Transportation Research Part C 169 (2024) 104860

Contents lists available at ScienceDirect

Transportation Research Part C

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

A cooperative perception based adaptive signal control under early deployment of connected and automated vehicles☆

Wangzhi Li, Tianheng Zhu, Yiheng Feng *

Lyles School of Civil Engineering, Purdue University, United States of America

ARTICLE INFO

Keywords:
Traffic signal control
Deep reinforcement learning
Connected and automated vehicles
Cooperative perception
Cell transmission model

ABSTRACT

Connected vehicle-based adaptive traffic signal control requires certain market penetration rates (MPRs) to be effective, usually exceeding 10%. Cooperative perception based on connected and automated vehicles (CAVs) can effectively improve overall data collection efficiency and reduce required MPR. However, the distribution of observed vehicles under cooperative perception is highly skewed and imbalanced, especially under very low CAV MPRs (e.g., 1%). To address this challenge, this paper proposes a novel deep reinforcement learning-based adaptive traffic signal control (RL-TSC) method that integrates a traffic flow model, known as the cell transmission model (CTM), denoted as CAVLight. Traffic states estimated from the CTM are integrated with the data collected from the cooperative perception environment to update the states in the CAVLight model. The design of reward function aims for reducing total vehicle delays and stabilizing agent behaviors. Extensive numerical experiments under a real-world intersection with varying traffic demand levels and CAV MPRs are conducted to compare the performance of CAVLight and other benchmark algorithms, including a fixed-time controller, an actuated controller, the max pressure model, and an optimization-based adaptive TSC. Results demonstrate the superiority of CAVLight in performance and generalizability over benchmarks, especially under 1% CAV MPR scenario with high traffic demands. The influence of CTM integration on CAVLight is further explored through RL agent policy visualization and sensitivity analysis in CTM parameters and CAV perception capabilities (i.e., detection range and detection accuracy).

1. Introduction

Connected Vehicle (CV) technology demonstrates the advantages of real-time provision of spatial–temporal vehicle trajectory information compared with infrastructure-based sensors, which are limited to specific locations and often entail significant installation and maintenance costs. CV-based adaptive signal control has gained increasing attention from researchers (Goodall et al., 2013; Guler et al., 2014; Feng et al., 2015; Guo et al., 2019; Li and Ban, 2020; Zhang et al., 2020; Wang et al., 2024) in the past decade. Since a full CV market penetration rate (MPR) cannot be achieved shortly, two main categories of strategies are designed to address this challenge.

☆ This article belongs to the Virtual Special Issue on "VSI:MLCATS".

* Correspondence to: 550 Stadium Mall Drive, West Lafayette, IN 47907, United States of America.
E-mail address: feng333@purdue.edu (Y. Feng).

https://doi.org/10.1016/j.trc.2024.104860
Received 17 August 2023; Received in revised form 14 September 2024; Accepted 16 September 2024

Model-based approaches usually consider traffic state estimation as the first step and then utilize estimated complete traffic information as input to the signal control model. Objective and constraints calculation in optimization models require accurate estimation of the traffic state, either microscopic level or macroscopic level. The microscopic traffic state estimation methods infer the status of each unobserved vehicle (denoted as *legacy vehicle*) based on the observed CV trajectories. For example, Feng et al. (2015) separated the road segments near the intersections into three regions based on observed CV states, namely queuing, slow-down, and free-flow regions. Then, legacy vehicles were inserted into each region so as to estimate the overall traffic state for the proposed TSC system. Wang et al. (2021) further developed the idea and predict vehicle arrivals based on the calculated traffic volume of an intersection with estimated CV MPR and observed CV flow rate. Besides the traffic flow models, machine learning technique such as imitation learning is also applied to predict vehicle trajectories (Ying and Feng, 2024). The macroscopic traffic state estimation methods generate aggregated traffic information such as queue length (Zhao et al., 2019), traffic volume (Zhang et al., 2022), or vehicle density (Al Islam et al., 2020) from observed CV data.

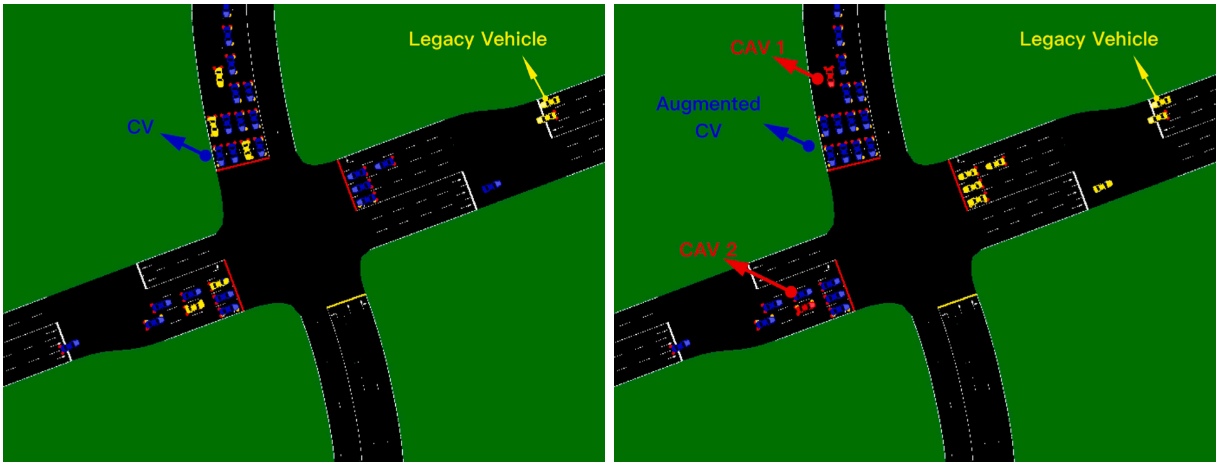
On the other hand, learning-based approaches typically skip the traffic estimation step and rely on the deep neural network to find the mapping relations between the partially observed traffic data and the control (policy) output. The learning-based approaches are typically formulated as a reinforcement learning problem, which is proved to be robust against partial connectivity scenarios by design (Zhang et al., 2020; Wu et al., 2020; Song and Fan, 2021; Mo et al., 2022). For example, Wu et al. (2020) proposed a reinforcement learning-based traffic signal control (RL-TSC) with recurrent neural network (RNN) layers to learn historical traffic data and therefore enhanced the robustness against partial observation with a 20% CV MPR. Some researchers also aim to combine the learning-based method and model-based adaptive traffic signal control algorithm. More recently, Wang et al. (2022) proposed a learning-based approach to tune a max-pressure algorithm, a model-based adaptive traffic signal control (ATSC) method.

However, no matter model-based or learning-based approaches, the effective implementation of real-time control necessitates a critical CV market penetration rate (MPR), typically exceeding 10% (Feng et al., 2018; Guo and Ma, 2021; Wang et al., 2021; Mo et al., 2022; Shabestary and Abdulhai, 2022). Such a CV MPR may not be achieved in the near future. Leveraging perception and communication capabilities of connected and automated vehicles (CAVs), recent research has shown the effectiveness of cooperative perception, utilizing the collected and shared traffic data from CAVs' perception sensors, e.g., cameras and LiDAR, to achieve similar traffic data quality even under very low CAV MPRs (Li et al., 2021; Chen et al., 2022; Cao et al., 2022). The cooperative perception-based data collection has the potential to enable adaptive signal control in a very early-stage deployment of CAVs. Note that in this paper, the term CAV is used to represent any vehicles (not necessary to be automated) that are capable of status-sharing cooperation, defined in Society of Automotive Engineers (SAE) J3216 standard (SAE International, 2020). It means that vehicles can share the perception information about the traffic environment including themselves.

Current research on cooperative perception mainly focuses on improving road safety (Caillot et al., 2022). A limited number of works are found in utilizing cooperative perception for traffic state estimation and control, especially at signalized intersections. Li et al. (2021) designed a cooperative perception framework to estimate and predict the microscopic traffic states, where a particle filtering algorithm was adopted to estimate the distribution of the non-observable vehicles at the link level. Results show that the accuracy of position and speed estimation could reach 50%–90% when CAV MPR is 12.5%. Chen et al. (2022) developed a cooperative perception co-simulation platform based on SUMO and CARLA. A max-pressure TSC method was applied to show equal performance under cooperative perception with CAV MPR below 5% and pure CV environment with above 30% CV MPRs. However, one infrastructure LiDAR sensor is assumed to be installed at the intersection, which may not be the case in real-world situations. One recent research developed a learning-based TSC framework that estimates traffic state with CAV perception (Guo and Ma, 2021). The proposed state estimation model requires information from pairs of CAVs or CAVs and observed vehicles and thus needs a relatively high CAV MPR (e.g., 10%).

Previous works also use connected vehicle information to estimate traffic state (Liu et al., 2019; Wang et al., 2024). However, the cooperative perception scenario with a limited number of CAVs is significantly different from the connected vehicle environment in terms of data patterns. Although in the cooperative perception environment, a very low CAV MRP can achieve good coverage on average, it is critical to acknowledge that the observed traffic condition or collected traffic data are highly unstable and skewed, contingent upon the presence and distribution of the limited number of CAVs. For example, the detected vehicles, denoted as *augmented CVs*, are clustered around the CAV (under cooperative perception) rather than randomly distributed across the road network (under CV environment), given the same number of legacy vehicles are detected, as shown in Fig. 1. In Fig. 1(b), vehicles within the detection range of a CAV are denoted as augmented CVs (i.e., observed vehicles). Vehicles that are neither CAVs nor detected are denoted as legacy vehicles. This substantial variability in traffic data poses a significant challenge to the stability of current traffic signal control (TSC) strategies, as it undermines the prerequisite of accurate traffic state estimation relying on traffic flow models or well-trained neural networks. Therefore, directly applying the traffic state estimation and TSC models designed for a low CV MPR scenario may not work under the equivalent CAV cooperative perception situation.

To the best of our knowledge, there are very limited studies on solving the high-variance traffic data challenge in the cooperative perception environment and successfully apply real-time adaptive traffic signal control in extremely low CAV MRPs (e.g., ≤3%). In this paper, we aim to address this challenge by combining a traffic flow model, known as the cell transmission model (CTM) (Daganzo, 1994, 1995), with a deep reinforcement learning (DRL)-based TSC model. Previous research on combining RL-TSC with a traffic model is mainly about improving training efficiency (Korecki and Helbing, 2022; Xiao et al., 2022) or optimizing action space (Zhang et al., 2022). One recent work designed a physic-informed RL state based on CTM (Liang et al., 2022) where the



(a) Pure CVs environment

(b) Cooperative perception with CAVs environment

Fig. 1. An example of a pure CVs environment and a cooperative perception with CAVs environment, given the same number of legacy vehicles but different distributions.

authors divided incoming lanes into cells containing density as well as average speed information. Such a design takes advantage of balancing the representation ability and learning complexity, but the traffic flow propagation logic in CTM is not directly applied. In this work, by combining observed real-time surrounding vehicle information and update from the CTM, we can improve the traffic state transition accuracy and stability for the downstream DRL model in extremely low CAV MPR environments. This work adopts an RL-TSC framework as it is proven to be effective and robust against missing data in partial connectivity scenarios in our previous work (Mo et al., 2022). We test the proposed method under varying CAV MRPs from 1% to 50% in microscopic simulation. The performance in terms of average vehicle delay outperforms all baselines including fixed time control, actuated control, model-based adaptive control, and DRL-based adaptive control without integrating the traffic flow model. Especially, the proposed method works well under even 1% CAV MPR.

We claim the following contributions in this paper:

1. This is the first study, to the best of our knowledge, that implements an RL-TSC method, denoted as CAVLight, specifically designed for an early-stage deployment of CAV MPR as low as 1%. We also explore the RL agent state and reward design considering the availability and patterns of traffic data collected from the cooperative perception environment.
2. We introduce the CTM into RL state estimation. By updating CTM estimation given the real-time cooperative perception result, the prediction can be adjusted and used to stabilize RL agent state and therefore improve its performance. This work demonstrates the advantage of combining transportation domain models with learning-based traffic signal control methods.
3. We compare the performance of the proposed RL-TSC method with that of several prevailing TSC methods, including a fixed-time, an actuated, the max-pressure and an optimization-based adaptive TSC method. The adaptive TSC method, namely I-SIG, is also tested in the cooperative perception environment with the same CTM-based traffic state estimation. Such comparison experiments provide valuable insights on the differences between learning-based and optimization-based signal optimization models.

