it has shown promise in managing large-scale urban networks. Techniques such as Independent Q-Learning with information sharing [38], multi-agent A2C [39], multi-agent DDPG [24], and Value Decomposition MARL [40], [41] enable decentralized decision-making while ensuring different levels of cooperation among traffic signals according to their learning schemes. The centralized training scheme offers advantages in the high cooperation among agents, the collective good performance, and the stable learning process over decentralized training [42]. In addition, these methods have demonstrated significant improvements in minimizing traffic delays and enhancing overall traffic efficiency. However, few of these approaches have incorporated a transit priority strategy within environments featuring complex transit routes. The fixed 5-second step length commonly employed in most studies [24], [36], [43] is inadequate for responding promptly to real-time transit priority requests. Additionally, few studies consider signal constraints, including minimum and maximum green time, yellow time, and all-red time before phase transitions, which are essential for ensuring the safety and rationality of RL actions. In this work, we propose a cooperative TSP strategy within a centralized training decentralized execution framework, considering multiple conflicting bus routes and basic signal constraints, aimed at improving headway adherence and alleviating bus bunching issues. As the asynchronous decision-making among intersections due to constraints of yellow and all-red time would pose challenges to obtaining all agents' observations, actions, and rewards at each time step for centralized training, we conduct synchronous processing. In this process, green, yellow, and all-red times are treated as actions, with their sequences and durations constrained by an Invalid Action Masking algorithm [44]. Furthermore, the proposed method incorporates a simple yet comprehensive state representation designed to alleviate high computational demands and enhance scalability in larger network environments.

B. Innovative transit control

Many researchers have explored innovative control methods to improve the efficiency and reliability of public transit systems. Those methods can have two common types, including the spatial and temporal approaches. Spatial control is optimizing the allocation of road space to improve transit movement, such as dedicated transit lanes [45], [46], queue jump lanes [47], and curb management [48]. To improve the effectiveness of transit lanes, Zhao and Zhou [49] proposed a dynamic exclusive transit lane strategy. It set a median opening upstream of the intersection and used a pre-signal to allow left-turn buses to enter the opposing transit lane intermittently. Then, exclusive transit lanes at the exit can be dynamically used for the left turn and the opposite through buses during the various periods of a signal cycle. With the advent of Connected Vehicle (CV) technology, Xie et al. [50] developed a Cooperative Dynamic Bus Lane System. In this system, both CVs and buses can use the bus lane, but CVs ahead of buses on the bus lane would implement cooperative lateral control with CVs on the adjacent lane to clear the bus lane and then benefit buses.

In addition to spatial control, various temporal approaches have also been widely used. They refer to methods that control buses from a temporal aspect, such as bus holding [9], [51] and TSP [18], [52]. Daganzo [9] proposed a dynamic model to determine bus holding times at control points, targeting consistent headways. However, static control has an inherent technical limitation, as the optimal headway is neither fixed nor known prior. Wang and Sun [10] proposed a dynamic bus-holding strategy based on the RL framework. This method models each bus as an agent that interacts with its leader and follower, and determines the holding time when a bus arrives at a bus stop. Later, they further introduced distributional MARL and designed a meta-learning-based framework for learning

effectiveness and robustness in holding control [51]. While bus holding can only delay buses, TSP can modify signal timings to affect buses passing intersections quickly or slowly. TSP strategies can be classified into three types: passive [53], active [54], and adaptive [55] strategy. Adaptive TSP methods have the capability to respond dynamically to real-time traffic conditions and are commonly formulated using optimization models [4], [55], [56].

A limited number of studies have explored the application of RL for adaptive TSP [20], [28], [30], [52]. Recently, Hu et al. [20] developed an adaptive TSP for an isolated intersection by Double Deep Q-Networks (DDQN) to improve headway adherence, but they only considered a two-way transit line and extended or reduced the green time by a multiple value of five. To consider the cooperation among intersections in arterials, Yu et al. [57] introduced a cooperativeness state into state representation, including the set of last actions taken by neighbor agents, and proposed a decentralized TSP with the DQN algorithm. However, this decentralized learning approach achieves limited cooperation compared to centralized learning, and their designed reward function overlooks the precise value of headway deviation and treats all buses requesting priority equally. Also, they only considered the two-phase traffic signal scenarios and each decision step lasts for 5s. Thus, these works are limited in their ability to handle complex traffic environments, including multiple-intersection networks, common four-phase traffic signal schemes, and multiple conflicting priority requests, and the lengthy decision-making steps do not facilitate immediate responses to TSP requests.

Therefore, to address the identified research gaps, this study develops a centralized-learning-decentralized-execution MARL framework for cooperative TSP strategy in multi-intersection scenarios. The framework enhances headway adherence through second-by-second decision-making, accommodating realistic environments with four-phase traffic signal schemes and complex bus routes. This approach aims to facilitate more effective, responsive, and reliable public transportation systems.

III. Methodology

A. Problem statement

This work considers a multi-intersection environment with *N* signalized intersections and complex bus routes from different directions. Each intersection is equipped with a multi-phase signal controller with three signals per phase, containing green, yellow, and red. This work aims to develop a TSP strategy to modify traffic signals in real-time to promote headway adherence based on the observed traffic states. The modification of the traffic signal is regarded as the action, and the improvement of headway adherence is the reward. Therefore, this problem can be modeled as a Markov Decision Process (MDP) with the objective of decreasing headway deviation and be addressed by MARL.

Headway deviation is the deviation of the current headway from the scheduled headway, which is an essential indicator shaping reward to reflect the headway adherence and is required to evaluate actions at each decision step. It is easy to calculate buses' headway deviation when they arrive at bus stops, as the arrival time of all buses at stops are collected, and the headway is the difference between the arrival time of the current bus and the last bus at this stop. However, it is rather complicated to obtain the headway deviation in real-time, especially when buses are running between stops.

Here are two headway deviations: forward headway deviation FHD and backward headway deviation BHD. Fig. 2 shows the headway and schedule delay diagram. The green vehicle is the current bus, and the bus in front (or behind) is called the forward (or backward) bus here. Forward (or backward) headway is the headway between the current bus and the corresponding forward (or backward) bus. Forward (or backward) headway deviation is the deviation of forward (or backward) headway from scheduled headway. To ensure the general applicability of the method, the studied road network does not necessarily start from the bus terminal. Therefore, each bus is assigned a random initial schedule deviation ISD upon entering the network to account for the traffic randomness encountered before reaching the network. ISD_k denotes the initial schedule deviation of bus k; FH_t^k and BH_t^k are the forward and backward headway of bus k at time t; SH denotes the scheduled headway; FHD_t^k and BHD_t^k are the forward and backward headway deviation of bus k at time t. For each bus line, the real-time headway deviations are computed as follows.

When the first bus of each line enters the simulated network at time $t_0 = ISD_1$, to make it follow the schedule, we regard the forward headway deviation $FHD_{t_0}^1$ as equal to the initial schedule deviation ISD_1. Though the backward bus 2 hasn't emitted in the simulated area, it would enter at time $t_1 = SH + ISD_2$. Therefore, the $BHD_{t_0}^1$ can be calculated by the difference of $(t_1 - t_0)$ and SH. Specifically, $FHD_{t_0}^1$ and $BHD_{t_0}^1$ can be given by

$$FHD_{t_0}^1 = ISD_1 \tag{1}$$

$$BHD_{t_0}^1 = ISD_2 - ISD_1 \tag{2}$$

When the following bus k enters the network at time $t_{k-1} = (k-1) * SH + ISD_k$, its FHD equals to the forward bus $(k-1)$'s BHD. Though the backward bus $k+1$ is still not emitted, it would enter the network at time $t_k = k * SH + ISD_{k+1}$. Thus, $BHD_{t_{k-1}}^k = (t_k - t_{k-1}) - SH$. That is

$$FHD_{t_{k-1}}^k = BHD_{t_{k-1}}^{k-1} \tag{3}$$

$$BHD_{t_{k-1}}^k = ISD_{k+1} - ISD_k \tag{4}$$

When bus k runs in the network: In Fig. 2, we describe the distance of buses by time as all buses of one line set the same commercial speed. Commercial speed is the average journey speed, including delays at bus stops and traffic lights [58]. Distance can be approximated by multiplying the time by the scheduled speed when buses run between two bus stops. From Fig. 2, we can derive:

$$FH_t^k = FH_{t-\Delta t}^k + \delta_c - \delta_f \tag{5}$$

where $\delta_c = \Delta\ t - \frac{\Delta D_c}{V}$, $\delta_f = \Delta\ t - \frac{\Delta D_f}{V}$; δ_c and δ_f are the delay of the current and forward buses in Δt; ΔD_c and ΔD_f

