A new reinforcement learning-based variable speed limit control approach to improve traffic efficiency against freeway jam waves

Han, Yu; Hegyi, Andreas; Zhang, Le; He, Zhengbing; Chung, Edward; Liu, Pan

DOI
10.1016/j.trc.2022.103900

Publication date
2022

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Transportation Research Part C 144 (2022) 103900

Contents lists available at ScienceDirect

Transportation Research Part C

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

A new reinforcement learning-based variable speed limit control approach to improve traffic efficiency against freeway jam waves

Yu Han [a,*], Andreas Hegyi [b], Le Zhang [c], Zhengbing He [d], Edward Chung [e], Pan Liu [a,*]

[a] School of Transportation, Southeast University, Nanjing, China
[b] Department of Transport and Planning, Delft University of Technology, The Netherlands
[c] School of Economics and Management, Nanjing University of Science and Technology, Nanjing, China
[d] Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, China
[e] Department of Electrical Engineering, Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:
Variable speed limits
Freeway traffic control
Reinforcement learning
Data-driven approach

ABSTRACT

Conventional reinforcement learning (RL) models of variable speed limit (VSL) control systems (and traffic control systems in general) cannot be trained in real traffic process because new control actions are usually explored randomly, which may result in high costs (delays) due to exploration and learning. For this reason, existing RL-based VSL control approaches need a traffic simulator for training. However, the performance of those approaches are dependent on the accuracy of the simulators. This paper proposes a new RL-based VSL control approach to overcome the aforementioned problems. The proposed VSL control approach is designed to improve traffic efficiency by using VSLs against freeway jam waves. It applies an iterative training framework, where the optimal control policy is updated by exploring new control actions both online and offline in each iteration. The explored control actions are evaluated in real traffic process, thus it avoids that the RL model learns only from a traffic simulator. The proposed VSL control approach is tested using a macroscopic traffic simulation model to represent real world traffic flow dynamics. By comparing with existing VSL control approaches, the proposed approach is demonstrated to have advantages in the following two aspects: (i) it alleviates the impact of model mismatch, which occurs in both model-based VSL control approaches and existing RL-based VSL control approaches, via replacing knowledge from the models by knowledge from the real process, and (ii) it significantly reduces the exploration and learning costs compared to existing RL-based VSL control approaches.

1. Introduction

A jam wave, also known as wide moving jam or shock wave in some studies, e.g., Kerner and Rehborn (1996), Hegyi et al. (2005b), is a common type of traffic jams on freeways. A jam wave usually originates from a traffic breakdown that occurs due to high traffic demand, and its head and tail are both propagating upstream. From various empirical studies, some common features of jam waves are distilled. For example, the propagation speed of jam waves is roughly a constant, typically between 15–20 km/h (Kerner, 2002). It can propagate for a long time and distance, and resolves only when the traffic demand decreases (Kerner and Rehborn, 1996). The queue discharge rate from a jam wave is typically around 30 percent lower than the free-flow capacity (Schönhof and Helbing, 2007). Jam waves create many problems, including capacity reduction, travel delays, and safety risks. Therefore, eliminating jam waves can greatly improve freeway traffic operation efficiency.

* Corresponding authors.
E-mail addresses: yuhan@seu.edu.cn (Y. Han), liupan@seu.edu.cn (P. Liu).

https://doi.org/10.1016/j.trc.2022.103900
Received 24 November 2020; Received in revised form 11 August 2022; Accepted 17 September 2022
Available online 12 October 2022