The remainder of this paper is organized as follows. Section 2 presents the CAVLight agent design, algorithm, and CTM-based state estimation method. Numerical experiment settings are introduced and results are analyzed in Section 3. We further discuss the experiment result under a realistic CAV detection model and an improved CTM in Section 4. Finally, conclusions and future works are delivered in Section 5.

2. Methodology

In this section, we first briefly introduce the concepts and notations of RL-TSC, following with RL agent design of CAVLight and the asymmetric advantage actor-critic (Asym-A2C) algorithm. Then we discuss the necessity of combining a traffic flow model into the state estimation process with a CTM-based state estimation method.

Before presenting the model, major notations are introduced below.

Major Notations

Reinforcement Learning

$\mathbf{s}$	State
S	State space
a	Action
A	Action space
o	Observation
O	Observation space
r	Reward
R	Reward space
Pr	State transition function
γ	Discounted factor
π	Policy
$Q(s,a)$	State–action value function
$V(s)$	State value function
θ	Parameters of critic's value neural network
ϕ	Parameters of actor's policy neural network

Traffic Signal Control

P	Set of all feasible phases
J_p	Set of segments of incoming lanes controlled by signal phase p
$L_{in,p}$ or $L_{out,p}$	Set of incoming (outgoing) lanes of phase p
p_t	Current signal phase at time t
d_t	Current phase elapse time at time t
n_t^{in} or n_t^{out}	Number of vehicles on all incoming (outgoing) lanes at time t
$\Delta\bar{v}_{j,p,t}^{in}$	Average vehicle speed difference at segment j of incoming lanes controlled by phase p at time t
M_t^{in}	Set of vehicles on all incoming lanes at time t
$x_{m,t}$	Travel delay of vehicle m at time t
c	Radius of the study area
d_m	Distance between vehicle m and intersection center
$n_i(t)$	Number of vehicles in cell i at time t in CTM
$y_{i-1,i}(t)$	Number of vehicles from cell $i-1$ enter cell i at time t in CTM

2.1. Preliminaries: RL-TSC

RL is a machine learning method to learn the optimal policy mapping from states to actions via iterative interactions between an intelligent agent and the environment. Detailed information about RL can be found in Sutton et al. (1998). In the context of traffic control, we regard the traffic signal controller as an agent and the traffic composed of vehicles or other transportation participants as the environment. Such a dynamic environment can be considered as a Markov decision process (MDP) specified by a tuple of (S, A, R, Pr, γ). Where, S is the state space that contains all necessary information to describe the environment, for example, queue length in each incoming lane as well as the current traffic signal phase. A is the action space that defines a set of actions an agent can take to interact with the environment, e.g., choosing the next phase. R is the reward function defined on state and action space to evaluate the performance of the action, for instance, the difference in queue length between consecutive time steps. Pr is the transition probability function that models the environment's change when a certain action is chosen. γ is a discount factor.

As shown in Fig. 2, given a certain traffic state $s_t \in S$ at time t, the RL-TSC agent chooses its action $a_t \in A$, following the learned policy $\pi(a_t|s_t)$ which is a probability distribution mapping from $s_t \in S$ to $a_t \in A$. Once the chosen action a_t, for example, the next signal phase is executed, the traffic environment will transit from state s_t to the next state $s_{t+1} \in S$ based on Pr, and a reward $r_t \in R$ is generated to evaluate the action. The goal of RL is to find, through iterative interactions, the optimal policy $\pi^*(a|s)$ that can maximize the return. The return is a function of a sequence of expected future rewards weighted by the discount factor γ and two value functions, state value function $V^\pi(s)$ and state–action value function $Q^\pi(s,a)$, are used to estimate the expected return given a certain circumstance. For instance, $V^\pi(s)$ describes the expected future return when the agent is at state s following policy π. For a complicated real-world transportation system, the exact transition model Pr is usually unknown, and therefore temporal difference (TD) techniques, e.g. Q-learning, are usually applied to solve the MDP problem.

In the cooperative perception environment with low CAV MPRs, the TSC agent cannot observe the ground truth traffic information, but only partial information. In this case, the MDP problem becomes a partially observable Markov decision process (POMDP) and the observation is denoted as $o \in O$ where O is the observation space and $O \subseteq S$. When the CAV MPR is exceptionally low, the observations (o) can be significantly different from the true states (s), which could mislead the RL-TSC agent in the decision-making process. To cope with this issue, we propose CAVLight and develop a CTM-based state estimation method to close the difference between o and s.

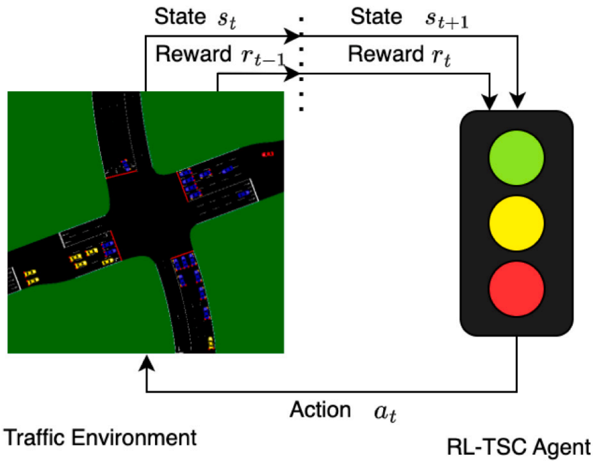


Fig. 2. An illustration of the interaction between RL-TSC agent and environment.

2.2. CAVLight agent design

State. At time step t, the state s_t is represented as a vector of the current phase index p_t ($p_t \in P$ where P is the set of all available phases), the ratio of phase elapse time over maximum green time d_t, the total number of vehicles $n^{in}_{j,p,t}$ on segment j of all incoming lanes $l \in L_{in,p}$ of corresponding phases p ($L_{in,p}$ is the set of incoming lanes of phase p), the total number of vehicles $n^{out}_{p,t}$ in all outgoing lanes $l \in L_{out,p}$ ($L_{out,p}$ is the set of outgoing lanes of phase p) of phase p, and the average speed difference $\Delta\bar{v}^{in}_{j,p,t}$ of all vehicles on segment j ($j \in J_p$ where J_p is the set of segments of incoming lanes controlled by phase p) of all incoming lane $l \in L_{in,p}$ of phase p. Specifically, the state s_t is written as below:

$$s_t = [\ \overbrace{p_t}^{\text{current phase}}\ ,\ \underbrace{d_t}_{\substack{\text{current phase}\\ \text{elapse time ratio}}}\ ,\ \overbrace{n^{in}_{j,p,t}}^{\substack{\text{number of vehicles}\\ \text{in incoming lanes}}}\ ,\ \underbrace{n^{out}_{p,t}}_{\substack{\text{number of vehicles}\\ \text{in outgoing lanes}}}\ ,\ \overbrace{\Delta\bar{v}^{in}_{j,p,t}}^{\substack{\text{average speed difference}\\ \text{in incoming lanes}}}\ ],\ for\ p \in P, j \in J_p. \tag{2.1}$$

Now we describe how we obtain the value of each component.

- The phase p_t is a one-hot vector that indicates the current traffic phase index at the intersection at time t.
- Similar to Mo et al. (2022), we include phase elapse time d into the state, which enables the agent to build the connection between the phase length and the reward received. We take the ratio of phase elapse time over the maximum green time so as to scale this value in the state.
- Following the idea in PressLight (Wei et al., 2019) and Maxpressure (Varaiya, 2013), we include the number of vehicles on each segment of the incoming lanes as well as the number of vehicles in outgoing lanes as part of the state. For incoming lanes, we divide each lane into three segments to provide vehicle spatial distribution information (Yang and Menendez, 2018; Wei et al., 2019). We scale each $n^{in}_{j,p,t}$ in the state by dividing $|n^{in}_t|$, the L2 norm of the $n^{in}_{j,p,t}$ vector of each segment of each phase at t step. Also, $n^{out}_{p,t}$ is scaled by $|n^{out}_t|$, the L2 norm of the vector of $n^{out}_{p,t}$ of each phase at t step.
- Lastly, we take the average speed difference of vehicles on each segment of each phase into the state design, denoted as $\Delta\bar{v}^{in}_{j,p,t}$. The average speed difference is defined as the difference between the free flow speed and the average speed of all vehicles in a road segment at a certain time step. Note that We set 0 as the lower boundary of the average speed difference. The reason for using average speed difference is threefold: first, by using average speed difference, we can align the speed-related state vector with the number of vehicle-related vectors, i.e. the higher the number of vehicles at that segment, the higher the average speed difference. Second, vehicle speed information can be directly collected by the cooperative perception in real-time and does not require tracking. Last, CTM can easily estimate the average speed difference in the state estimation process. If no vehicle exists in a certain segment, we set the average speed of that segment as the free flow speed, and thus the average speed difference equals zero. We also normalize the average speed difference vector by its norm $|\Delta\bar{v}^{in}_t|$ and add an offset to avoid zero-value input in neural network training. In this work, the offset is empirically set to be 0.2.

Observation. The observation o_t has the same format as the state. Note that in the early CAV deployment stage, only part of the traffic state can be observed. When CTM is used to estimate the traffic state, the "observation" generated by CTM is denoted as $o^{CTM}_t \in O$.

Action. The agent action a_t at time t is defined as "choosing the next phase to be executed", which makes the signal timing plan acyclic. If the chosen phase is the same as the current phase, the current phase (1 s by default) will be extended; if the chosen phase is different, a transition phase will be inserted, including a 4-s yellow interval and a 1-s all-red clearance interval, and then the chosen phase will be executed for a minimum green time (10 s by default). A maximum green time (40 s by default) is enforced for each phase. If the maximum green time is reached and the chosen phase remains the same as the current phase, the agent will be forced to randomly choose from other available phases as the next phase based on a new policy. Specifically, in training, the new policy is uniformly distributed for all other phases to encourage exploration. In testing, the new policy is updated by removing the current phase probability and recalculating the probability distribution based on the ratio of remaining phase probability values.

Reward. The reward function r is defined as a weighted sum of position-weighted phase pressure r^p, average delay r^d, and phase switching penalty r^s. Only vehicles within the study area centering at the intersection (200 m in this work) will be considered. During the training stage, we assume that the training data contains complete information about the system, and the reward is calculated based on all vehicles' states. Reward at time t, r_t, is written as below:

$$r_t = -\alpha r_t^d - \beta r_t^p - (1 - \alpha - \beta) r_t^s \tag{2.2}$$

where

- α and β are the weights, which are set as 0.7 and 0.2 in this work based on empirical results (i.e. performance comparison among other value pairs);
- r_t^d is the average delay of all vehicles at this intersection, as defined in Eq. (2.3);
- r_t^p is the position-weighted pressure at this intersection, specified in Eq. (2.5);
- r_t^s is the penalty item for phase switching. If the newly selected phase is not the same as the current phase, the r_t^s is set to be a constant, which is empirically set to be 2 in this work. Otherwise, it will be set as 0.

Average delay is used in the reward as one of the main objectives for the proposed RL-TSC and is calculated as below:

$$r_t^d = \sigma^d \frac{\sum_{m \in M_t^{in}} x_{m,t}}{n_t^{in}}, \tag{2.3}$$

where

- σ^d is a scale factor, which is set as 0.001 in this work;
- M_t^{in} is the set of all vehicles on incoming lanes to this intersection at time t;
- $x_{m,t}$ is the delay of vehicle m at current intersection at time t;
- n_t^{in} is the number of vehicles on all incoming lanes of this intersection at time t.

The delay of vehicle m at time t, $x_{m,t}$ is calculated as below:

$$x_{m,t} = \sum_{t \in T} \Delta t (1 - \frac{v_{m,t}}{v_{m,max}}), \tag{2.4}$$

where

- T is the time duration (until time t) after the vehicle enters the study area
- Δt is the step length, which is 1 s in this paper;
- $v_{m,t}$ is the speed of vehicle m at time t;
- $v_{m,max}$ is the max speed of vehicle m.