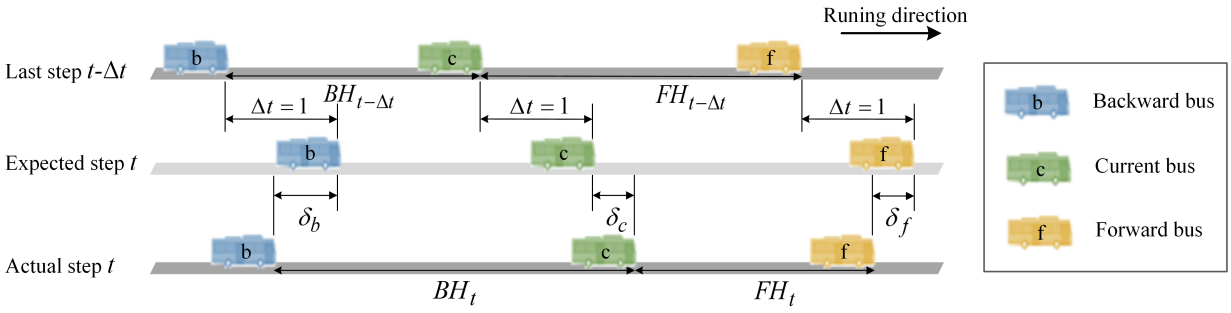


Fig. 2. The headway and delay diagram.

are the traveled distance of the current and forward buses in Δt; V is the pre-set commercial speed of buses across various traffic conditions, 20 km/h. The green bus of Fig. 2 actually runs ahead of the expected location after Δt, so δ_c is negative and has a positive sign in the above formula. According to the definition, we know

$$FHD_t^k = FH_t^k - SH \tag{6}$$

Based on Eq. 6, Eq. 5 can be derived to

$$FHD_t^k = FHD_{t-\Delta t}^k + \delta_c - \delta_f \tag{7}$$

Similarly, the calculation of backward schedule deviation can be derived as follows:

$$BHD_t^k = BH_t^k - SH \tag{8}$$

$$BH_t^k = BH_{t-\Delta t}^k + \delta_b - \delta_c \tag{9}$$

$$BHD_t^k = BHD_{t-\Delta t}^k + \delta_b - \delta_c \tag{10}$$

where $\delta_b = \Delta t - \frac{\Delta D_b}{V}$; δ_b is the delay of the backward buses in Δt; ΔD_b is the traveled distance of the backward buses in Δt. In addition, the relationship between forward and backward schedule deviation can be described as

$$FHD_t^k = BHD_t^{k-1} \tag{11}$$

Hence, we can calculate real-time BHD by Eq. 10 and use Eq. 11 to obtain FHD when buses run in the network. Then, each component of the MDP process is able to be obtained.

B. Components of MARL

RL is a subfield of machine learning and an effective tool for solving problems formulated as the MDP. Given the process is in one state s_t at step t, the agent interacting with the environment chooses an action a_t based on a policy π, and the process would move to another state s_{t+1} (also labeled as s') at the next step $t+1$, finally obtaining a reward r_t correspondingly. State, action, reward, and next state form one transition experience $< s_t,\ a_t,\ r_t,\ s_{t+1} >$. After learning from numerous experiences, the RL agent strives to find the optimal policy π^* which can maximize the expected return (also known as the discounted cumulative reward and Q-value). MARL is an extension of the single-agent RL for more complex environments. Considering the complexity of the TSP strategy in a multi-intersection environment, the MARL approach is utilized. The following describes the key components of this MARL framework, including agent, state, action, and reward.

1) Agent definition: The signal controller of each intersection is a learning-based agent. Given the observed state s_t^i of the environment at time step t ($t \in T$), the agent of intersection i ($i \in N$) takes an action $a_t^i \in \Lambda_t^i$ (Λ_t^i is the valid action space) following a policy π and then obtains a reward r_t^i when the environment transitions to the next state ${s'}_t^i$ (i.e., s_{t+1}^i). The objective of agents is to find an optimal policy π^* which maximizes the expected return.

2) State representation: For intersection i, the state represents four-part information at each time step t, i.e., $s_t^i = (S_{A_t}{}^i,\ L_{Q_t}^i,\ P_{C_t}{}^i,\ S_{Bus_t}{}^i)$.

Average speed $\mathbf{S_{A_t}^i}$

$$S_{A_t}{}^i = \left\{ S_{Alm_t}{}^i \right\},\ l \in \Psi, m \in M_l \tag{12}$$

where $S_{Alm_t}{}^i$ represents the average speed of movement m at the approach link l for intersection i at time t; Ψ is the approach set, including southbound (SB), northbound (NB), eastbound (EB), and westbound (WB) approaches; M_l is the movement set of approach link l, containing through and left-turn movements for right-hand traffic, in which right-turns are permitted to pass on red.

Queue length $\mathbf{L_{Q_t}^i}$

$$L_{Q_t}^i = \left\{ L_{Qlm_t}{}^i \right\},\ l \in \Psi, m \in M_l \tag{13}$$

where $L_{Qlm_t}{}^i$ represents the queue length of movement m at the approach link l for intersection i at time t. If a traffic movement occupies more than one lane, e.g., two lanes for through traffic, then we take the maximum queue length of those lanes as the queue length of that movement.

Current phase $\mathbf{P_{C_t}^i}$

It is a one-hot encoding vector with several values, each representing one traffic light indication: green, yellow, and all-red time of all phases. The vector consists of 0 in all cells except for a single 1 in a cell used uniquely to represent the current signal situation. For example, if the current signal shows the green time of the second phase in a common four-phase signal scheme, $P_{C_t}^i = (0,0,0,1,0,0,0,0,0,0,0,0)^T$.

Bus state $\mathbf{S_{Bus_t}^i}$

The representation of bus information will determine the effectiveness of RL agents serving bus priority requests on headway adherence, especially in complex situations with

multiple conflicting priority requests. In this work, we also collect the distance to stop line D, schedule delay SD, and passenger occupancy O of buses approaching the intersection. Bus schedule delay is the delay of a bus compared to its schedule, which would be negative if the bus is earlier than its schedule. Passenger occupancy is the number of bus passengers available by the onboard Automatic Passenger Counters. Moreover, forward headway deviation FHD, backward headway deviation BHD, and whether at bus stops AtS are also included.

For the metric AtS, many sensors are installed at bus stops that can detect when a bus has arrived and also the GPS can retrieve the real-time location of buses to verify whether buses are at bus stops. $AtS = 1$ if the bus is at stops; otherwise, $AtS = 0$. Finally, the bus state is denoted by six-tuple information of buses arriving at intersections:

$$S_{Bus} = \{D_{pb}, SD_{pb}, O_{pb}, FHD_{pb}, BHD_{pb}, AtS_{pb}\}, \quad p \in P, b \in \Gamma \tag{14}$$

where D_{pb}, SD_{pb}, O_{pb}, FHD_{pb}, and BHD_{pb} represent the distance to the stop line, schedule delay, passenger occupancy, forward headway deviation, and backward headway deviation of bus b, which belongs to the movements of phase p. AtS_{pb} denotes whether bus b controlled by phase p is at bus stops. P is the phase set, such as $P = \{0, 1, 2, 3\}$ in a four-phase signal scheme. It should be noted that the bus state only considers buses that have already arrived at intersections. We define the arrival of a bus as 1) its distance to the stop line is less than 40 m, or 2) it queues at the intersection. Γ is the bus set, and here we define $|\Gamma| = 3$. Thus, if there are more than three arrived buses detected at an intersection and belong to movements of phase p, we only put the information of the three most urgent buses in S_{Bus} for phase p. As buses with larger headway deviation and closer to the stop line would be more urgent to take into account, the urgency is measured by $\frac{|FHD_{pb}|+|BHD_{pb}|}{D_{pb}+\delta}$ where $\delta = 1$ to avoid invalid calculations. By this formula, the higher headway deviation and closer to the stop line would cause a larger urgency value. Therefore, we can simultaneously consider up to 12 bus requests from different directions at each intersection. Finally, S_{Bus} is a 12x6 matrix.

3) Action definition: Consider a series of traffic phases allowing phase skipping. The valid action space Λ_t^i is a set of valid signal phases for intersection i at each step t. The validity is determined by signal constraints (namely, the minimum and maximum green time) and skipping rules [37] using Invalid Action Masking [44]. Specifically, the phase-skipping rules are (1) When the queue length of one phase flow exceeds the predefined maximum threshold, this phase cannot be skipped; (2) Each phase cannot be skipped twice in a row if vehicles are detected in its corresponding approach lane. Actions that don't satisfy signal constraints and skipping rules are regarded as invalid actions considering traffic safety, equality, and efficiency. And those invalid actions would add a large negative value on their original action-value Q, forming masked action-value Q^{masked}. This penalty for masking must be smaller than the minimum action-value, ensuring that agents would not choose them as they will select the action maximizing Q^{masked}. Therefore, the action a_t^i of agent i is to select one phase for the next step $t+1$ from Λ_t^i. In this case, the phase duration and sequence are totally determined by agents.

The signal constraints would cause asynchronous decision-making time for different agents as agents are restricted to experiencing yellow time and all-red time when the last action is to change to another phase. Such asynchrony would cause difficulty in obtaining all agents' observations, actions, and rewards at each time step for centralized learning. To address the asynchrony of agents, we conduct a simple synchronous processing, which also treats the yellow time and all-red time as an action and constrains their sequences and durations by Invalid Action Masking so that each agent can select one action at every step. Then the universal action space is a set of the green, yellow, and all-red of all considered phases. The valid action space is a subset of them. In this study, we set 3s yellow time and 1s all-red time before transitioning to another phase to enhance traffic safety at intersections and also consider the phase transition loss.