One way to alleviate jam waves is to avoid the activation of infrastructural bottlenecks, e.g., on-ramp bottlenecks, by applying traffic control measures such as ramp metering and variable speed limits (Hadiuzzaman et al., 2013; Lu et al., 2015). As jam waves usually originate from standing queues that form at infrastructural bottlenecks, removing those bottlenecks may significantly reduce the occurrences of jam waves. However, due to the limited storage space at on-ramps, on-ramp bottlenecks cannot be fully avoided by ramp metering. On the other hand, variable speed limits (VSLs) can reduce the mainstream flow upstream of a bottleneck, so as to avoid the activation of the bottleneck. Carlson et al. (2011) proposed a feedback-based variable speed limit control method for local bottlenecks. Chen et al. (2014), Chen and Ahn (2015) developed analytical approaches of VSLs based on the kinematic wave theory for recurrent and non-recurrent infrastructural bottlenecks. Studies of Hegyi et al. (2005a), Lu et al. (2010), Zhang and Ioannou (2016), Carlson et al. (2014) combined VSLs with other control measures, such as ramp metering and lane-changing control, to improve traffic operation efficiency at infrastructural bottlenecks. Carlson et al. (2010b), Wang et al. (2020) proposed optimal control methods of VSLs for large scale freeway networks. The aforementioned VSL control approaches may create a high-density region in or upstream of the VSL-controlled area, which may trigger new jam waves. Hence, traffic control measures aiming for eliminating stationary bottlenecks may not be able to fully avoid the formation of jam waves.

Another way to alleviate jam waves is to suppress them after their formation using VSLs. There are different theories and algorithms to determine the parameter values of the VSLs. The SPECIALIST algorithm proposed by Hegyi et al. (2008) is an analytical approach for determining VSL parameters using the shockwave theory (Lighthill and Whitham, 1955; Richards, 1956). It was successfully implemented and tested in practice (Hegyi and Hoogendoorn, 2010). However, since the SPECIALIST algorithm has a feed-forward structure, disturbances that occur after the activation of a VSL scheme cannot be handled. Hegyi et al. (2005b) presented a model predictive control (MPC) approach of VSLs, where the design was based on a macroscopic second-order traffic flow model, METANET (Messmer and Papageorgiou, 1990; Kotsialos et al., 2002a). The nonlinear and non-convex formulation of METANET-based MPC approaches might result in high computation load, especially if the optimization is solved by the standard SQP algorithm (Hegyi et al., 2005a). Moreover, globally optimal VSL control is often unattainable for that type of approaches (Frejo and Camacho, 2012; Frejo et al., 2014). Studies of Muralidharan and Horowitz (2015), Roncoli et al. (2015), Hadiuzzaman and Qiu (2013), Han et al. (2017b), Zhang and Ioannou (2018) developed simpler MPC approaches that have less computational complexity based on the cell transmission model and its variants. However, those models cannot accurately reproduce the propagation of jam waves (Han et al., 2016). Han et al. (2017b, 2021) proposed MPC approaches of VSLs based on discrete first-order traffic flow models formulated in Eulerian and Lagrangian coordinates. Due to the linear formulations of the optimal controllers, those approaches significantly improved the computational efficiency. Despite the successful demonstration of the above MPC approaches via simulation, in general MPC for traffic systems are difficult to be implemented in practice, partially because MPC approaches are sensitive to the accuracy of the prediction models.

In recent years, data-driven approaches, such as reinforcement learning (RL), have attracted greater attentions in the realm of road traffic control as more traffic data become available. RL applications to road traffic control were initially investigated in urban traffic networks for traffic signal optimization problems (Arel et al., 2010; Prashanth and Bhatnagar, 2010; El-Tantawy et al., 2013; Li et al., 2016; Ozan et al., 2015). Regarding freeway traffic control, most of the RL applications focused on improving traffic operation efficiency at local bottlenecks. Davarynejad et al. (2011) addressed a local ramp metering problem considering the storage capacity of on-ramps using a Q-learning algorithm. Li et al. (2017) presented a Q-learning-based VSL control approach for recurrent freeway bottlenecks. Schmidt-Dumont and van Vuuren (2015) proposed a decentralized RL approach that integrated ramp metering and VSLs. Belletti et al. (2017) presented a deep RL-based ramp metering strategy. In a simulation test, the strategy achieved a control performance comparable to the classical feedback ramp metering method, ALINEA (Papageorgiou et al., 1991). Wu et al. (2020) proposed a deep actor–critic algorithm of lane-based VSLs to eliminate recurrent freeway bottlenecks. Han et al. (2022) proposed a physics-informed reinforcement learning approach for local and coordinated ramp metering.