We also adopt the idea of position-weighted pressure (Li and Jabari, 2019) in the reward calculation. The closer a vehicle is to the intersection, the more weight it will contribute to the pressure calculation. As illustrated in Fig. 3, the weight for a certain vehicle m is calculated as the difference between the study area radius c and its distance to the stop bar d_m. The formulation is specified below:

$$r_t^p = \sigma^p \left| \sum_{p \in P} \left(\frac{\sum_{m \in M_{p,t}^{in}} (c - d_m)}{c} - \frac{\sum_{m \in M_{p,t}^{out}} (c - d_m)}{c} \right) \right|, \tag{2.5}$$

where

- σ^p is a scale factor, which is set as 10 in this work;
- $M_{p,t}^{in}$ (or $M_{p,t}^{out}$) is the set of vehicles on incoming (or outgoing) lanes of phase p at time t;
- c is the radius of the study area;
- d_m is the distance between vehicle m and the stop bar.

2.3. Asymmetric advantage actor-critic algorithm

The asymmetric advantage actor-critic (Asym-A2C) algorithm is proposed in our previous work (Mo et al., 2022), which is an asymmetric variant of the advantage actor-critic algorithm (Mnih et al., 2016). The basic idea of Asym-A2C is to provide the actor and critic with asymmetric input (the actor gets the observation, and the critic gets the state), that allows the actor to generate a

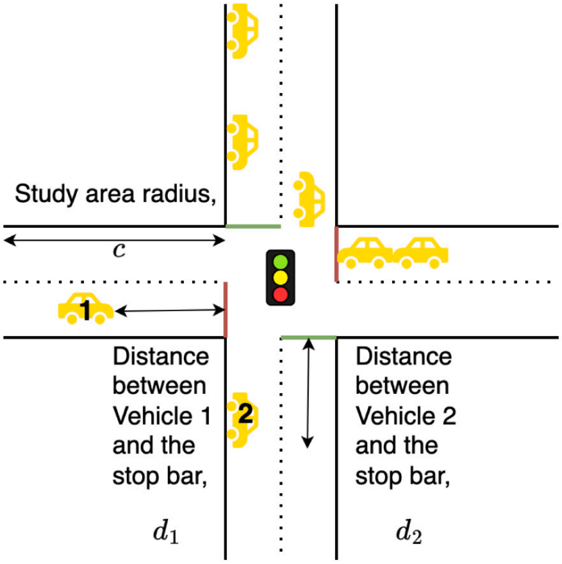

Study area radius,
c
1
2
Distance between Vehicle 1 and the stop bar,
d_1
Distance between Vehicle 2 and the stop bar,
d_2

Fig. 3. An illustration of position-weighted pressure.

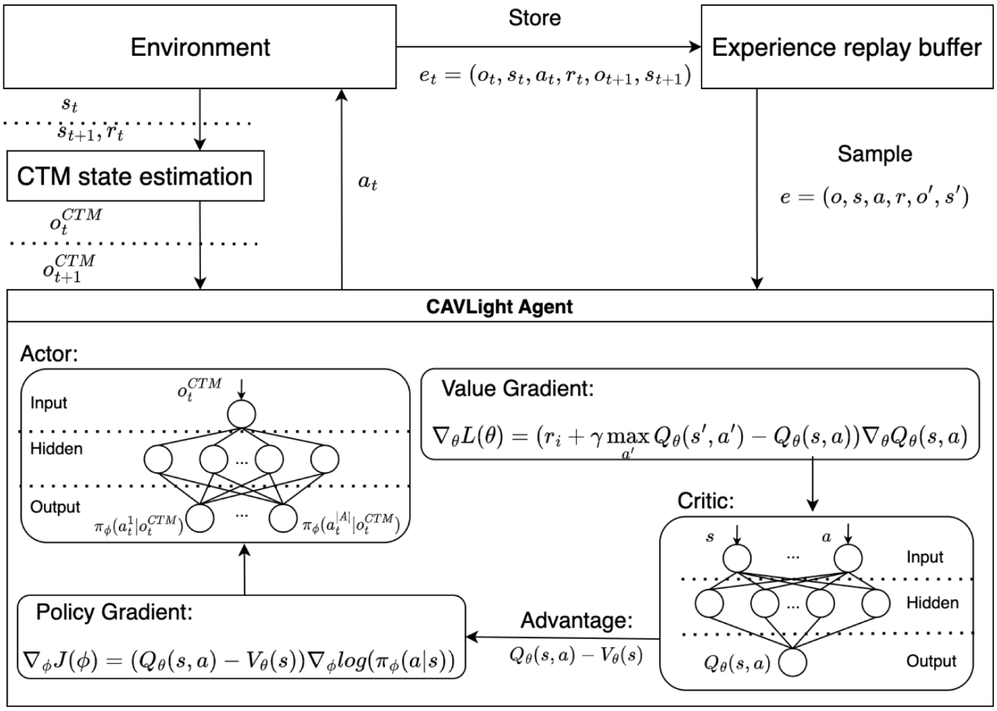

Environment
Store
$e_t = (o_t, s_t, a_t, r_t, o_{t+1}, s_{t+1})$
Experience replay buffer
s_t
s_{t+1}, r_t
CTM state estimation
o_t^{CTM}
o_{t+1}^{CTM}
a_t
Sample
$e = (o, s, a, r, o', s')$
CAVLight Agent
Actor:
Input
Hidden
Output
o_t^{CTM}
$\pi_\phi(a_t^1|o_t^{CTM})$
$\pi_\phi(a_t^{|A|}|o_t^{CTM})$
Value Gradient:
$\nabla_\theta L(\theta) = (r_i + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s,a))\nabla_\theta Q_\theta(s,a)$
Critic:
s
a
Input
Hidden
Output
$Q_\theta(s,a)$
Policy Gradient:
$\nabla_\phi J(\phi) = (Q_\theta(s,a) - V_\theta(s))\nabla_\phi log(\pi_\phi(a|s))$
Advantage:
$Q_\theta(s,a) - V_\theta(s)$

Fig. 4. An illustration of Asym-A2C for CAVLight.

policy based on its observation while considering the ground truth state. This algorithm shows advantages in the case when the actor can only observe partial or inaccurate information while the critic has access to the ground truth state. In this work, the CTM estimation can deviate from the ground truth state and therefore we choose to use Asym-A2C and embed the proposed CTM estimation into the actor state generation process.

Fig. 4 details the Asym-A2C implementation in this work (Mo et al., 2022; Shou and Di, 2020). Deep neural networks (DNN) are used to estimate the value function of the critic and the policy model of the actor. The critic DNN, parameterized by θ, estimates the value function, and the actor updates its DNN parameter ϕ based on the guidance from the critic. For an isolated intersection

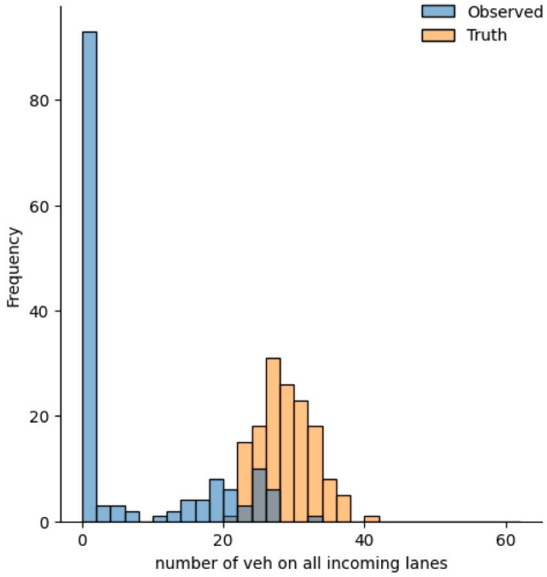


Fig. 5. Comparison between the ground truth and the observed number of vehicles on incoming lanes based on cooperative perception with a 1% CAV MPR.

controlled by the proposed CAVLight agent, at time t, the estimation of the true traffic state o_t^{CTM} is generated by the CTM-based state estimation method after receiving the CAV cooperative perception information. Then, o_t^{CTM} is fed into the actor, and one action a_t is chosen from the action space A based on the generated policy π_ϕ. After executing action a_t, the environment will generate a new traffic state s_{t+1} and a reward r_t. An experience tuple $e_t = (o_t, s_t, a, o_{t+1}, s_{t+1})$ will then be sent to the experience replay buffer. During the offline training process, experience tuples will be drawn uniformly from the data set $D = \{e_1, \dots, e_t\}$, denoted as $e = (o, s, a, o', s') \sim U(D)$. We assume that the critic can get the ground truth information during the offline training. In other words, the input to the critic is state s, and the reward r is calculated based on the ground truth traffic status. We stabilize learning further by using the Advantage function $A(s, a) = Q(s, a) - V(s)$. The critic network θ is then updated using the gradient $\nabla_\theta L(\theta)$ in Eq. (2.6):

$$\nabla_\theta L(\theta) = \mathbb{E}_{(o,s,a,r,o',s')\sim U(D)}[(r + \gamma \max_{a'} Q_{\theta_i}(s', a') - Q_\theta(s, a))\nabla_\theta Q_\theta(s, a)] \tag{2.6}$$

The policy network ϕ is updated using the gradient $\nabla_\phi J(\phi)$ in Eq. (2.7):

$$\nabla_\phi J(\phi) = \mathbb{E}_{(o,s,a,r,o',s')\sim U(D)}[\nabla_\phi \log \pi_\phi(a_i|o_{i,CV})(Q_\theta(s, a) - V_\theta(s))], \tag{2.7}$$

2.4. Cell transmission model for state estimation

The proposed Asym-A2C algorithm works well when observations are close to true states. However, under extreme scenarios (e.g., 1% CAV MRP) when observations are significantly different from true states, the proposed model fails to converge. An illustration is shown in Fig. 5. The x-axis is the number of vehicles on incoming lanes to the intersection. The y-axis is the frequency across one simulation experiment (1 h simulation time). The figure shows that the distribution of the observed number of vehicles on incoming lanes is significantly underestimated. More importantly, due to the random arrival of limited CAVs, the observed areas are highly random. One approach with CAV(s) may get full coverage while other approaches may have zero coverage (i.e., very high frequency of no observations in Fig. 5). Therefore, in such cases, solely relying on observations from cooperative perception cannot provide accurate information for the RL agent to make decisions. To solve this challenge, we propose to integrate a traffic flow model with the CAV observations to better estimate the traffic states. In this work, CTM is chosen for two reasons. First, CTM is a widely accepted macroscopic traffic model used to capture the dynamic of traffic. Second, the CTM framework fits the proposed RL-TSC method well as the model can directly generate the number of vehicles and speed difference which are states defined in the RL model.

CTM is the first-order Lighthill–Whitham–Richards (LWR) partial differential equation approximation. In CTM, the traffic flow propagation process is discretized into homogeneous cells in terms of space and uniform interval in terms of time. A triangular fundamental diagram (TFD) is assumed. Given a certain traffic density k, the corresponding traffic flow rate q can be calculated as:

$$q = min\{v_f k, q_{max}, w(k_{jam} - k)\}, \tag{2.8}$$

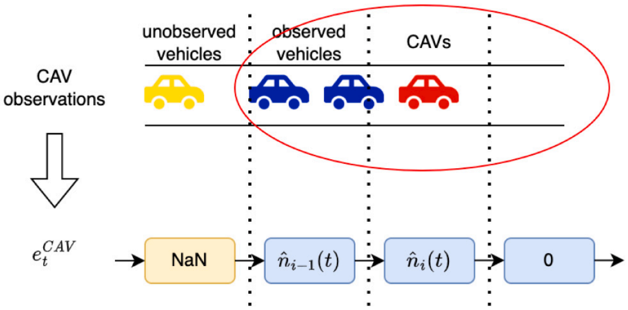


Fig. 6. The generation process of cell-based CAV observations (the NaN in unobserved cells represents *not available*).

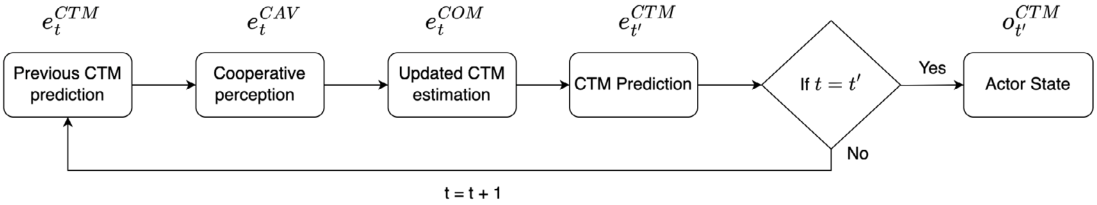


Fig. 7. CTM for state estimation.

where q_{max} is the maximum flow rate, k_{jam} is the jam density, v_f is the free flow speed, and w is the shockwave speed. The density corresponding to q_{max} is denoted as critical density k_{cri}.