4) Reward shaping: This work seeks the optimal policy that can 1) prioritize transits to promote bus headway adherence while bringing less detrimental effects on non-transit vehicles, and 2) improve the general traffic efficiency when there are few buses. Given the need for cooperation among the agents and their collective impact on the network performance, all agents share a common joint reward r_t, which serves as the individual reward r_t^i for each agent. Hence, the reward r_t^i is defined as the weighted sum of queue density and headway gains, given by

$$r_t = w_1 \frac{\sum_i \sum_l \sum_\varphi L_{Q_{l\varphi t}}^i}{\sum_i \sum_l \sum_\varphi Det_{l\varphi}^i} + w_2 \frac{\sum_k \phi_k \left(FHD_t^k - BHD_t^k\right)}{\sum_k |\phi_k|} \tag{15}$$

where $L_{Q_{l\varphi t}}^i$ and $Det_{l\varphi}^i$ represent the queue length at time t and detector length on the lane φ of the approach link l at intersection i, respectively; ϕ_k is a parameter given by $\phi_k = \begin{cases} 1 & \text{action=the green phase for bus } k \\ -1 & \text{others} \end{cases}$ and bus k belongs to buses in S_{Bus} of all intersections; w_1 and w_2 are a negative weight for queues of general traffic and a positive weight for headway gains of buses, respectively.

Here we take both forward and backward directions into account for headway gains. Note that, according to Eq. 6 and 8, the positive value of FHD and BHD denote a larger headway than the scheduled one. It is positive to select the green phase for bus k when bus k has a larger forward headway than scheduled headway as prioritizing bus k benefits on shortening the distance with the forward bus. In contrast, it is harmful to select the green phase for bus k when bus k has a larger backward headway than scheduled headway as prioritizing bus k enlarges the distance with backward bus and deteriorates headway adherence.

C. CTVH algorithm

This work utilizes a MARL framework to propose the *Cooperative TSP strategy with Variable phase for Headway*

adherence (CTVH). It is based on two RL approaches, D3QN [59] and QMIX [60].

D3QN, combining Double Q-learning and the dueling architecture, can mitigate overestimation bias and obtain more accurate and stable action-value. This leads to more reliable and efficient learning, which is crucial for our complex TSP decision-making problem, where overestimation could result in suboptimal signal decisions. Additionally, previous studies have shown that D3QN outperforms other RL methods in TSC, such as policy gradient and actor-critic, in problems with limited and discrete actions [37]. These alternative approaches suffer from lower sample utilization and the potential for erroneous parameter updates, especially during the early stages of training.

In detail, to reduce overestimation bias, Double Q-Learning separates action selection from action evaluation by using a target network. Then the action is chosen using the current Q-network $\arg\max_a Q(s', a; \theta)$, but the value of that action is evaluated using the target Q-network $Q'(s', a; \theta^-)$ The target value is computed as:

$$y_i^{\text{Double-Q}} = r + \gamma Q'\big(s', \arg\max_a Q(s', a; \theta); \theta^-\big) \tag{16}$$

where $y_i^{\text{Double-Q}}$ is the target value for the i-th update in Double Q-Learning; r is the reward received after taking action a in state s; γ is the discount factor that determines the importance of future rewards $(0 \leq \gamma \leq 1)$; Q' is the Q-value estimated by the target network; $\arg\max_a$ selects the action that maximizes the Q-value according to the current Q-network; and θ and θ^- represent the parameters of the online and target Q-networks, respectively.

To improve learning efficiency, the dueling architecture splits the Q-function into two streams: the value function, which measures the value of being in state s, and the advantage function, which measures the relative importance of action a in state s. The Q-value is computed as:

$$Q(s, a; \theta) = V(s; \theta) + \left(\dot{A}(s, a; \theta) - \frac{1}{|\mathcal{A}|} \sum_{a'} \dot{A}(s, a'; \theta) \right) \tag{17}$$

where $V(s; \theta, \beta)$ is the state-value function representing the overall value of state s; $\dot{A}(s, a; \theta)$ is the advantage function measuring the relative importance of action a in state s; $|\mathcal{A}|$ is the total number of actions.

To further enhance the cooperation, scalability, and efficiency of our multi-agent system, we employ the QMIX architecture. QMIX aggregates local decisions into a global policy, ensuring that local TSP actions contribute positively to overall network performance. It has a mixing network that decomposes the joint action-value function into individual agent action-value functions, ensuring consistent and stable cooperation among agents. This aligns with our goal of achieving headway adherence across the entire network, and each intersection agent, after centralized training with a mixing network, can choose independent actions while still yielding cooperative performance.

QMIX combines the individual Q-values of agents into a global Q-value using a monotonic mixing function to ensure consistency between local and global optimization. The global Q-value is represented as:

$$Q_{\text{tot}}(S, A) = f_{\text{mix}}(Q_1, Q_2, \ldots, Q_n; \Phi) \tag{18}$$

where Q_{tot} is the global Q-value for the joint state S, and joint actions A; f_{mix} is the mixing function, implemented as a neural network; $Q_1, Q_2, \ldots, Q_n$ are the individual Q-values for agents $1, 2, \ldots, n$; and Φ represents the parameters of the mixing network.

The monotonicity constraint in QMIX is given by:

$$\frac{\partial Q_{\text{tot}}}{\partial Q_i} \geq 0 \tag{19}$$

where Q_i is the local Q-value for agent i; and $\frac{\partial Q_{\text{tot}}}{\partial Q_i}$ is the partial derivative of the global Q-value with respect to the local Q-value of agent i, ensuring monotonicity. This guarantees that improving an agent's local Q-value will not decrease the global Q-value. By maintaining such a direct and consistent relationship between the agents' individual learning processes and the global system performance, QMIX is able to scale to more agents and more complex environments without sacrificing performance.

Therefore, the proposed CTVH leverages the strengths of D3QN and QMIX to implement TSP, improving headway adherence under a multi-intersection environment. The CTVH framework, depicted in Fig. 3, utilizes Deep Neural Networks (DNN) to approximate the Q-value for each agent, denoted as $Q_t^i(s_t^i, a_t^i)$, and a Mixing Network to derive the global Q-value, $Q_{tot}(S_t, A_t)$. Here, $S_t = \left\{s_t^i, i \in N\right\}$ represents the set of states, and $A_t = \left\{a_t^i, i \in N\right\}$represents the set of actions for all agents.

Specifically, as Fig. 4 shows, given that the state encompasses various information such as speed, queue length, current phase, and bus information, a DNN with convolutional layers (Conv) and fully connected layers (FC) is employed to first extract state features and then approximate the action-value for each possible action in a given state. Considering that different intersections may have varying impacts on overall system performance, e.g., improvement at a key intersection may benefit the entire network more, the Mixing Network is leveraged for value decomposition. This network takes S_t and each agent's Q_t^i as inputs to produce the total Q-value Q_{tot}. Within the Mixing Network, various components include absolute value functions (ABS), layer weights (w1 and w2), and nonlinear activation functions such as Exponential Linear Units (ELU) and Rectified Linear Units (ReLU). These elements facilitate neural network learning and capture highly complex relationships within the data [60]. Thus, the Mixing Network can learn the nonlinear relationships between individual agent contributions and overall system performance. The computed Q_{tot} and targeted global value y_{tot} will be utilized to calculate the loss by Eq. 20 and train the entire network.

$$\mathcal{L}(\Theta) = \mathbb{E}\left[\left(Q_{\text{tot}}(S, A; \Theta) - y_{\text{tot}}(S, A)\right)^2\right] \tag{20}$$

In addition, a synchronous action processing is conducted to enable global Q-value estimation at each step while adhering to constraints such as minimum and maximum green time,

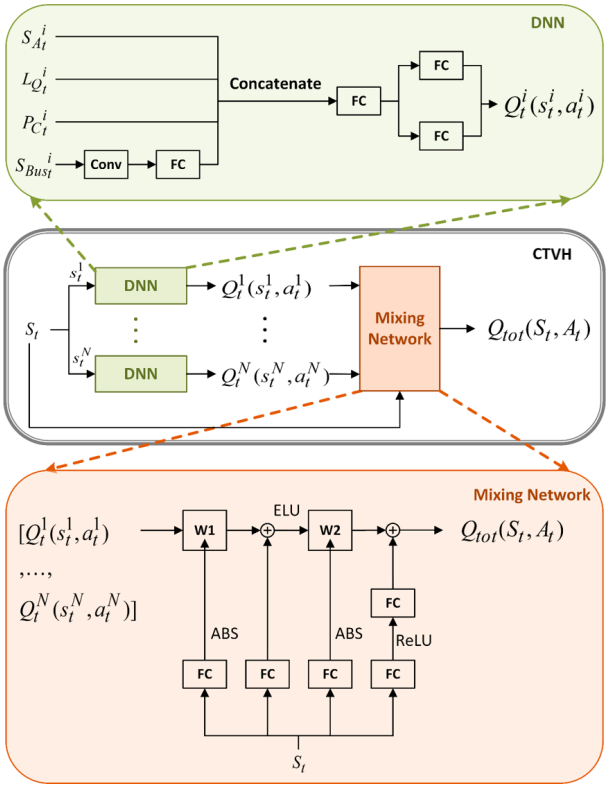


Fig. 3. The CTVH framework.

fixed yellow and all-red time, i.e., we consider yellow and all-red phases as separate actions rather than bundling them with the next phase's green time. Consequently, the minimum and maximum time constraints for the yellow (all-red) phase are all set as 3s (1s). This allows yellow and all-red time to be treated as discrete actions similar to green time. Using an invalid action masking approach ensures that these signal constraints are met and also enables agents to compute $Q_t^i(s_t^i, a_t^i)$ and subsequently $Q_{tot}(S_t, A_t)$ for network training.