Most of existing RL-based traffic control approaches train their RL models using traffic simulators. Therefore, similar as the aforementioned MPC approaches, which are sensitive to the accuracy of the prediction models, the control performances of those RL-based approaches are also dependent on the accuracy of the simulators. Nevertheless, the training processes of those RL models cannot be performed in real world. The reason is twofold. First, control actions are usually explored randomly in those approaches. Such way of action exploration can only be performed in a simulation environment, as a real traffic control system cannot accept randomly generated control actions that may lead to very poor traffic performance. Secondly, the training process with random exploration may require a large amount of training data, which may not be feasible to collect because the speed of data collection in real world is restricted by physical time and the "slowness" of the traffic process. Furthermore, training those RL models using historical field data is also infeasible. The reason is that effective training data collected from the field are lacking, as traffic flows in real world are regulated by a limited number of pre-defined control strategies. In addition, many practical traffic control systems are not used for eliminating traffic jams or improving traffic efficiency. For example, many traffic signal control systems and speed control systems in reality only implement fixed signal timing plans and fixed speed limit values. The field data collected from those control systems cannot be used for training a RL model. Therefore, it is still a challenge to develop RL-based traffic control strategies for real world implementation.

In this paper, we propose a new RL-based VSL control approach that trains the RL model based on both offline synthetic data and data collected from the real system, where the real data gradually replace the synthetic data. The proposed VSL control approach consists of an offline training process and an online control process, which interact iteratively. In the online control process data are collected of the states, control actions, and the related performances as they occur in the real traffic process. In the offline training process the data collected online are fed into a learning algorithm to update the control policy. To explore new control actions that may lead to a better traffic performance, synthetic data generated from a macroscopic traffic flow model are also added to

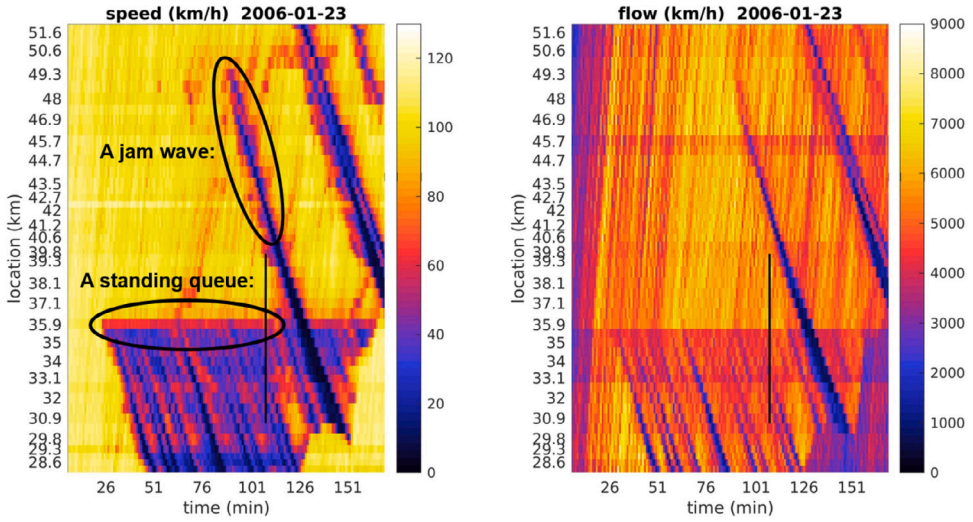

Fig. 1. Examples of a jam wave and a standing queue observed in real data. Data were collected from Dutch freeway A20 on January 23, 2006.

the training data set in the offline process. In the online control process, the VSL control policy obtained from the offline training process is applied to regulate traffic flow, and at the same time a new batch of data is collected. During the course of the iterations, the control performance is expected to improve as over time more real data are utilized by the RL.

The proposed approach is tested using the METANET model, which simulates real-world traffic flow dynamics. Therefore, in this paper the data generated from the METANET model are referred to as real data. To reproduce the difference between the traffic prediction model and the real traffic process, we use another traffic flow model, the extended cell transmission model (CTM), as the offline data generation model. Data generated from the extended CTM are referred to as synthetic data. To demonstrate the performance of the proposed approach against the model mismatch, it is compared with an MPC approach that uses the same extended CTM for prediction and an existing RL-based VSL control approach which also uses the same extended CTM for training. The proposed approach is also compared with an existing RL-based VSL control approach with random exploration to demonstrate the performance of reducing the exploration and learning costs.