Originally developed for highway traffic flow, the CTM model is extended to model traffic flows at signalized intersections (Daganzo, 1995) with six cell types: ordinary cell, merging cell, diverging cell, intersection cell, source cell, and sink cell. An ordinary cell has one preceding and one following cell, with limited capacity and jam density. A merging cell has multiple preceding cells and one following cell, while a diverging cell has multiple following cells and one preceding cell. Both merging and diverging cells have limited capacity and jam density as well. An intersection cell is similar to an ordinary cell but its outflow is controlled by the corresponding traffic signal. A source cell has no preceding cell as it generates vehicles, and a sink cell has no following cell as it exits vehicles. Both of them have unlimited capacity.

We take an ordinary cell i as an example to illustrate the evolution of the CTM model. More details can be found in Daganzo (1994, 1995). The number of vehicles in cell i at time $t+1$ is represented as:

$$n_i(t+1) = n_i(t) + y_{i-1,i}(t) - y_{i,i+1}(t), \tag{2.9}$$

where $n_i(t)$ is the number of vehicles in cell i at time t and $y_{i-1,i}(t)$ is the number of vehicles entering cell i from cell $i-1$ at time t. Specifically, $y_{i-1,i}(t)$ can be calculated according to Eq. (2.10):

$$y_{i-1,i}(t) = min\{n_{i-1}(t), min(Q_{i-1}, Q_i), \delta(N_i - n_i(t))\}, \tag{2.10}$$

where Q_i is the flow capacity of cell i, δ is calculated as $\delta = -w/v_f$, and N_i is the maximum number of vehicles cell i can store.

Now we introduce the integration of CTM-based traffic state estimation with cooperative perception. An intersection CTM model is constructed first and loop detectors located at the entrance of each approach are used to generate input demands to CTM source cells. The flows at intersection cells are controlled by the actions made by the RL agent. With the input demand and control, the CTM model can update the number of vehicles in each cell at every time step dynamically, denoted as $e_t^{CTM} = (n_0(t), n_1(t), \ldots, n_i(t), \ldots, n_N(t))$, where N is the total number of cells. Then at each time step t, the observation areas from CAVs will be projected to the CTM cells as shown in Fig. 6.

Given the CAVs' locations, all CTM cells will be categorized into observed cells (blue color) and unobserved cells (yellow color). Specifically, based on the detection range, we assume a number of neighboring cells ahead and behind the cell with CAVs can be observed. For each observed cell i, the number of vehicles derived from CTM $n_i(t)$ will be replaced by the real observations from CAV, denoted as $\hat{n}_i(t)$ (note that $\hat{n}_i(t)$ can be zero if no vehicle exists). For unobserved cells, the number of vehicles remains the same as calculated by the CTM model. After the replacement of the number of vehicles in all observed cells, the CTM will keep updating until the next decision-making time point and provide the CTM-based observations $o_{t'}^{CTM}$, which is then fed into the actor network. Note that t' could be equal to $t+1$ when the minimum green time has passed and the actor decides to keep the current phase. t' could also take multiple time steps including the transition period and minimum green time when the actor decides to switch to another phase. The update process is illustrated in Fig. 7. Integration of the CTM can not only generate the number of vehicles in unobserved areas, but also provide more accurate state estimation in future time steps when the actor makes a decision. Especially when CAVs pass the intersection during the green phase, CTM can keep updating the estimation of the unobserved area based on

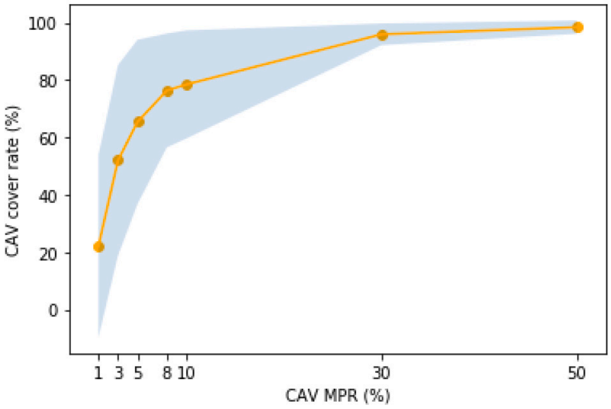


Fig. 8. CAV cooperative perception coverage under different CAV MPRs.

the observations, as long as such area or neighboring areas are within CAV's detection range. Otherwise, the CAVLight agent can solely make decisions based on the currently observed traffic state, which can result in a large deviation from the truth state if no or a limited number of CAVs exist. Note that the other state variable average speed difference can also be calculated easily from the CTM model given the number of vehicles $n_i(t)$.

3. Numerical experiments

In this section, we conduct numerical experiments under a real-world isolated intersection to demonstrate the effectiveness of the proposed CAVLight with the CTM state estimation model. We first introduce experiment settings in Section 3.1. Then, the performance between the proposed method and other state-of-the-art benchmarks are compared and presented in Section 3.2. Based on this, we further visualize and analyze the learned policy of CAVLight to demonstrate the necessity of integrating the CTM model under certain scenarios in Section 3.3. In addition, we investigate the influence of CTM accuracy on the CAVLight algorithm performance in Section 3.4.

3.1. Experiment settings

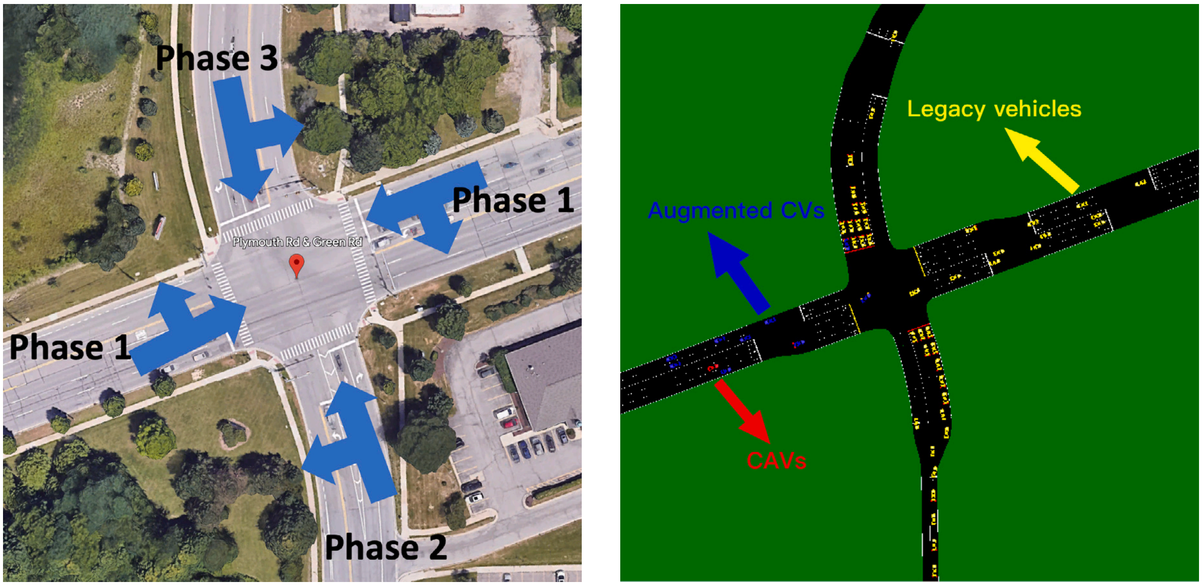
This subsection introduces the experiment settings, including the simulation environment, measurement of performance, benchmark algorithms, and parameters. Fig. 8 presents the CAV coverage (the ratio of the number of vehicles observed over the total number of vehicles in the study area) under various CAV MPRs and the real-world traffic demand in one-hour simulation. We assume the CAV detection is ideal (i.e., no occlusion or errors) for experiments in this section and will release this assumption in Section 4. The x-axis is CAV MPR and the y-axis is CAV cover rate. The orange line represents the mean cover rate at a certain CAV MPR and the blue shadow represents the corresponding standard deviation. From the figure, we can see a clear trend that higher CAV MPRs lead to higher cover rates and smaller variances. Also, the CAV coverage increases rapidly as CAV MPR increases from 1% to 10%. Since the coverage can reach almost 100% under the case with a 50% CAV MPR, the highest CAV MPR in this work is set as 50%.

3.1.1. Environment set-up

A real-world intersection of Plymouth Road and Green Road in Ann Arbor, MI is used in our numerical experiments, as shown in Fig. 9(a). We use SUMO (Behrisch et al., 2011; Lopez et al., 2018), a widely accepted open-source microscopic traffic simulation platform, to conduct the simulation experiment. The SUMO network of the same intersection is shown in Fig. 9(b). The length of each approach in the SUMO intersection is more than 300 m, while the study area radius is set to be 200 m. In other words, vehicles that are on the approach but outside the study area will not be considered.

A split 3-phase signal timing plan is used, represented by orange arrows in Fig. 9(a), similar as in real-world operations. Phase 1 is the main street (Plymouth Road) through phase with permissive left turns. Phase 2 is the Northbound green phase and phase 3 is the Southbound green phase. A 10-s minimum green time and a 40-s maximum green time are adopted. A 4-s yellow and a 1-s all-red clearance interval are enforced between each phase transition.

Each simulation run lasts for 2100s with 0.1s resolution. The first 100 s are considered as the warm-up stage when the signal is controlled by a fixed-time TSC. The vehicle arrival process follows a Poisson process generated by SUMO. Given a certain CAV MPR, we randomly decide whether a newly generated vehicle is a CAV. At a certain time step, vehicles that are within the detection range of a CAV (set as 80 m in this work) will be denoted as augmented CVs (i.e., observed vehicles). Vehicles that are neither selected as CAVs nor detected are denoted as legacy vehicles, as shown in Fig. 9(b).



(a) A real-world intersection of Plymouth Road and Green Road, Ann Arbor, MI, USA

(b) The real-world intersection in SUMO simulation

Fig. 9. Illustrations of a real-world intersection structure and its SUMO simulation, including signal phase setting.

Table 1
Traffic demand level.

Demand levels	EB[a]	WB	NB	SB	Total
100%[b]	61-1112-55[c]	76-787-249	158-176-365	548-115-102	3804
v/s at 100%	0.31	0.22	0.20	0.15	/
70%	43-778-39	53-551-174	111-123-256	384-81-71	2663
120%	73-1334-66	91-944-299	190-211-438	658-138-122	4565

[a] The EB represents eastbound.

[b] The 100% represents that the demand level is the real-world afternoon peak data. Other demand levels are percentages of the real-world data demand.

[c] The series of numbers represents the traffic demand (vph) in the left-turning, through, and right-turning directions, respectively.

The simulation model is calibrated by the real-world traffic data. The original data was collected during PM peak (4:00pm - 5:00pm) and recorded by cameras from which the hourly volume and turn ratios of each approach are manually extracted and set in the SUMO simulation, along with the speed limit of the road. The traffic volume information is detailed in Table 1. The first column lists the demand levels, as percentages of the real-world peak hour demand. The second to fifth columns list the traffic demand (vph) for each movement (left-through-right). The sixth column lists the total demand at the intersection. Each row lists the demand in each direction of each approach. The 100% demand level represents the real-world afternoon peak data and other demand levels are percentages of the real-world demand. The second row lists the volume-to-saturation flow (v/s) ratios of the critical lane groups where the saturation flow rate is assumed to be 1800 veh/h/lane. To validate the calibration results, we calculate the GEH value of each movement to make sure that all values are smaller than 5 (Chu et al., 2003).

3.1.2. Measurement of performance

We use the average vehicle delay, denoted as $\overline{TTL}$, as the measurement of performance in this work, which is calculated following Eq. (3.1):

$$\overline{TTL} = \frac{1}{M^{total}} \sum_{m \in M^{total}} x_{m,t_m}, \tag{3.1}$$

where

- M^{total} is a set of all vehicles that finish their trips in this simulation run (excluding vehicles that are generated during the warm-up stage);
- x_{m,t_m} is the delay of vehicle m which finishes its trip at time t_m. The calculation follows Eq. (2.4).

The final measurement of performance is the mean and standard deviation of $\overline{TTL}$ of all testing rounds. We take the average of 3 rounds of testing by default for all experiments with different random seeds.