D. Training procedure

The training process of CTVH involves the following 7 main steps, as shown in Table I: (1) Initialize primary Q_{tot} and target Q'_{tot}networks, including the DNN of all agents and the mixing network. (2) In each episode, each agent perceives the current state s_t^i of the environment through observations and selects an action randomly with a probability of ϵ or according to current policy with a probability of 1-ϵ. (3) The environment executes each agent's action, modifies the signal, and moves to the next state. Each agent can receive a new observation s_{t+1}^i and compute an immediate reward r_t using queue length and headway gains. (4) Store such transition experience $< S_t,\ A_t,\ r_t,\ S_{t+1} >$ with calculated importance weight for priority experience replay into the memory buffer. (5) Sample a minibatch of experiences from the buffer by priority experience replay for training. (6) Agents replay these experiences and pass the individual agent state through their DNNs to obtain the individual Q-values. Combining the individual Q-values using a mixing network to estimate the joint action-value. Obtain predicated joint action-value $Q_{tot}(S_t, A_t)$ and target joint action-value y_{tot} (7) Update the primary network by minimizing the mean-squared error between the target and the predicted joint action-value over the minibatch. Also, periodically update the target network with the weights of the primary network.

TABLE I
THE TRAINING PROCEDURE OF THE CTVH ALGORITHM.

Input: Parameters of the CTVH algorithm
Output: Trained parameters of the CTSPV algorithm
Initialize primary and target network parameters Θ and Θ^-
for *episode* $= 1$ **to** the number of episodes **do**
  for $t = 1$ **to** the simulation time of one episode **do**
    // Execution and experience storing
    Perceive state s_t^i and select action
    $$a_t^i = \begin{cases} \arg\max_a Q^{\text{masked}}(s_t^i, a_t^i; \theta), & \text{if rand} > \epsilon, \\ \text{random } a \in \Lambda_t^i, & \text{otherwise.} \end{cases}$$
    for all agents $i \in \{1, \ldots, N\}$
    Execute actions $A_t = \{a_t^i\}$ and observe new state $S_{t+1} = \{s_{t+1}^i\}$ and reward r_t
    Store $(S_t,\ A_t,\ r_t,\ S_{t+1})$ into the memory buffer
    // Training
    Sample a minibatch from memory buffer
    Calculate local Q-values by DNN and global Q-value by mixing network. Obtain Q_{tot} and $y_{\text{tot}} = r_t + \gamma * Q'_{tot}\big(S_{t+1}, \arg\max_{A'}\ Q_{tot}(S_{t+1}, A'; \Theta); \Theta^-\big)$.
    Update primary network by calculated loss (Eq. 20)
    Update target network every N' replace iterations
  end for
end for

TABLE II
THE PARAMETERS SETTINGS.

Parameters	Value
Discounting factor	0.99
Learning rate	0.0001
ϵ of Epsilon-greedy strategy	Exponentially decade from 0.6 to 0.1 with a rate 5e-5
Target networks replace iterations	2000
Size of the replay buffer	20000
Size of minibatch	64
Number of episodes	160
w_1 and w_2 for the reward	-1 and 1/5120
Penalty for masking	-500

For algorithm parameters, most of them are first determined by default values from the open-source package "OpenAI Spinning Up" [61] and PyMARL [62] (e.g., discount factor, learning rate, and batch size), and the early relevant work [37] (e.g., epsilon for greedy strategy, and size of the replay buffer). Starting with those default parameters, we then fine-tuned them via empirical trial and error and also considered the affordability of computational resources (Our computer is equipped with a CPU "Intel Core i9-12900K", and a GPU "GeForce RIX 3070"). By observing cumulative rewards and Q-values over episodes during training, we determine parameters as listed in Table II. In addition, weight settings are calculated by balancing queue density and bi-headway deviations to put equal emphasis on them, which are further discussed in Section V.C. Penalty for masking are set as -500 which must be smaller than the minimum value of the Q-value. During training, the proposed CTVH learns directly from interactions with the environment and captures the nonlinear

relationships between traffic signal settings and traffic flow (including general traffic and bus information). After numerous repeated interactions, RL agents strive to find the optimal policy that can reduce queue length and headway deviation.

IV. Experiment

A. Simulated environment

To assess the performance of the proposed CTVH strategy, we simulated a three-intersection arterial by SUMO as the environment to interact with RL agents, as shown in Fig. 4. Each intersection has four legs, of which EB and WB approaches on the major roads have three lanes, and SB and NB approaches on minor roads have only two lanes. The spacing between two adjacent intersections is 450m. There are bus stops along the intersection, indicated by gray rectangles. Next to each bus stop are labeled the routes that will stop at that location.

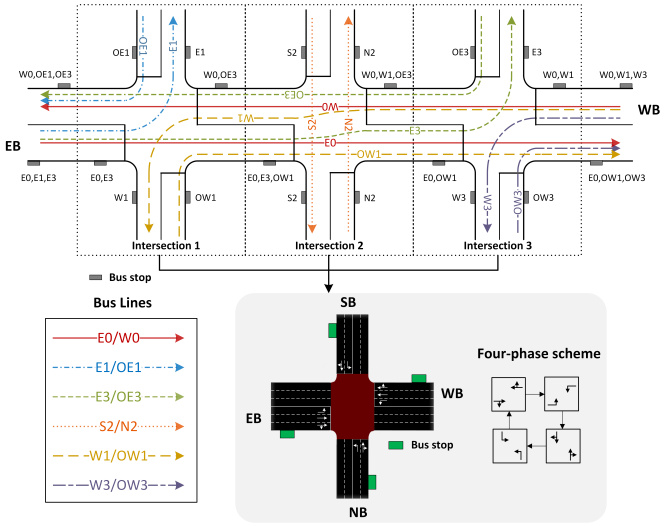


Fig. 4. The simulated environment.

1) Traffic demands: The simulated network has eight inflows and eight outflows, so 56 OD pairs are set. We consider 5-period demands, of which the flow ratios are around 0.35, 0.5, 0.55, 0.45, and 0.3, respectively. The total simulation time is 95 mins, in which the warm-up is 5 mins, and the time of S1∼S5 are 15, 15, 30, 15, and 15 mins, respectively.

The environment has six two-way bus lines, as shown in Fig. 4. To determine bus schedules for better studying headway adherence, related literature is summarized in Table III. Referring to those studies, we set a 5-min headway for P2∼P4 with higher demand and a 10-min headway for S1 and S5 with lower demand.

2) Traffic signal phase: The traffic signal of each intersection adopts a standard four-phase scheme as shown in Fig. 4, namely, major through phase P_0, major left-turn phase P_1, minor through phase P_2, and minor left-turn phase P_3. All right turns are allowed to pass on red. We emphasize there is a 1s all-red time after 3s yellow and before the green time of other phases.

3) Bus settings: All buses are given one random initial schedule deviation, following a normal distribution N(0,180^2), so that the inserted time of each bus is determined by the sum of scheduled headway and that initial schedule deviation. The capacity of each bus is 70 passengers, and the boarding time per passenger is 2 s. At each stop, the passenger arrival rate of each line is 0.0083 person/s for S1 and S5 and 0.0167 person/s for other periods.

TABLE III
Summarization of scheduled headway of related literature.

Paper	Scheduled headway
Van Oort et al. (2010) [11]	5 mins
Anderson and Daganzo (2020) [18]	10 mins
Long et al. (2020) [19]	1.5∼3 mins
Khan et al. (2023) [6]	2∼20 mins
Wang and Sun (2020) [10]	8 mins
Zhou et al. (2022) [8]	5 mins

B. Compared methods

We compared the proposed CTVH approach with the following baselines.

- **Coordinated Fixed-Time signal (CFT):** We obtain the optimal fixed signal timing considering arterial coordination by SIDRA INTERSECTION, which is an advanced micro-analytical traffic evaluation software.
- **Cooperative Traffic Signal Control (CTSC):**This is a cooperative traffic signal control by MARL framework with a variable phase scheme to decrease queue lengths at intersections and improve traffic efficiency. We use a similar framework to the CTVH algorithm, but the reward function only has the first part of Eq. 15 without considering transit vehicles' reliability.
- **CTVS:** This is a cooperative TSP strategy by MARL framework for arterial roads to improve schedule adherence. We use a similar framework to the CTVH algorithm, but the second part of the reward function (Eq. 15) is formulated by bus schedule delay, instead of headway deviation.
- **Bus Holding strategy (BH):** We refer to the work of [9] and use this strategy to hold buses with small headway at bus stops. Traffic signals at all intersections are controlled by CFT.

C. Evaluation metrics

We evaluate the MARL-based TSP strategies regarding learning efficiency and traffic performance.

Learning efficiency can be clearly obtained from the learning curve. Two indicators are computed to plot the learning curve: scores and average Q-value. Scores are the cumulative rewards in one episode. Average Q-value is another more stable metric, which provides an estimate of how much discounted reward the agent can obtain by following its policy from any given state. We randomly provide 64 fixed states from the simulated environment and obtain the Q-value of the best action selected for each state under the current policy. The average Q-value is an average of Q-values computed over those 64 states. As agents' policies are updated with learning,

changes in scores and average Q-value illustrate algorithm learning performance.