3.1.3. Benchmark algorithms

Several TSC methods are used as benchmarks in this work:

- **Fixed-time TSC.** A fine-tuned fixed-time TSC based on Webster's method (Webster, 1958) under 100% demand level is adopted. The green phase length is 26 s, 17 s, and 12 s for Phases 1, 2, and 3, respectively. The transition time is set to 5 s per phase, which is the same as CAVLight and applies to all benchmarks.
- **Actuated TSC.** An actuated TSC is adopted and calibrated based on multiple rounds of simulation results. Specifically, we select combinations of critical actuated signal parameters such as minimum green time, maximum green time, and extension time for each phase and then run simulations in SUMO to evaluate the performance in terms of average vehicle delay. Finally, we select the best performance parameters as the actuated TSC parameters. The minimum green time for each phase in order is 15 s, 10 s, and 5s, respectively. The maximum green time for each phase is set as 35 s, 25 s, and 18 s, respectively. The max gap, detector gap, and passing time are set as 3 s, 2 s, and 2 s, respectively.
- **CAVLight (no CTM).** For CAVLight (no CTM), the critic input is the ground truth information and the actor input is directly from CAV observations without CTM estimation. The state, action, and reward design are the same as the CAVLight (CTM).
- **CAVLight (CTM).** The demand values of CTM source cells in CAVLight (CTM) are obtained by loop detectors placed at the entrance of all incoming lanes. The loop detectors report the number of passing vehicles during the past 1 s, which serves as the real-time demand for the CTM source nodes.
- **I-SIG (CTM) and I-SIG (no CTM).** The I-SIG algorithm is initially proposed by Feng et al. (2015). The algorithm is derived from enhancements made to the Controlled Optimization of Phases (COP) algorithm (Sen and Head, 1997). The I-SIG algorithm utilizes dynamic programming (DP) to optimize phase duration and sequences with the objective of minimizing total delay. Although a phase sequence is provided, the I-SIG algorithm can choose to skip the next phase based on the total delay calculation and disregard the dual-ring structure, and therefore is acyclic. To solve the DP problem, an arrival table is required and serves as the input. The arrival table comprises predictions of future arrival flows for each phase at each timestamp. Similarly to CAVLight, we integrate I-SIG with CTM, denoted as I-SIG (CTM), where the arrival table is constructed by the CTM-based traffic state estimation model. For I-SIG (no CTM), the arrival table is directly constructed using CAV observations.
- **Max-pressure TSC (CTM).** Max-pressure (Varaiya, 2013) algorithm is a rule-based adaptive TSC method that can be analytically proven to stabilize queue length within a demand range. The max-pressure method greedily selects the phase with the highest pressure where the pressure is defined as the difference between the number of vehicles on incoming lanes of the phase and the number of vehicles on outgoing lanes. In this work, since the study area is an isolated intersection, the pressure calculation only considers the number of vehicles on incoming lanes. Meanwhile in this benchmark, loop detector data is also integrated with CTM to provide a more accurate state estimation of the number of vehicles. The max-pressure method is executed by the minimum green time (10 s) and then every 1 s the algorithm decides to continue or switch to the next phase with the highest pressure.

3.1.4. Parameters and experiment design

We summarize the parameters used in our proposed method as follows. The study area is assumed to be 200 m from the intersection center (only vehicles within the study area will be considered). The parameters of the CTM in this paper are listed as follows. Free flow speed v_f is set to 64.37 km/h (17.88 m/s), which is the same as the speed limit of this intersection. The time step is set to 1 s, and therefore the cell length is 17.88 m. Maximum flow rate q_{max} is set to 1800 vph. The corresponding critical density k_{cri} is 27.96 vpkm and jam density k_{jam} is 133.33 vpkm, which is calculated based on the SUMO vehicle length and the minimum gap between vehicles. The backward shockwave speed w is 17.1 km/h (4.75 m/s). For the CTM-based state estimation process, the number of upstream or downstream cells that can be detected by a CAV is set as 4, considering the detection range as 80 m and cell length as 17.88 m. Specifically, the intersection size is taken into consideration when determining detected cells. If the CAV is near the stop bar, we assume the intersection width equals 2 cells for the through direction, 2 cells for the left-turn direction, and 1 cell for the right-turn direction. For example, if one CAV is at the closest cell to the intersection, the number of downstream cells it can observe in the through direction will be 2 rather than 4.

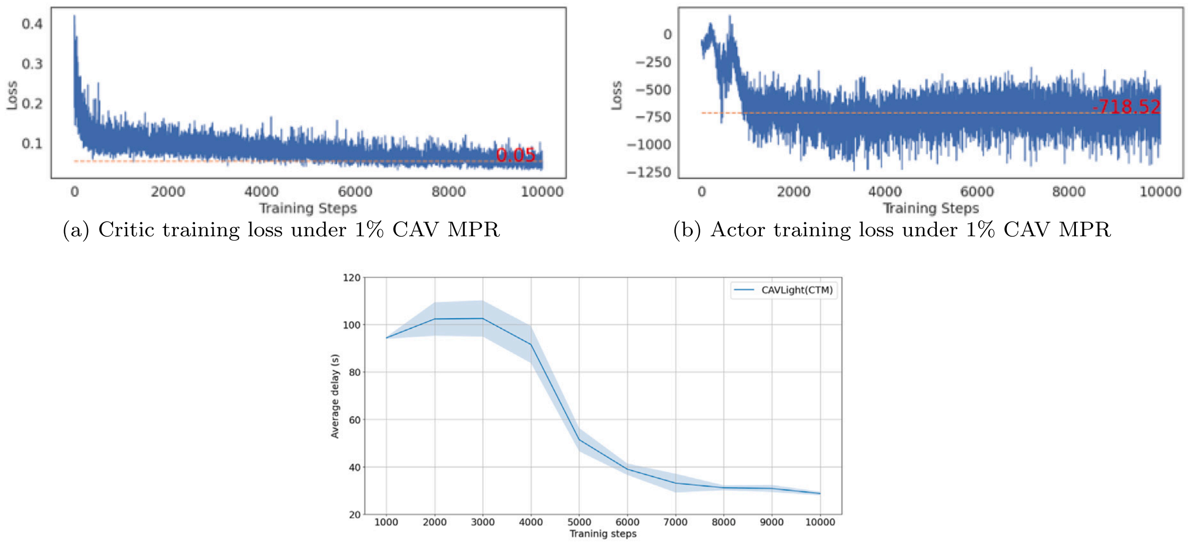
As to the neural networks, both the critic DNN and actor DNN consist of two hidden fully connected layers. For the critic DNN, each hidden layer consists of $3|s|$ neurons ($|s|$ is the length of the state vector s) and the exponential linear unit (ELU) activation functions. For the actor DNN, each hidden layer consists of $3|o^{CTM}|$ neurons and ELU activation functions. The output layer activation function is a softmax function with a 100 temperature value for the actor DNN and a linear function for the critic DNN. The Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1e−4 is adopted for the actor and with a learning rate of 1e−3 for the critic. We adopt He uniform (He et al., 2015) as the layer weight initializer for the hidden and output layers. The experience replay buffer size, batch size, discount rate, and iteration number used in this work are 10,000, 128, 0.99, and 10,000, respectively. The computation time for CAVLight to generate one action is around 0.004 s under all testing cases on an Ubuntu 20.04 desktop with an Intel i7-12700KF CPU.

The experiment design is summarized in Table 2, including algorithms, demand levels, and CAV MPRs. The first column lists the experiment names. The second column gives the algorithms used in the corresponding experiment. The third and fourth columns summarize the traffic demand and CAV MPR used in RL agent training. The fifth and sixth columns show the traffic demand and

Table 2
Numerical experiments design.

Experiments	Algorithms	Training		Testing	
		Demand (%)	CAV MPR (%)	Demand (%)	CAV MPR (%)
Generalizability in traffic demands and CAV MPR	CAVLight (CTM), CAVLight (no CTM); I-SIG (CTM), I-SIG (no CTM), Max-pressure (CTM) Fixed-time, Actuated	70	1	70, 100, 120	1,3,5,8,10,30,50
Sensitivity analysis on CTM parameters	CAVLight (CTM), CAVLight (CTM vf), CAVLight (CTM qmax)[a]	70	1	100	1,3,5,8,10,30,50

[a] The CTM vf and CTM qmax are two variants of CTM using different parameters. Details can be found in Section 3.4.



(a) Critic training loss under 1% CAV MPR

(b) Actor training loss under 1% CAV MPR

(c) Performance during the training process under 1% CAV MPR

Fig. 10. Training process of the CAVLight (CTM) agent under 1%.

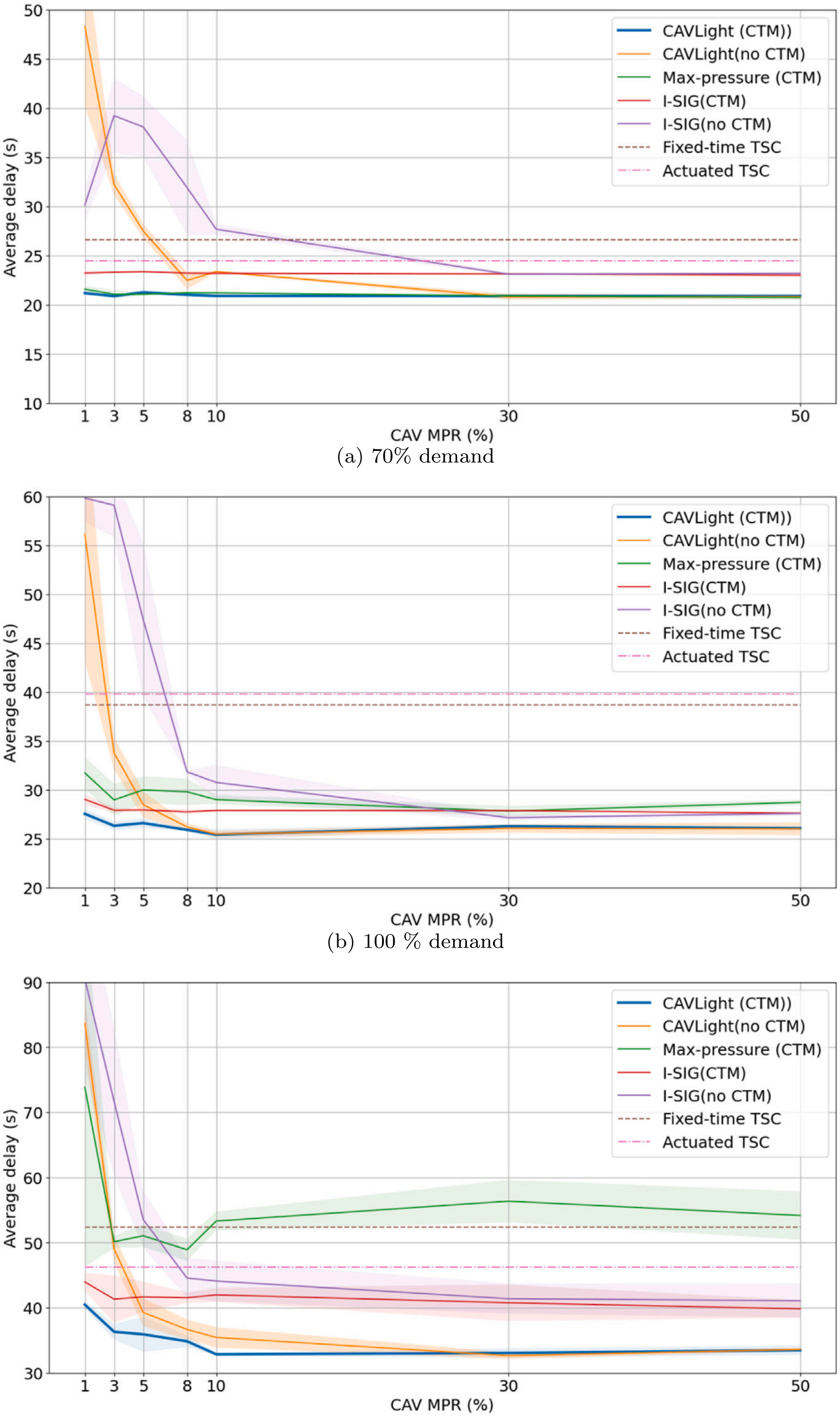
CAV MPR used in testing the performance of RL agents as well as other benchmarks. We train the CAVLight model only at the 1% CAV MPR and 70% demand level (with CTM and loop detector data) and test its performance under various scenarios to show its generalizability in CAV MPRs and demand levels. The reason we select 1% as the training CAV MPR is twofold: first, we can show the capability of the proposed method in learning patterns from significantly different states of critic and actor; second, learning under the 1% CAV MPR can help the RL agent to better prepare for low CAV MPR scenarios and the learned policy can be easily extended to higher CAV MPR scenarios. We take 70% as the training demand for two reasons: first, the 100% demand level is the peak hour demand, which does not represent normal traffic conditions. Therefore we choose a more common demand level. Second, a smaller demand in training can help RL-TSC agent to better converge as it can avoid possible congestion at the exploration stage and improve the quality of collected experience. In the following subsections, we will come back to this table and explain experiment settings in detail.

The training process of CAVLight (CTM) under 1% CAV MPR is illustrated in Fig. 10. For the upper two sub-figures, the x-axis is the training iteration number, and the y-axis is the training loss. Figures in the upper left column are critic training loss and those in the upper right column are actor training loss. The dashed orange line and numbers give the average loss values in the last 1000 iterations. From Fig. 10, both the actor and critic converged. Fig. 10(c) shows the average delay per vehicle of the CAVLight (CTM) model after every 1000 steps of training under the 100% demand scenario. We can tell that the performance of the proposed model converged after around 8000 steps.

3.2. Performance comparison

In this subsection, we compare the performance of CAVLight and other benchmarks under different demand levels and CAV MPRs, as shown in the first row of Table 2.

We test the performance of the CAVLight under different traffic demand levels by comparing its performance with other benchmarks, as shown in Fig. 11 and Table 3. Fig. 11 illustrates the performance of the CAVLight (CTM) and all other benchmarks



(a) 70% demand

(b) 100 % demand

(c) 120% demand

Fig. 11. Performance comparison under three demand levels.

under three demand levels with various CAV MPRs. Each sub-figure of Fig. 11 shows the comparison under a certain demand level, as detailed in Table 1. The y-axis is the average vehicle delay, and the x-axis is the CAV MPR. The blue, orange, green, red, and purple lines represent the performance of CAVLight (CTM), CAVLight (no CTM), max-pressure (CTM), I-SIG (CTM), and I-SIG (no CTM),

Table 3
Average delay of TSC algorithms under different demand levels and CAV MPRs.

TSC algorithms	Demand level	CAV MPR (%)						
		1	3	5	8	10	30	50
CAVLight (CTM)	70%[a]	**21.2**[b]	**20.9**	21.3	**21.0**	**20.9**	20.9	20.9
	100%	**27.5***[c]	**26.3***	**26.6***	**25.9**	**25.4**	26.3	26.1
	120%	**40.5***	**36.3***	**35.9***	**34.9***	**32.9***	33.1	**33.5**
CAVLight (no CTM)	70%	48.3	32.2	27.5	22.5	23.4	**20.8**	20.8
	100%	56.1	33.8	28.5	26.2	25.5	**26.1**	**26.0**
	120%	83.7	49.1	39.1	36.7	35.4	**32.6**	33.6
I-SIG (CTM)	70%	23.3	23.2	23.3	23.4	23.3	23.4	22.9
	100%	28.5	28.4	28.2	27.7	27.5	26.9	27.2
	120%	47.6	39.6	38.8	40.8	41.5	39.9	40.0
I-SIG (no CTM)	70%	27.4	31.4	28.0	29.9	26.2	23.0	23.0
	100%	43.1	43.4	40.6	31.4	30.4	27.4	27.7
	120%	87.4	58.3	52.6	46.6	41.8	39.6	40.9
Maxpressure (CTM)	70%	21.6	21.1	**21.1**	21.2	21.2	20.9	**20.7**
	100%	31.7	29.0	30.0	29.8	29.0	27.8	28.7
	120%	73.9	50.1	51.1	48.9	53.3	56.4	54.2
Actuated	70%	24.5						
	100%	39.8						
	120%	46.2						
Fixed-time	70%	25.6						
	100%	38.7						
	120%	52.3						

[a] 70% represents the demand level, as detailed in Table 1.

[b] The bold number represents the best-performance algorithm in terms of average delay under the corresponding demand level and CAV MPR.

[c] The * sign shows that the result significantly outperforms the second-best performed algorithm in a t-test (p = 0.05).

respectively. The shadow area around each line represents the corresponding standard deviation value. The brown and magenta dash lines represent the performance of fixed-time TSC and actuated TSC, respectively. In Table 3, the result of the best-performed algorithm under each pair of CAV MPR and demand is marked as bold.

We interpret the result from the following four perspectives:

Comparison between methods with CTM estimation and those without CTM estimation. Under scenarios with low CAV MPRs (below 10%), comparing algorithms with CTM and those without CTM, we can see the effectiveness of integrating the traffic flow model. For example, CAVLight (CTM) can significantly outperform the CAVLight (no CTM) under low CAV MPRs. Such a pattern can be observed from I-SIG (CTM) and I-SIG (no CTM) as well.

Comparison among CAVLight (CTM), I-SIG (CTM), and Max-pressure (CTM). CAVLight (CTM) and Max-pressure (CTM) achieve a similar level of performance under the 70% demand level and significantly outperform Max-pressure (CTM) under 100% and 120% demand scenarios. The good performance of CAVLight (CTM) over Max-pressure can be explained by the fact that the greedy design of Max-pressure cannot generate a better policy than the CAVLight (CTM). CAVLight (CTM) also significantly outperforms I-SIG (CTM) under 70%, 100%, and 120% demand levels. This can be explained by the fact that the arrival table used in I-SIG is constructed with an assumption of constant speed of arrival vehicles, which can deviate from the truth considering the queuing behavior.

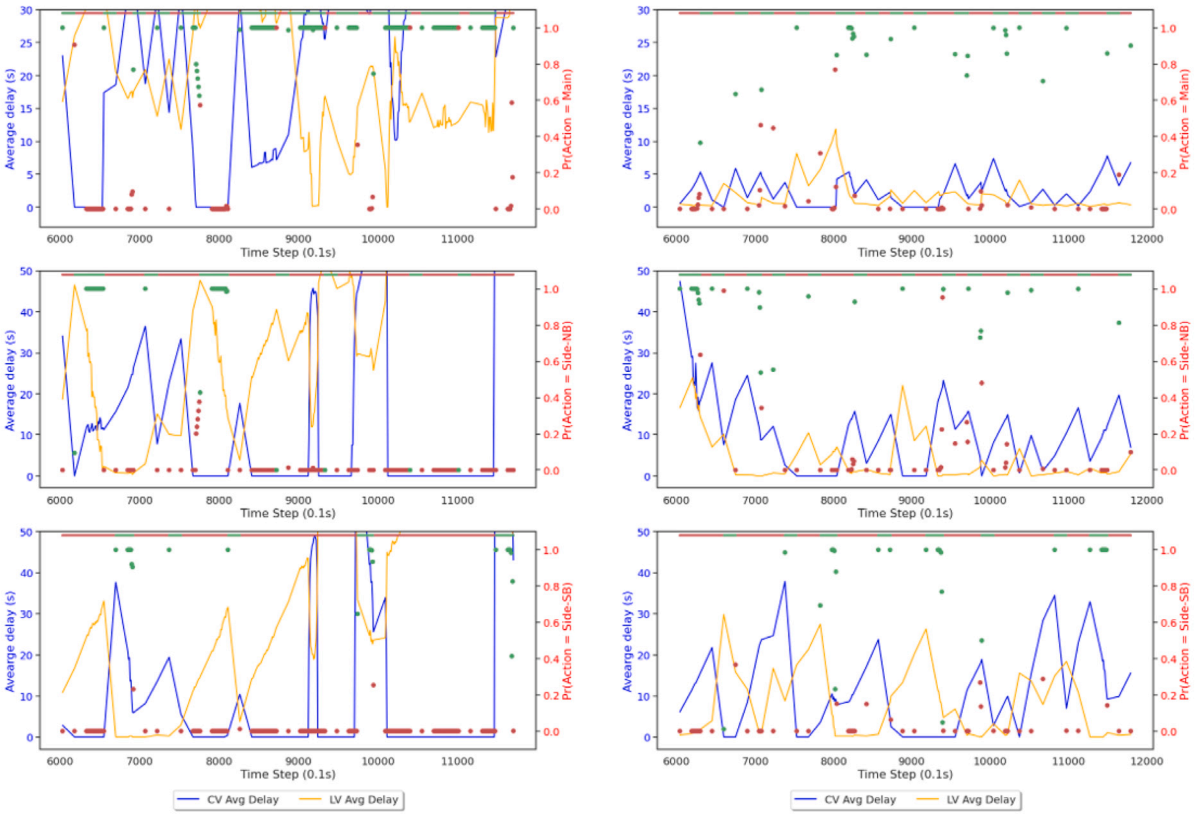
Generalizability of CAVLight (CTM) in traffic demands. By comparing the performance of methods under different demand levels, the result demonstrates a good generalizability of CAVLight (CTM) under varying traffic demands. Under all tested demand levels, CAVLight outperforms all benchmarks and achieves a similar level of performance under 1% CAV MPR as that under 50% CAV MPR. Algorithms such as fixed-time TSC and actuated TSC can perform well under the 70% demand but perform worse as the demand level increases.

Generalizability of CAVLight (CTM) in CAV MPR. Even though the CAVLight model is trained under 1% CAV MPR, learned policy can be extended towards higher CAV MPR cases and outperform all other benchmarks. This can be explained by the following two reasons. First, the CTM estimation can effectively provide information with the RL actor even under extremely low CAV MPR cases. Second, the proposed Asym-A2C architecture can bridge the ground truth information and the observed partial information and therefore guide the policy learning under low CAV MPR scenarios.

3.3. *Policy visualization*

In this subsection, we visualize and analyze the learned policy of CAVLight (CTM) agents under 120% demand. The difference in policy between CAVLight (CTM) and CAVLight (no CTM) under the 1% CAV MPR is shown in Fig. 12(a) and (b).

In each sub-figure, three panels represent three actions, i.e., signal phases. In each panel, the left y-axis is the average delay per vehicle at a certain time step, the right y-axis is the probability of choosing the corresponding phase, and the x-axis is the



(a) CAVLight (no CTM) under 1% CAV MPR

(b) CAVLight (CTM) under 1% CAV MPR

Fig. 12. Comparison between the learned policy of CAVLight variants under 1% CAV MPR and 120% demand.

simulation time step with 0.1s resolution. The blue line and orange line represent the average delay of CVs (including CAVs and augmented CVs) and LVs (legacy vehicles or unobserved vehicles), respectively. The green and red dots represent the corresponding phase status. In other words, if the phase is selected at that time step, the dot will be green otherwise red. The position of the dots represents the probability of choosing that action (or phase) in the generated policy. Groups of green dots mean the agent chooses to continue the current phase multiple times. The green and red bars on the top of each sub-figure represent the phase status. We compare the learned policies in a 10-min window (from 600.0 to 1200.0 s).

From Fig. 12(a) and (b), the CAVLight (CTM) shows a significantly different policy pattern compared with CAVLight (no CTM). CAVLight (CTM) will select the phase even if no CVs are detected (i.e. the red line is flat) while the CALight (no CTM) agent ignores the southbound vehicles and only assigns a green phase whenever CVs are detected. This behavior difference explains the performance difference as shown in Fig. 11: the CTM estimation method can provide necessary traffic status information with CAVLight (CTM) agent even if there are no CAVs and serve vehicles on the side street.

3.4. Sensitivity analysis on CTM parameters

In this subsection, we analyze the sensitivity of CAVLight on CTM parameters by comparing the testing performance of well-trained CAVLight agents under three sets of CTM parameters, as shown in the third row of Table 2. Based on the previous CTM parameters, we modify the free flow speed from 17.88 m/s to 10.9 m/s (denoted as *CTM vf*) and the critical density from 27.96 vpkm to 46.0 vpkm. We also modify the maximum flow rate from 1800 vph to 1500 vph (denoted as *CTM qmax*) and the critical density to 23.30 vpkm. Corresponding fundamental diagrams are shown in Fig. 13. The motivation for conducting CTM parameter sensitivity analysis is to further validate the generalizability of the proposed CAVLight model. It is interesting to see the performance of the controller when the assumed traffic flow model is inaccurate, which is common in real-world implementations. The CAVLight agents are trained under CTM and tested with CTM, CTM vf, and CTM qmax, respectively, to fairly compare the influence of CTM parameters.