Traffic performance metrics contain:

- **Average Person Delay:** We evaluate the Average Person Delay for general traffic, Cars, and Buses, denoted as APD, APDC, and APDB as follows:

$$APDB = \sum_t \sum_k (d_{kt} * O_{kt}) / \sum_t \sum_k O_{kt} \quad (21)$$

$$APDC = \sum_t \sum_c (d_{ct} * NP_{ct}) / \sum_t \sum_c NP_{ct} \quad (22)$$

$$APD = \frac{\sum_t (\sum_k d_{kt} * O_{kt} + \sum_c d_{ct} * NP_{ct})}{\sum_t (\sum_k O_{kt} + \sum_c NP_{ct})} \quad (23)$$

where d_{kt} and d_{ct} are the normal delay of bus k and car c at time t, given by $d_{kt} = TT_{kt} - \frac{TD_{kt}}{DV_{Bus}}$ and $d_{ct} = TT_{ct} - \frac{TD_{ct}}{DV_{Car}}$, DV_{Bus} and DV_{Car} is the desired velocity of buses and cars, set to be 16.67 m/s in this study; O_{kt} is the occupancy of bus k at time t; NP_c is the number of passengers in the car c; b and c is the bus and car index, respectively.

- **Queue:**

$$Queue = \sum_t \sum_l \sum_\varphi L_{Ql\varphi t} / (T * \sum_l |\Phi_l|) \quad (24)$$

where $L_{Ql\varphi t}$ is the queue length on the lane φ $(\varphi \in \Phi_l)$ of the approach link l $(l \in \Psi)$ at time t $(t \in [0,\ T])$. $|\Phi_l|$ is the size of the approach link l, i.e., the number of lanes of the approach link l.

- **Average Headway Deviation (AHD):** It is the average headway deviation of each bus, computed by

$$AHD = \sum_t \sum_k \left| FHD_t^k \right| / \sum_t |K_t|,\ k \in K_t \quad (25)$$

where K_t is the set of buses running in the network at time t and $|K_t|$ represents the total number of buses.

- **Average Bi-Headway Difference (ABHD):** This term shows how even the forward and backward headway is. It is defined as the average difference between forward and backward headway deviations:

$$ABHD = \sum_t \sum_k \left| FHD_t^k - BHD_t^k \right| / \sum_t |K_t|,\ k \in K_t \quad (26)$$

- **Average Person Waiting Time (APWT):** This belongs to person-based metrics and reflects the passenger waiting time at bus stops, which is calculated by:

$$APWT = \sum_t N_{wt} / \sum_t (N_{a_t} - N_f) \quad (27)$$

where N_{wt} denotes the number of passengers waiting for buses at time t, and its sum in time T equals the waiting time of all passengers; N_{at} denotes the number of passengers arriving at time t, and its sum in time T equals the total number of passengers arrived in the network; N_f denotes the number of passengers finishing their routes in simulation warm-up time, and this term should be neglected in a real traffic environment.

- **Average Person Running Delay on Bus (APRDB):** This is also a person-based metric to illustrate the running performance of buses, which can be computed by

$$APRDB = \sum_t \sum_k (d_{kt} * O_{kt}) / \sum_t (N_{at} - N_{wT}) \quad (28)$$

where N_{wT} denotes the number of passengers waiting for buses at the last time step T. The numerator is the total running delay of all passengers on buses, and the denominator is the total number of passengers arriving in time T.

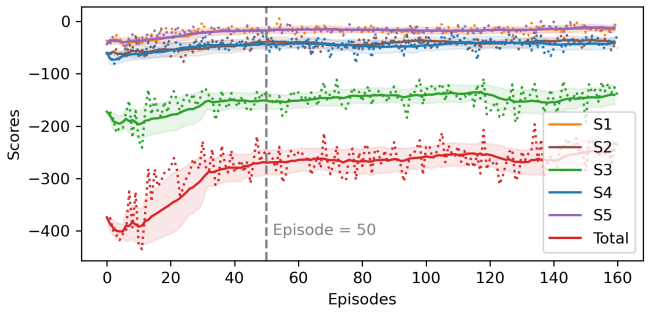
- **Total Person Delay for Bus (TPDB):** The total person delay consists of waiting delays at bus stops and running delays on buses. Hence, we can calculate it by summing APWT and APRDB together.

V. Results

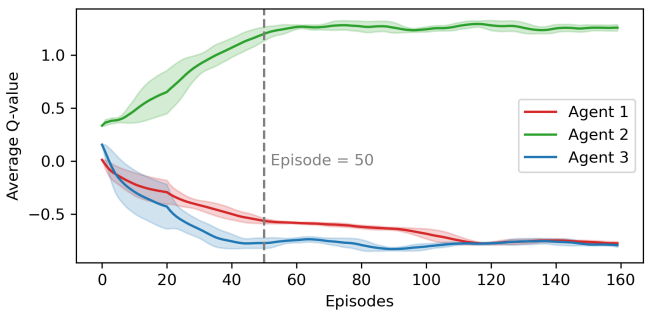
This section assesses the proposed CTVH approach and compares it with other baselines. We trained the proposed algorithm with different random seeds of traffic simulation at each episode, facilitating testing varied traffic situations, and reported the evaluation metrics of each episode. Training results show the learning efficiency and network performance of the proposed method.

A. Algorithm learning

As agents' policies are updated with learning, changes in scores and average Q-value illustrate algorithm learning performance. The results are shown in Fig. 5.



(a) Scores. (S1∼S5 refers to the five periods of traffic demands, and the Total is the sum of all five periods).



(b) Average Q-value.

Fig. 5. The learning curve of the CTVH strategy.

In Fig. 5a, dotted lines are actual scores with episodes, and solid lines are moving average scores per 20 episodes, making them more stable than dotted ones. Areas in the figure indicate the scores' standard deviation. The scores of the peak demand (S3) are lower than off-peaks (S1,2,4 and 5), and score improvements of peak demand are much larger than off-peaks with the training process. After 50 episodes, total scores become much more stable, and their standard deviation remains below 30, which illustrates the algorithm's convergence. In addition, Fig. 5b shows varieties of the average Q-value with the simulation time for different agents. Because the average Q-value is obtained by conditioning on 64 fixed states, its curve does not oscillate as much as the scores. Hence, curves of average Q-value are more intuitive to find that each agent's Q-values reach stability with under standard deviations of 0.1 after 50 episodes. These stable fluctuations in both scores and Q-values demonstrate the convergence of the proposed CTVH algorithm after 50 episodes. The stable learning curves of all three agents also demonstrate less mutual interference among all agents; otherwise, the curves would show constant heavy fluctuations or slow down due to disruptions from other agents rather than converging such smoothly. It should be noted that the Q-value decreases of agents 1 and 3 do not mean learning worse; conversely, it reflects agents are learning well therefore getting closer to the accurate Q-value. Specifically, for the given 64 states, the average Q-values of agents 1 and 3 are negative, so as they learn, their approximation becomes more and more accurate, and their learning curves show a decreasing trend. Also, the specific Q-values of different agents among different agents do not reflect their relative superiority or inferiority as for the three intersections, we do not use the same 64 states; instead, we use each intersection's individual state at the same 64 time points.

B. Performance comparison

The learning curve shows that the proposed algorithm has converged after 50 episodes. Hence, we reported average performance metrics of the last 60 episodes for performance evaluations. Fig. 6 summarizes the performance improvement rates of various approaches compared with the CFT method. BH reduces AHD, ABHD, and APWT by 18.4%, 28.0%, and 16.5%, respectively, which are the highest rates among all methods, but it also seriously increases APD, Queue, and APRDB by 10.7%, 9.4%, and 49.8%, respectively. BH holds buses with small headway to improve headway adherence and reduce passenger waiting time at bus stops while holding behaviors delay buses and passengers in buses heavily. Overall, the delay of transit users, namely TPDB, increases by 16.7%. Therefore, the implementation of BH improves transit headway adherence with severe adverse impacts on general traffic and running delays of buses. In contrast, CTVH significantly decreases AHD and ABHD by 6.1% and 5.3%, without any negative influence on general traffic. For example, it even reduces APD, Queue, APRDB, and TPDB by 3.2%, 15.3%, 6.3%, and 4.0%, respectively. The CTVS approach decreases Queue, APRDB, and TPDB by 27.7%, 6.7%, and 3.6%, respectively but with little reduction in headway deviation. The CTSC shows the best performance on queue length and delay reduction, decreasing APD and Queue by 5.6% and 29.2%, which is 2.4% and 13.9% more than CTVH. However, the CTSC increases AHD and ABHD by 0.12% and 0.02%.

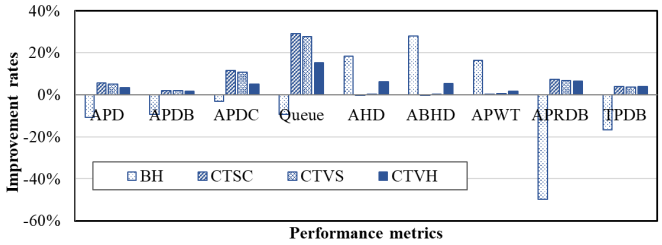


Fig. 6. The performance improvement rates for different methods compared to CFT. (APD: average person delay; APDB and APDC: average person delay of buses and cars, respectively; AHD: average headway deviation; ABHD: average bi-headway difference; APWT: average person waiting time; APRDB: average person running delay on bus; TPDB: total person delay for bus.)