We compare the CAVLight performance under different CTM parameters in Fig. 14(a) and CTM estimation accuracy in Fig. 14(b). For Fig. 14(a), the y-axis represents the average vehicle delay and the x-axis represents the CAV MPR. The blue, orange, and red lines represent the performance of CAVLight (CTM), CAVLight (CTM qmax), and CAVLight (CTM vf), respectively. The shadow

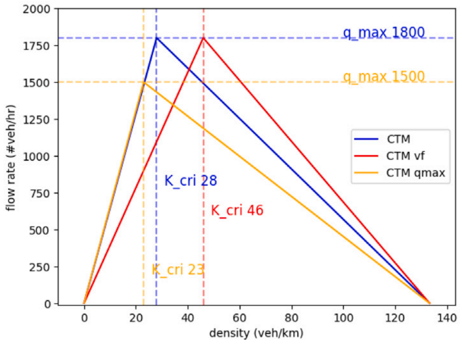
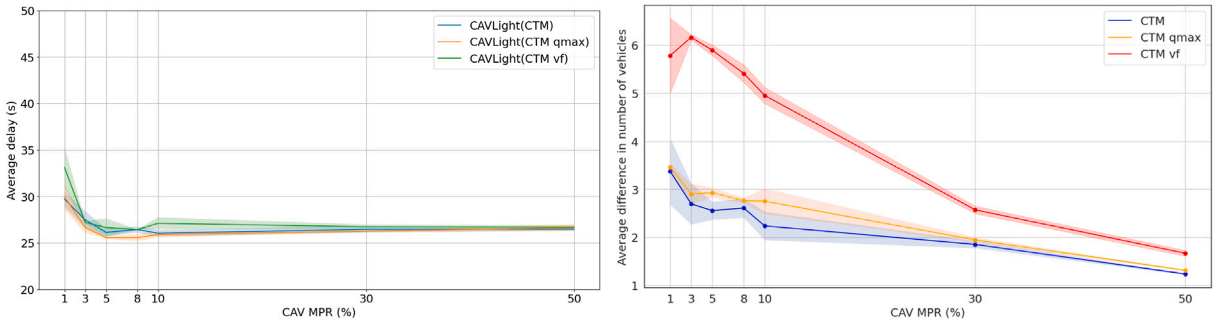


Fig. 13. Fitted fundamental diagram for CTM v2.



(a) Performance comparison among CAVLight (CTM), CAVLight (CTM vf), and CAVLight (CTM qmax)

(b) CTM estimation accuracy comparison

Fig. 14. Performance comparison between three CTM versions.

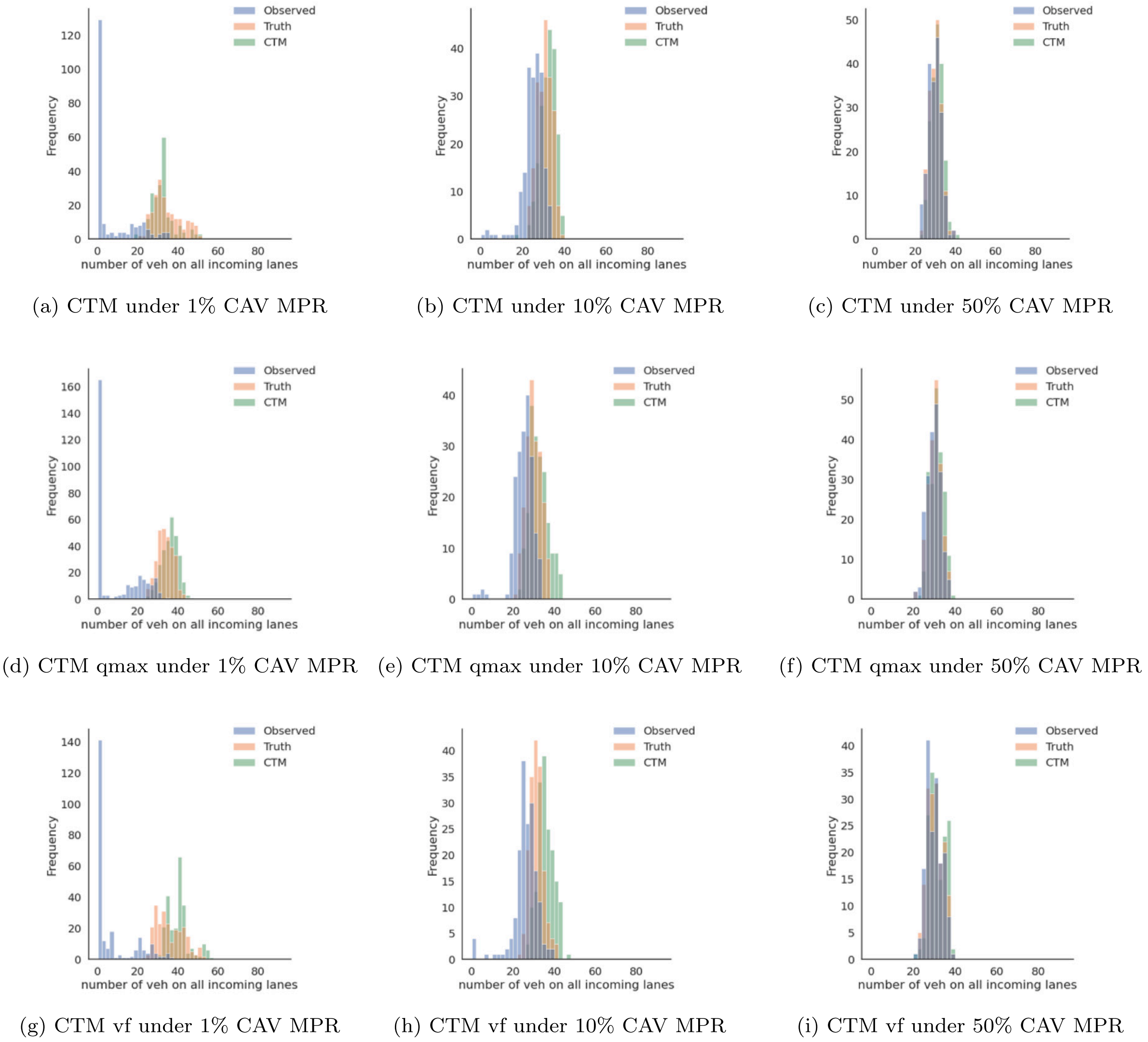
area represents the standard deviation. For Fig. 14(b), the y-axis is the average absolute difference in the number of vehicles on incoming lanes between CTM estimation and the ground truth, and the x-axis is the CAV MPR. The blue, orange, and red lines represent the performance of CTM, CTM qmax, and CTM vf, respectively. The shadow area represents the standard deviation. By comparing Fig. 14(a) and (b), we can conclude that CTM accuracy is insensitive to qmax, i.e. the maximum flow rate, to some extent. Consequently, the CAVLight (CTM) and CAVLight (CTM qmax) agents achieve the same level of performance under each CAV MPR scenario. The free flow speed, v_f, can significantly influence the CTM accuracy and therefore affect the performance of CAVLight (CTM vf). When it deviates from the truth, it can downgrade the performance of CAVLight (CTM) under 1% CAV MPR. However, such influence is still limited when CAV MPR is above 3%.

We further investigate the CTM estimation accuracy by visualizing the frequency distribution of CTM estimated number of vehicles on incoming lanes, as shown in Fig. 15.

In Fig. 15, we compare the frequency distribution of the estimated number of vehicles by CTM, CTM qmax, and CTM vf under 1%, 10%, and 50% CAV MPRs. The first row lists figures from CTM, the second row lists figures from CTM qmax, and the third row lists figures from CTM vf. For each sub-figure, the y-axis is the frequency, and the x-axis is the number of vehicles on all incoming lanes. The blue bars represent the distribution of the observed number of vehicles by CAVs, the orange bars represent the distribution of the ground truth, and the green bars represent the distribution of the CTM estimation.

By comparing the distribution of CTM estimation, observed vehicles, and the ground truth, for example in Fig. 15(a), CTM estimation can greatly improve the accuracy of the traffic state information than solely relying on CAV cooperative perception under low CAV MPRs. By comparing the CTM estimation and the ground truth across different CTM parameters, the accurate CTM can generate a better estimation than others and both CTM vf and CTM qmax overestimate the number of vehicles on incoming lanes. The lower maximum flow rate and the lower free-flow speed result in a lower queue discharging rate, leading to the overestimation of remaining vehicles on incoming lanes.

In summary, the result of the sensitivity analysis on CTM parameters indicates that one well-trained CAVLight under a certain CTM can be easily transferred to other scenarios with different CTM parameters. Still, for a CAV MPR as low as 1%, a more accurate CTM can help improve the CAVLight performance. The robustness of CAVLight (CTM) against inaccurate CTM parameters can be explained by the fact that (1) the RL state solely relies on macroscopic traffic information and is therefore tolerant to some errors in the CTM estimation; and (2) the Asym-A2C design allows the agent to be trained using the ground truth data, which can help the agent learn to estimate the true traffic state from the estimated CTM state.



(a) CTM under 1% CAV MPR (b) CTM under 10% CAV MPR (c) CTM under 50% CAV MPR

(d) CTM qmax under 1% CAV MPR (e) CTM qmax under 10% CAV MPR (f) CTM qmax under 50% CAV MPR

(g) CTM vf under 1% CAV MPR (h) CTM vf under 10% CAV MPR (i) CTM vf under 50% CAV MPR

Fig. 15. Frequency distributions of CTM estimated number of vehicles on incoming lanes under various CAV MPRs.

4. Discussion: realistic detection model and improved CTM integration

Results presented in the previous section assume ideal CAV detection. To better reflect realistic CAV detection capabilities, in this section, we adopt the average precision result of the RSN CarXL 3f algorithm from Waymo (Table 1 in Sun et al. (2021)). We assume the CAV detection average precision (AP) is different with different distances towards the surrounding vehicles. Specifically, within 30 m, the AP is 92%; between 30 to 50 m, the AP is 77%; and for vehicles within the 50 to 80 m range, we set the AP as 57%. To realize this, we sample surrounding vehicles based on the corresponding AP at each time step (every 0.1 s) and only report the detected vehicle information for downstream data processing.

The CTM integration is also updated to accommodate the detection precision loss. First, for each cell that is detected, we set the highest AP from all CAVs that can detect that cell as the confidence value of the cell. Then, instead of directly replacing the CTM estimation with the observation, we linearly combine the observed number of vehicles in the cell with a scaled CTM estimation, where the scale is one minus the confidence value. For example, if one cell is detected by two CAVs where one CAV is 60 m away from the cell and the other is 25 m away, the confidence value of the cell will be set as 0.92. Then, the total estimated number of vehicles in that cell will be the observed number of vehicles plus 0.08 multiplied by the CTM estimation.

Second, we consider the correlation among cells with observed stop cells (i.e., cells with jam density) to improve CTM accuracy. For each cell in CTM, we will first decide if the number of detected stopped vehicles (vehicle speed below 0.3 m/s) surpasses a threshold determined by cell capacity and the detection confidence value. If so, we denote such a cell as a stop cell and set the number of vehicles in the cell according to its jam density. We further estimate other cells' status using the stop cell positions and

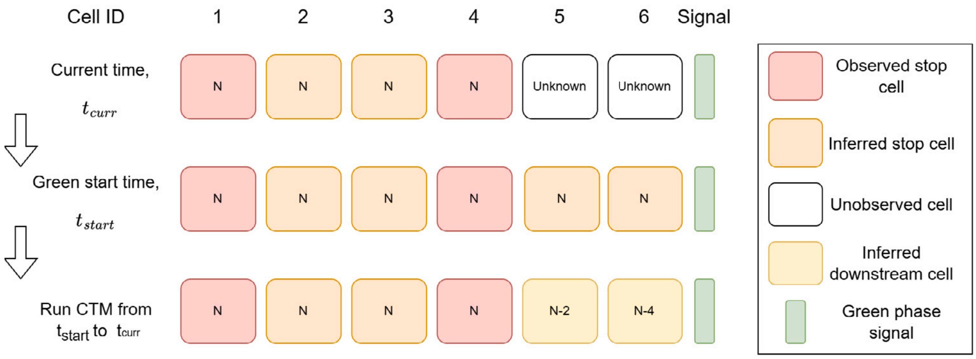
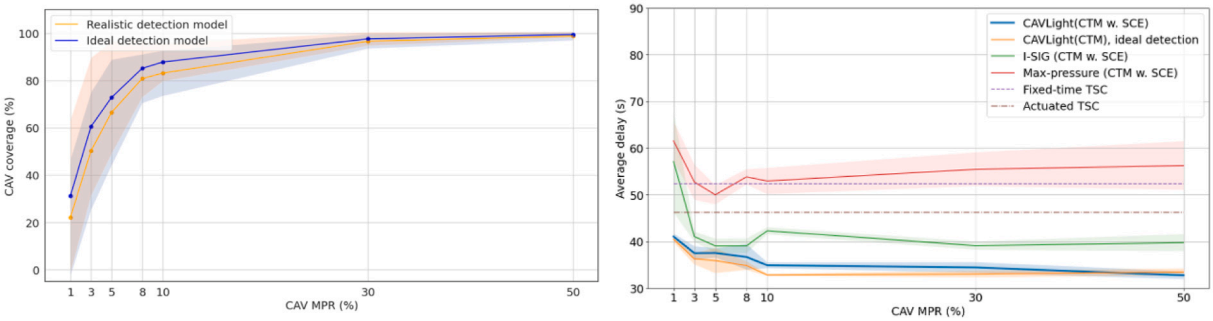


Fig. 16. Illustration of stop cell estimation in CTM.



(a) CAV coverage rate comparison (b) Performance comparison

Fig. 17. CAV coverage rate and model performance comparison under two detection models.

traffic signal status. If the corresponding signal status for an approach with stop cells is red, we set all cells between the farthest stop cells and the stop bar as stop cells. If the signal status is green, we set the cells in between the farthest stop cell and the closest stop cell towards the stop bar as stop cells. The number of vehicles in each downstream cell of the closest stop cell is determined in the following way. First, they are all set as stop cells at the start time of the current green phase. Then, the CTM is executed from the phase starting time until the current observation time to calculate the actual number of vehicles. An example is provided in Fig. 16. The numbers in the cells represent the number of vehicles where N is the cell capacity. Cell 1 and cell 4 are observed and determined as stop cells (denoted by red cells). We first infer cells in between to be stop cells (denoted by orange cells). For the unobserved downstream cells of the closest stop cell, i.e., cell 5 and cell 6, we set them as stop cells at the start time of the current green phase; then, we run CTM from the start time to the current observation time to estimate the current cell status (denoted by light yellow cells).

Fig. 17 shows the difference in CAV coverage rate (Fig. 17(a)) and model performance (Fig. 17(b)) under the ideal detection model and the realistic detection model at 120% demand. From Fig. 17(a), the realistic detection model has a lower coverage, around 10% lower on average when CAV MPR is below 3%. The coverage difference in the two detection models gets smaller as the CAV MPR increases. This can be explained by the fact that a higher CAV MPR allows more CAVs to share the perception results and eliminate the negative effects of imperfect perception from a single vehicle. Using the realistic detection model, the performance of CAVLight (CTM w. SCE), I-Sig (CTM w. SCE), and Max pressure (CTM w. SCE) are compared. We also include the performance of CAVLight (CTM) under the ideal detection model, fixed-time TSC, and actuated TSC as references, as shown in Fig. 17(b). It can be seen that CAVLight (CTM w. SCE) outperforms both the I-Sig and Max-pressure under the realistic detection model, which is consistent with the findings in the previous section. Moreover, it achieves a similar level of performance as the CAVLight (CTM) under the ideal detection model. The results demonstrate the robustness of the proposed DRL and CTM integration framework under the realistic CAV detection environment.

5. Conclusions and future research

This paper introduced a novel RL-TSC method designed specifically for scenarios with extremely low CAV MPRs. The proposed approach integrated the CTM into the traffic state estimation process to address the challenges posed by highly skewed and imbalanced CAV cooperative perception observations during the early stages of CAV deployment. Through extensive numerical experiments conducted at a real-world intersection, with varying traffic demand levels and CAV MPRs, the superiority of the proposed CAVLight with CTM method was demonstrated and compared to a few benchmark algorithms. Furthermore, the generalization of CAVLight (CTM) was examined, revealing that once the model was well-trained under one CAV MPR and demand level, it could be effectively extended to other scenarios without re-training. Visualizing the learned policies of CAVLight agents further shed light on the benefit of CTM state estimation on stabilizing the behaviors of CAVLight agents under 1% CAV MPR scenario. A sensitivity analysis on CTM parameters indicated that CAVLight was insensitive to CTM parameters, which demonstrated its generalizability over different traffic scenarios. Still, a more accurate CTM can enhance the performance of CAVLight agents with low CAV MPRs. We also investigate the proposed method under a realistic CAV detection scenario and compare its performance with benchmarks in the discussion section. The result demonstrates the robustness of CAVLight and CTM integration framework against an imperfect detection environment.

Future work can be directed towards several potential avenues: (1) Expanding the investigation to encompass corridors and road networks, while exploring the incorporation of model-based TSC in the training stage of RL-TSC to encourage coordination between RL agents. (2) Enhancing the realism of the CAV cooperative perception model used in the state estimation process by better simulating realistic detection errors and occlusions. (3) Addressing the challenge of estimating reliable microscopic traffic information, as current research on joint control of CAV trajectories and TSC often necessitates such data, prompting further exploration of alternative microscopic state estimation methods under extremely low CAV MPR scenarios.

CRediT authorship contribution statement

Wangzhi Li: Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis. **Tianheng Zhu:** Writing – review & editing, Methodology, Formal analysis. **Yiheng Feng:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Data availability

Data will be made available on request.

Acknowledgments

This research is supported in part by the U.S. National Science Foundation through grant CPS #2038215. The views presented in this paper are those of the authors alone.

References

Al Islam, S.B., Hajbabaie, A., Aziz, H.A., 2020. A real-time network-level traffic signal control methodology with partial connected vehicle information. Transp. Res. C 121, 102830.

Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D., 2011. SUMO–simulation of urban mobility: an overview. In: Proceedings of SIMUL 2011, the Third International Conference on Advances in System Simulation. ThinkMind.

Caillot, A., Ouerghi, S., Vasseur, P., Boutteau, R., Dupuis, Y., 2022. Survey on cooperative perception in an automotive context. IEEE Trans. Intell. Transp. Syst. 23 (9), 14204–14223.

Cao, P., Xiong, Z., Liu, X., 2022. An analytical model for quantifying the efficiency of traffic-data collection using instrumented vehicles. Transp. Res. C 136, 103558.

Chen, H., Liu, B., Zhang, X., Qian, F., Mao, Z.M., Feng, Y., 2022. A cooperative perception environment for traffic operations and control. arXiv preprint arXiv:2208.02792.

Chu, L., Liu, H.X., Oh, J.-S., Recker, W., 2003. A calibration procedure for microscopic traffic simulation. In: Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems. Vol. 2, IEEE, pp. 1574–1579.

Daganzo, C.F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. B 28 (4), 269–287.

Daganzo, C.F., 1995. The cell transmission model, part II: network traffic. Transp. Res. B 29 (2), 79–93.

Feng, Y., Head, K.L., Khoshmagham, S., Zamanipour, M., 2015. A real-time adaptive signal control in a connected vehicle environment. Transp. Res. C 55, 460–473.

Feng, Y., Zheng, J., Liu, H.X., 2018. Real-time detector-free adaptive signal control with low penetration of connected vehicles. Transp. Res. Rec. 2672 (18), 35–44.

Goodall, N.J., Smith, B.L., Park, B., 2013. Traffic signal control with connected vehicles. Transp. Res. Rec. 2381 (1), 65–72.

Guler, S.I., Menendez, M., Meier, L., 2014. Using connected vehicle technology to improve the efficiency of intersections. Transp. Res. C 46, 121–131.

Guo, Q., Li, L., Ban, X.J., 2019. Urban traffic signal control with connected and automated vehicles: A survey. Transp. Res. C 101, 313–334.

Guo, Y., Ma, J., 2021. DRL-TP3: A learning and control framework for signalized intersections with mixed connected automated traffic. Transp. Res. C 132, 103416.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1026–1034.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Korecki, M., Helbing, D., 2022. Analytically guided reinforcement learning for green it and fluent traffic. IEEE Access 10, 96348–96358. http://dx.doi.org/10.1109/ACCESS.2022.3204057.

Li, W., Ban, X., 2020. Connected vehicle-based traffic signal coordination. Engineering 6 (12), 1463–1472.

Li, T., Han, X., Ma, J., 2021. Cooperative perception for estimating and predicting microscopic traffic states to manage connected and automated traffic. IEEE Trans. Intell. Transp. Syst. 23 (8), 13694–13707.

Li, L., Jabari, S.E., 2019. Position weighted backpressure intersection control for urban networks. Transp. Res. B 128, 435–461.

Liang, E., Su, Z., Fang, C., Zhong, R., 2022. OAM: An option-action reinforcement learning framework for universal multi-intersection control. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36, pp. 4550–4558.

Liu, H., Liang, W., Rai, L., Teng, K., Wang, S., 2019. A real-time queue length estimation method based on probe vehicles in CV environment. IEEE Access 7, 20825–20839. http://dx.doi.org/10.1109/ACCESS.2019.2898424.

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using SUMO. In: The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE, URL https://elib.dlr.de/124092/.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning. PMLR, pp. 1928–1937.

Mo, Z., Li, W., Fu, Y., Ruan, K., Di, X., 2022. Cvlight: Decentralized learning for adaptive traffic signal control with connected vehicles. Transp. Res. C 141, 103728.

SAE International, 2020. Taxonomy and definitions for terms related to cooperative driving automation for on-road motor vehicles.

Sen, S., Head, K.L., 1997. Controlled optimization of phases at an intersection. Transp. Sci. 31 (1), 5–17.

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. Transp. Syst. 23 (11), 20021–20035.

Shou, Z., Di, X., 2020. Reward design for driver repositioning using multi-agent reinforcement learning. Transp. Res. C 119 (102738).

Song, L., Fan, W., 2021. Traffic signal control under mixed traffic with connected and automated vehicles: a transfer-based deep reinforcement learning approach. IEEE Access 9, 145228–145237.

Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D., 2021. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5725–5734.

Sutton, R.S., Barto, A.G., et al., 1998. Introduction to Reinforcement Learning. Vol. 135, MIT press Cambridge.

Varaiya, P., 2013. Max pressure control of a network of signalized intersections. Transp. Res. C 36, 177–195.

Wang, X., Jerome, Z., Wang, Z., Zhang, C., Shen, S., Kumar, V.V., Bai, F., Krajewski, P., Deneau, D., Jawad, A., et al., 2024. Traffic light optimization with low penetration rate vehicle trajectory data. Nature Commun. 15 (1), 1306.

Wang, X., Yin, Y., Feng, Y., Liu, H.X., 2022. Learning the max pressure control for urban traffic networks considering the phase switching loss. Transp. Res. C 140, 103670.

Wang, Q., Yuan, Y., Yang, X.T., Huang, Z., 2021. Adaptive and multi-path progression signal control under connected vehicle environment. Transp. Res. C 124, 102965.

Webster, F.V., 1958. Traffic Signal Settings. Technical Report.

Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., Li, Z., 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1290–1298.

Wu, T., Zhou, P., Liu, K., Yuan, Y., Wang, X., Huang, H., Wu, D.O., 2020. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. IEEE Trans. Veh. Technol. 69 (8), 8243–8256.

Xiao, N., Yu, L., Yu, J., Chen, P., Liu, Y., 2022. A cold-start-free reinforcement learning approach for traffic signal control. J. Intell. Transp. Syst. 26 (4), 476–485.

Yang, K., Menendez, M., 2018. Queue estimation in a connected vehicle environment: A convex approach. IEEE Trans. Intell. Transp. Syst. 20 (7), 2480–2496.

Ying, J., Feng, Y., 2024. Infrastructure-assisted cooperative driving and intersection management in mixed traffic conditions. Transp. Res. C 158, 104443.

Zhang, Z., Guo, M., Fu, D., Mo, L., Zhang, S., 2022. Traffic signal optimization for partially observable traffic system and low penetration rate of connected vehicles. Comput.-Aided Civ. Infrastruct. Eng..

Zhang, R., Ishikawa, A., Wang, W., Striner, B., Tonguz, O.K., 2020. Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. IEEE Trans. Intell. Transp. Syst. 22 (1), 404–415.

Zhao, Y., Zheng, J., Wong, W., Wang, X., Meng, Y., Liu, H.X., 2019. Various methods for queue length and traffic volume estimation using probe vehicle trajectories. Transp. Res. C 107, 70–91.