In summary, BH minimizes headway deviation, but delays and queues increase dramatically; CTSC and CTVS minimizes queues and delays significantly, but have no improvement on headway adherence; however, CTVH significantly improves headway adherence while ensuring nonnegative impacts on queue and delays.

C. Weight setting

To facilitate practical application, we need to find proper weights to formulate the reward function. In Eq. 15, we can set $w_1 = -1$ and test various values of w_2. Fig. 7 shows the performance comparison when w_2 varies from 1/1200 to 1/100. As the formulation, the higher w_2 is, the more emphasis is on headway gains compared to queue length. Hence, results of larger w_2 show a smaller decrease in delay, and queue, but greater improvements in headway deviation and passenger waiting time. As the changes of APRDB are always larger than APWT with the changes of w_2, the TPDB improvement would change correspondingly with APRDB.

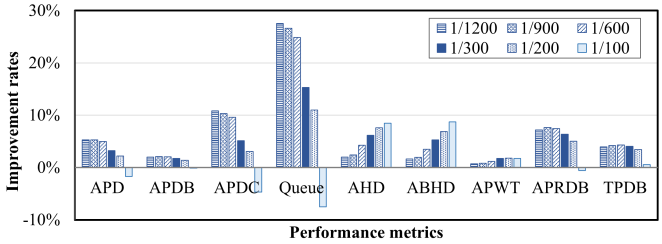


Fig. 7. Performance comparison of different weight settings. (APD: average person delay; APDB and APDC: average person delay of buses and cars, respectively; AHD: average headway deviation; ABHD: average bi-headway difference; APWT: average person waiting time; APRDB: average person running delay on bus; TPDB: total person delay for bus.)

This work aims to minimize headway deviation without negative impacts on general traffic, including delays and queues. Therefore, we need to find a balance between those two parts. We obtain the CFT approach's ABHD = 316.3 s. For the reward function, the first part is the queue density which would fall into [0,1], and the second part is formulated by

average bi-headway gains. Therefore, when balancing these two parts and putting equal emphasis on them, we can set $w_1 = -1$, $w_2 = \frac{1}{316.3} \approx \frac{1}{300}$. As Fig. 7 shows, in this case, CTVH can reduce AHD, ABHD, and APWT by 6.1%, 5.3%, and 1.7%, while APD, Queue, APRDB, and TPDB also decrease by 3.2%, 15.3%, 6.3%, and 4.0%.

In practice, after collecting bi-headway difference, balancing weights can be determined to perform well in headway adherence without harm to general traffic.

D. Ablation study

The proposed CTVH method is based on D3QN and QMIX. The ablation study involves systematically removing or altering the key component of QMIX, the mixing network, to evaluate its contribution. Hence, this comparative analysis includes three methods: Independent D3QN, D3QN+VDN (linear summation), and D3QN+QMIX (mixing network). For Independent D3QN, each agent executes and trains independently without any cooperation. The linear summation method means the joint action-value is the sum of the independent action-values of all agents, commonly known as VDN. After training, results reveal that VDN and QMIX converge around 50 episodes, while Independent D3QN converges after 60 episodes. We also recorded performance metrics of the last 60 episodes to calculate the average improvement rates compared to the Coordinated Fixed-Time signal (CFT).

Fig. 8 shows the results of the ablation study on headway adherence and passenger waiting time. From this figure, it is evident that D3QN with cooperation reduces headway deviation and passenger waiting time more significantly than the Independent D3QN method. Cooperative traffic signal timing between intersections better controls stops and delays experienced by buses, significantly affecting their headway. Furthermore, in a MARL environment, agents interact with each other, influencing each other's behavior and learning. As agents adapt their policies, the environment becomes nonstationary from any individual agent's perspective, making it challenging to attribute rewards or performance to their actions. This dynamic complicates convergence as agents must adapt to evolving strategies of others. Comparing QMIX with VDN, the QMIX-based method achieves higher improvements in headway deviation and passenger waiting time, underscoring the significance of the mixing network in value decomposition. Effective value decomposition helps agents learn their exact contributions to overall performance, enabling optimal action selection and improving headway adherence across the network.

This ablation analysis highlights the impact of cooperation and the mixing network on the performance of the proposed method. Furthermore, the superior performance of the proposed CTVH algorithm compared to the other two methods demonstrates effective cooperation and less mutual interference.

VI. Discussions

This section focuses on the limitations of BH and the benefits of CTVH to answer the following two questions: one is why does BH perform so poorly on delay in this study, and the other one is why does CTVH improve a little on APWT?

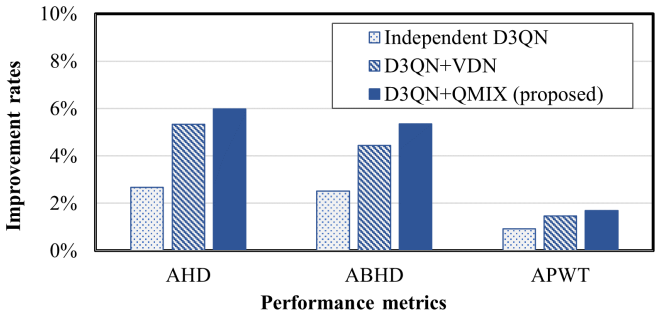


Fig. 8. Results of ablation study. (AHD: average headway deviation; ABHD: average bi-headway difference; APWT: average person waiting time.)

A. Limitations of the BH method

Bus holding strategy has been widely implemented to improve transit reliability; however, in this experiment, simulation results show that the BH method improves headway adherence and passenger waiting time significantly but imposes greater costs on bus delay, car delay, and queue. According to the principle of bus holding strategy, we can infer that the bus delay and person delay on buses would increase a little after holding buses at a stop, but the simulation results are much worse than we expected. By visualizing simulations, we found a severe problem, i.e., bottleneck slowdown and even some queue after bus stops due to the bus holding strategy, as shown in Fig. 9.

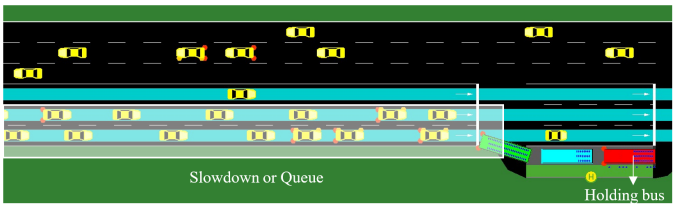


Fig. 9. Queuing after bus stops due to bus holding strategy.

This problem happens when the length of the bus stop bay is limited, and one ahead holding bus occupies this bay area; other following buses scheduled to use the same stop would queue and even block car movements. The queue of following buses and cars results in increased delays for both bus and car travelers and decreased bus service reliability. Moreover, after that holding bus finished its holding time, the following buses of other lines may also need to hold at this stop, so that the queue is not easy to clear due to the holding strategy of all bus lines. In reality, we usually focus on the key bus lines or routes with high passenger demand or serious irregular headways to consider implementing bus holding. Then, those essential buses can benefit a lot from the holding strategy with fewer adverse effects on other vehicles. Overall, there are two reasons for this queueing problem: the limited space of bus stop bays and the large number of bus lines being held.

Consequently, the above two reasons for the BH strategy's poor performance are its limitations in real traffic; namely, the BH strategy can only be applied to networks with long bus stop bays and for a limited number of bus lines.

B. Benefits of the CTVH method

Simulation results do not show a significant improvement of the CTVH approach on APWT compared to the CFT approach, only 1.7%, even though it greatly decreases headway deviation by 6.1%, which seems inconsistent with the advantages of headway adherence on reducing passenger waiting time. To answer this question, we plot three bus trajectories of one line to analyze possible reasons.

As Fig. 10 shows, the X-axis denotes the time, and the Y-axis represents the space. One bus line has many bus stops with an intersection between two stops. The studied network with three intersections only contains around four stops from i to $i+3$ for most lines like Line E0/W0, and other stops after $i+3$ are downstream of the studied network. For simplification, we do not plot the queue time of buses before stop lines at signalized intersections and just change the slopes of bus trajectories between two stops. For example, if a bus queues a long time before the stop line to wait for the green time, the slop of corresponding trajectories between two stops, including this intersection, becomes much smaller; otherwise, the slop becomes steeper when a bus is given a signal priority and pass this intersection quickly. There are three bus trajectories: red dotted lines are scheduled bus trajectories, black solid lines are bus trajectories controlled by CFT without TSP, and green dash lines are bus trajectories with CTVH strategy. We assume that the first bus travels on schedule, and the initial schedule deviation set for the second (third) bus is positive (negative), so the second (third) bus enters the studied network with a large (small) headway.

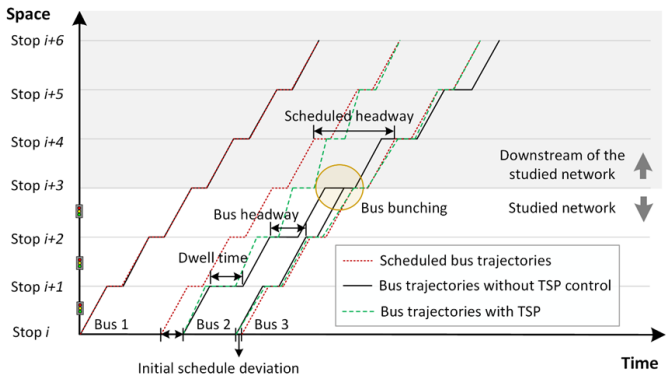


Fig. 10. Bus trajectories of one line.

For the second bus without TSP control, the large headway will encounter more passengers and lengthen its dwell time. Then, the headway gradually increases, and the situation becomes increasingly worse. However, the CTVH strategy can give signal priority to this bus so that it travels faster at an intersection and arrives at the next bus stop earlier than without TSP control. A smaller headway deviation than without control would shorten the dwell time and further reduce headway. After TSP of several intersections, headway will gradually decrease and eventually reach the scheduled one, thus can bringing the bus trajectory back to the scheduled one.

In contrast, for the third bus without TSP control, the small headway at the beginning will make the headway shorter and shorter. This third bus even catches the second bus at stop $i+3$, and those two buses bunch together. However, the CTVH can delay this bus a bit at intersections by giving the signal phase to other movements. After a few intersections, such headway deviation can also be corrected.

The headway irregularities and bus bunching problems can lead to a large passenger waiting time, but these conditions are gradually exacerbated downstream (the gray area of Fig. 10) with time. Because our studied network contains only one short arterial road with three intersections (the white area of Fig. 10), the negative impacts of headway irregularities do not affect passenger waiting time too much, and the effect of early and late arriving buses on passenger waiting time is somewhat neutralized, so the improvements of CTVH strategy would not be so significant. The improvement of APWT by the CTVH approach would be larger if the studied environment covered the whole bus line because it can significantly improve the headway deviation, as the results in the studied three-intersection network show.

To verify the effectiveness of the proposed CTVH algorithm on a complete bus route, we evaluated it in another scenario modeled after Bus Line 2 in Xi'an, China. This route includes 34 bus stops and traverses 22 signalized intersections, as shown in Fig. 11. For each one-hour simulation episode, the environment randomly generates a traffic volume ranging from 6,000 to 15,000 veh/h. Buses on Line 2 depart from both terminals every 5 minutes. The passenger arrival rate at each stop is 0.05 persons per second under medium traffic conditions, and it varies with car flow, calculated by multiplying a demand factor (actual flow/medium flow). Each passenger is assumed to get off after 5 stops. We compared the headway deviation and passenger waiting time results of the CTVH algorithm with the Coordinated Fixed-Time (CFT) signal control. The CFT timings were determined using Webster's Formula and implemented with the tools tlsCycleAdaptation.py and tlsCoordinator.py in SUMO.

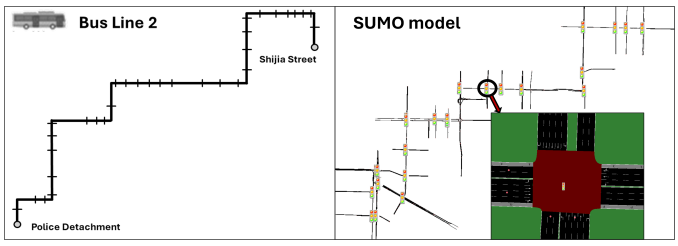


Fig. 11. The entire-line environment.

As traffic flow varies with episodes, a boxplot is used to show the minimum, the maximum, the median, and the first and third quartiles of the improvement rates across 60 episodes, as shown in Fig. 12. The average value is represented by a small cross, and the average values of several indicators are connected by line segments for better observation. From

this figure, the CTVH method can also have significant improvements on both AHD and ABHD, up to 19.4% and 22.9% in mean, respectively. Moreover, it also shows a 6.3% reduction in APWT on average, more significant than 1.7% in the three-intersection scenario. Meanwhile, the proposed method reduces APD and Queue by 4.4% and 7.4%, respectively. Hence, the CTVH can improve headway adherence and reduce passenger waiting time effectively without negative effects on general traffic.

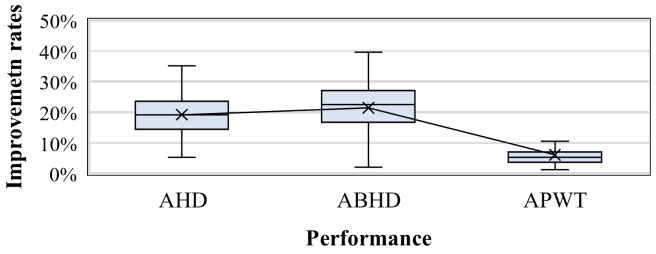


Fig. 12. The performance improvement rates compared to CFT in Line 2 scenario. (AHD: average headway deviation; ABHD: average bi-headway difference; APWT: average person waiting time.)

Additionally, the TSP strategy enables handling early and late buses, while the BH strategy can only hold early buses. This is another benefit of the CTVH method. Compared to the two limitations of the BH strategy, CTVH also has no limits on bus stop bays and has the ability to serve various bus lines.

VII. Conclusion

This paper proposes a CTVH algorithm by MARL for the arterial road to improve transit headway adherence. It determines traffic signals of the next step based on real-time conditions of transits and non-transits, considering multiple conflicting bus requests, rational action constraints, a balance between buses and cars, and collaboration among intersections. Simulation results show that the proposed method outperforms fixed-time signal and bus holding strategy when phase transition losses are considered due to greater improvements in transits and less harm to general traffic. The reward function's weight determines the focus of the proposed algorithm: the higher the weight of headway gains is, the more emphasis is on the transit headway, and the more significant improvements are achieved in headway adherence but with more negative effects on queue and delays. When the weight of the headway part is set to be the quotient of average queue density and average headway deviation, the CTVH algorithm decreases headway deviation significantly (i.e., by 6.1%) without negative effects on general traffic (i.e., APD decreases by 3.2% and Queue decreases by 15.3%) comparing to CFT method. Then we discussed the harm of the BH method due to limited bus bay length and many holding lines. Holding buses would block following buses that are expected to use the same stop and even cause queues of cars behind. In contrast, the benefits of CTVH were also discussed. Although the APWT at the three-intersection scenario decreased slightly by 1.7%, the improvement is more significant at the entire-line scenario by 6.3%. Due to these existing and potential advantages, the proposed method has a promising application in practice.

Some works can be further explored in the future. The performance of the TSP strategy can be improved by cooperating with other control measures, like BH or transit speed advisory. However, those combinations increase the complexity of research problems, as the mutual effects of bus holding time, real-time transit speed, and traffic signals must be considered. Whether using traditional model-based methods or RL algorithms, we need to figure out an appropriate problem formulation. In this case, an entire bus line should be selected as the case study for a thorough performance evaluation.

References

[1] M. A. Turnquist, "Strategies for improving reliability of bus transit service," *Transportation Research Record*, no. 818, 1981.

[2] L. Sun, A. Tirachini, K. W. Axhausen, A. Erath, and D.-H. Lee, "Models of bus boarding and alighting dynamics," *Transportation Research Part A: Policy and Practice*, vol. 69, pp. 447–460, 2014.

[3] X. Chen, L. Yu, Y. Zhang, and J. Guo, "Analyzing urban bus service reliability at the stop, route, and network levels," *Transportation Research Part A: Policy and Practice*, vol. 43, no. 8, pp. 722–734, 2009.

[4] X. Zeng, Y. Zhang, J. Jiao, and K. Yin, "Route-based transit signal priority using connected vehicle technology to promote bus schedule adherence," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1174–1184, 2021.

[5] Y. Zhang, R. Su, Y. Zhang, and B. Wang, "Dynamic multi-bus dispatching strategy with boarding and holding control for passenger delay alleviation and schedule reliability: A combined dispatching-operation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 846–12 860, 2022.

[6] Z. S. Khan, W. He, and M. Menéndez, "Application of modular vehicle technology to mitigate bus bunching," *Transportation Research Part C: Emerging Technologies*, vol. 146, 2023.

[7] J. Soza-Parra, S. Raveau, J. C. Muñoz, and O. Cats, "The underlying effect of public transport reliability on users' satisfaction," *Transportation Research Part A: Policy and Practice*, vol. 126, pp. 83–93, 2019.

[8] C. Zhou, Q. Tian, and D. Z. W. Wang, "A novel control strategy in mitigating bus bunching: Utilizing real-time information," *Transport Policy*, vol. 123, pp. 1–13, 2022.

[9] C. F. Daganzo, "A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons," *Transportation Research Part B: Methodological*, vol. 43, no. 10, pp. 913–921, 2009.

[10] J. Wang and L. Sun, "Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework," *Transportation Research Part C: Emerging Technologies*, vol. 116, 2020.

[11] N. Van Oort, N. H. M. Wilson, and R. Van Nes, "Reliability improvement in short headway transit services," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2143, no. 1, pp. 67–76, 2010.

[12] L. O. Seman, L. A. Koehler, E. Camponogara, L. Zimmermann, and W. Kraus, "Headway control in bus transit corridors served by multiple lines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4680–4692, 2020.

[13] J. Argote-Cabanero, C. F. Daganzo, and J. W. Lynn, "Dynamic control of complex transit systems," *Transportation Research Part B: Methodological*, vol. 81, pp. 146–160, 2015.

[14] J. Rodriguez, H. N. Koutsopoulos, S. Wang, and J. Zhao, "Cooperative bus holding and stop-skipping: A deep reinforcement learning framework," *Transportation Research Part C: Emerging Technologies*, vol. 155, 2023.

[15] C. F. Daganzo and J. Pilachowski, "Reducing bunching with bus-to-bus cooperation," *Transportation Research Part B: Methodological*, vol. 45, no. 1, pp. 267–277, 2011.

[16] K. Ampountolas and M. Kring, "Mitigating bunching with bus-following models and bus-to-bus cooperation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2637–2646, 2021.

[17] Y. Bie, X. Xiong, Y. Yan, and X. Qu, "Dynamic headway control for high-frequency bus line based on speed guidance and intersection signal adjustment," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 1, pp. 4–25, 2019.

[18] P. Anderson and C. F. Daganzo, "Effect of transit signal priority on bus service reliability," *Transportation Research Part B: Methodological*, vol. 132, pp. 2–14, 2020.

[19] K. Long, J. Wei, J. Gu, and X. Yang, "Headway-based multi-route transit signal priority at isolated intersection," *IEEE Access*, vol. 8, pp. 187 824–187 831, 2020.

[20] W. X. Hu, H. Ishihara, C. Chen, A. Shalaby, and B. Abdulhai, "Deep reinforcement learning two-way transit signal priority algorithm for optimizing headway adherence and speed," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 7920–7931, 2023.

[21] A. Sun and M. Hickman, "The real-time stop–skipping problem," *Journal of Intelligent Transportation Systems*, vol. 9, no. 2, pp. 91–109, 2005.

[22] A. Petit, C. Lei, and Y. Ouyang, "Multiline bus bunching control via vehicle substitution," *Transportation Research Part B: Methodological*, vol. 126, pp. 68–86, 2019.

[23] A. Abdelhalim and M. Abbas, "A value proposition of cooperative bus-holding transit signal priority strategy in connected and automated vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–6, 2021.

[24] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. O. Wu, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, 2020.

[25] M. Noaeen, A. Naik, L. Goodman, J. Crebo, T. Abrar, Z. S. H. Abad, A. L. C. Bazzan, and B. Far, "Reinforcement learning in urban network traffic signal control: A systematic literature review," *Expert Systems with Applications*, vol. 199, 2022.

[26] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, "Bridging the gap between value and policy based reinforcement learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 2772–2782.

[27] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[28] K. Ling and A. Shalaby, "Automated transit headway control via adaptive signal priority," *Journal of Advanced Transportation*, vol. 38, no. 1, pp. 45–67, 2003.

[29] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[30] S. M. Alizadeh Shabestray and B. Abdulhai, "Multimodal intelligent deep (mind) traffic signal controller," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4532–4539.

[31] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

[32] S. G. Rizzo, G. Vantini, and S. Chawla, "Reinforcement learning with explainability for traffic signal control," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3567–3572.

[33] J. Guo, L. Cheng, and S. Wang, "Cotv: Cooperative control for traffic light signals and connected autonomous vehicles using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10 501–10 512, 2023.

[34] G. Guo and Y. P. Wang, "An integrated mpc and deep reinforcement learning approach to trams-priority active signal control," *Control Engineering Practice*, vol. 110, 2021.

[35] D. Fagan and R. Meier, "Dynamic multi-agent reinforcement learning for control optimization," in *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, 2014, pp. 99–104.

[36] M. Wang, L. Wu, J. Li, and L. He, "Traffic signal control with reinforcement learning based on region-aware cooperative strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6774–6785, 2022.

[37] M. Long, X. Zou, Y. Zhou, and E. Chung, "Deep reinforcement learning for transit signal priority in a connected environment," *Transportation Research Part C: Emerging Technologies*, vol. 142, 2022.

[38] P. K.J., H. K. A.N, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 2529–2534.

[39] H. Su, Y. D. Zhong, J. Y. J. Chow, B. Dey, and L. Jin, "Emvlight: A multi-agent reinforcement learning framework for an emergency vehicle decentralized routing and traffic signal control system," *Transportation Research Part C: Emerging Technologies*, vol. 146, 2023.

[40] A. Chang, Y. Ji, C. Wang, and Y. Bie, "Cvdmarl: A communication-enhanced value decomposition multi-agent reinforcement learning traffic signal control method," *Sustainability*, vol. 16, no. 5, 2024.

[41] L. Yan, L. Zhu, K. Song, Z. Yuan, Y. Yan, Y. Tang, and C. Peng, "Graph cooperation deep reinforcement learning for ecological urban traffic signal control," *Applied Intelligence*, vol. 53, no. 6, pp. 6248–6265, 2022.

[42] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, 2021. [Online]. Available: https://dx.doi.org/10.1007/s10462-021-09996-w

[43] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2687–2700, 2020.

[44] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *arXiv pre-print server*, 2020.

[45] L. J. Basso, C. A. Guevara, A. Gschwender, and M. Fuster, "Congestion pricing, transit subsidies and dedicated bus lanes: Efficient and practical solutions to congestion," *Transport Policy*, vol. 18, no. 5, pp. 676–684, 2011.

[46] D. Tsitsokas, A. Kouvelas, and N. Geroliminis, "Modeling and optimization of dedicated bus lanes space allocation in large networks with dynamic congestion," *Transportation Research Part C: Emerging Technologies*, vol. 127, p. 103082, 2021.

[47] B. Cesme, S. Z. Altun, and B. Lane, "Queue jump lane, transit signal priority, and stop location evaluation of transit preferential treatments using microsimulation," *Transportation Research Record*, vol. 2533, no. 1, pp. 39–49, 2015.

[48] A. V. Kwami, Y. X. Kuan, and X. Zhi, "Effect of bus bays on capacity of curb lanes," *Journal of American Science*, vol. 5, no. 2, pp. 107–118, 2009.

[49] J. Zhao and X. Zhou, "Improving the operational efficiency of buses with dynamic use of exclusive bus lane at isolated intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 642–653, 2019.

[50] M. Xie, M. Winsor, T. Ma, A. Rau, F. Busch, and C. Antoniou, "Parameter sensitivity analysis of a cooperative dynamic bus lane system with connected vehicles," *Transportation Research Record*, vol. 2676, no. 1, pp. 311–323, 2022.

[51] J. Wang and L. Sun, "Robust dynamic bus control: a distributional multi-agent reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 4075–4088, 2023.

[52] Q. Nie, J. Ou, H. Zhang, J. Lu, S. Li, and H. Shi, "A robust integrated multi-strategy bus control system via deep reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 133, 2024.

[53] Y. Lin, X. Yang, and N. Zou, "Passive transit signal priority for high transit demand: model formulation and strategy selection," *Transportation Letters*, vol. 11, no. 3, pp. 119–129, 2019.

[54] A. Girijan, L. D. Vanajakshi, and B. R. Chilukuri, "Dynamic thresholds identification for green extension and red truncation strategies for bus priority," *IEEE Access*, vol. 9, pp. 64 291–64 305, 2021.

[55] W. H. Lee and H. C. Wang, "A person-based adaptive traffic signal control method with cooperative transit signal priority," *Journal of Advanced Transportation*, vol. 2022, pp. 1–17, 2022.

[56] P. Mirchandani, A. Knyazyan, L. Head, and W. Wu, "An approach towards the integration of bus priority, traffic adaptive signal control, and bus information/scheduling systems," *Computer-Aided Scheduling of Public Transport*, pp. 319–334, 2001.

[57] J. Yu, P.-A. Laharotte, Y. Han, and L. Leclercq, "Decentralized signal control for multi-modal traffic network: A deep reinforcement learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 154, 2023.

[58] C. E. Cortés, J. Gibson, A. Gschwender, M. Munizaga, and M. Zúñiga, "Commercial bus speed diagnosis based on gps-monitored data," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 695–707, 2011.

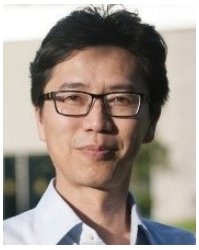
[59] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–33, 2015.

[60] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[61] OpenAI, "Spinning Up in Deep Reinforcement Learning," https://github.com/openai/spinningup, accessed: 11 09, 2024.

[62] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson, "The StarCraft Multi-Agent Challenge," *CoRR*, vol. abs/1902.04043, 2019.

Meng Long received the B.Eng. and M.S. degrees in transportation engineering from Dalian University of Technology in 2017 and 2020, respectively, and the Ph.D. degree in Future Mobility System at the Department of Electrical and Electronic Engineering of The Hong Kong Polytechnic University in 2024. She also visited the ITS Center at the University of Tokyo as an international visiting researcher for half a year in 2022. She is currently a Lecturer at the National Center for Applied Mathematics of Chongqing Normal University. Her current research interests include traffic signal control and artificial intelligence algorithms.



Edward Chung is a Professor of Intelligent Transport Systems (ITS) at the Department of Electrical and Electronic Engineering of The Hong Kong Polytechnic University. He received a Bachelor of Civil Engineering with Honours and Ph.D. from Monash University. He held positions such as Senior Research Scientist at the Australian Road Research Board, Visiting Professor at the Centre for Collaborative Research, University of Tokyo, Head of the ITS Group at LAVOC, EPFL, Switzerland, Professor at the Queensland University of Technology (QUT) and Director of the Smart Transport Research Centre at QUT.