The rest of this paper is organized as follows. Section 2 describes the VSL control problem. Section 3 presents the RL-based VSL control approach including the offline training and online control processes. Section 4 describes the simulation design for testing the proposed approach, and Section 5 discusses the simulation results. The conclusion and the topics for future research are discussed in Section 6.

2. The RL-based VSL control problem

This section presents the RL-based VSL control problem addressed in this paper. Section 2.1 describes the VSL control mechanism in resolving freeway jam waves. Section 2.2 defines the RL-based VSL control problem. A solution algorithm to that problem is presented in Section 2.3.

2.1. VSL control mechanism

As has been presented in Hegyi et al. (2008), two types of traffic jams are usually identified on freeways. Traffic jams with the head fixed at the bottleneck are known as standing queues, and jams that have an upstream moving head and tail are known as jam waves (also known as wide moving jams in some studies, e.g., Kerner and Rehborn (1996)). Fig. 1 shows a jam wave and a standing queue observed in real data. Both types of traffic jams can be eliminated by VSLs, based on two different mechanisms explained as follows.

Standing queues form at infrastructural bottlenecks, e.g., an on-ramp bottleneck or a lane-drop bottleneck. The VSL control strategies against infrastructural bottlenecks are developed based on the assumption that VSLs below the critical speed lead to a fundamental diagram that has lower capacity than under normal conditions. The application of VSLs upstream of a bottleneck permanently reduces the mainstream arriving flow, so as to avoid the bottleneck activation and the related throughput reduction as a result of the capacity drop. Then capacity flow can be established at the downstream bottleneck and the mainstream throughput is maximized, leading to a decrease of the total time spent. Fig. 2 shows the mechanism schematically. This mechanism forms the basis of the VSL control strategies in many studies such as Carlson et al. (2010a,b), Hadiuzzaman et al. (2013), Li et al. (2017), Wang et al. (2020).

The mechanism of VSLs in eliminating jam waves is different from that against standing queues. SPECIALIST is one of the earliest theories that systematically explained the mechanism of VSLs against jam waves (Hegyi et al., 2008). In Fig. 3, the time–space

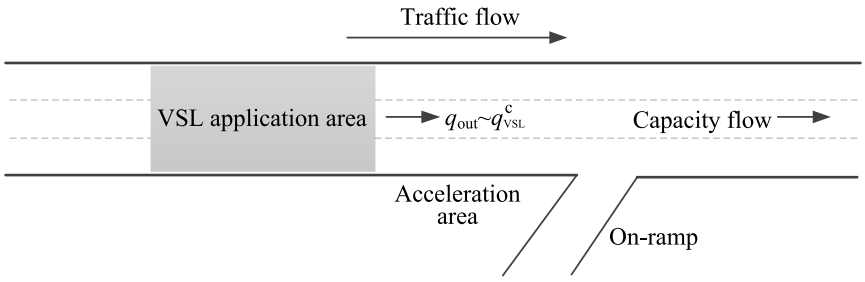

Fig. 2. The mechanism of VSLs against infrastructural bottlenecks. The on-ramp is a potential bottleneck. q_{out} is the outflow of the VSL- controlled area, and q^{c}_{VSL} is the VSL-induced capacity.

graph (left) shows the traffic states on a road stretch and their propagation over time. The density-flow diagram (right) shows the corresponding density and flow values for these states. According to kinematic wave theory, the boundary (front) between two states in the left figure has the same slope as the slope of the line that connects the two states in the right figure. Area 2 represents a jam wave that propagates upstream and which is surrounded by traffic in free-flow (areas 1 and 6). As soon as the jam wave is detected, VSLs are applied to the direct upstream of the jam wave, where the traffic state changes from state 6 to state 3. Subsequently, the size of the jam wave (area 2) is reduced because the inflow to the jam is lower than the outflow. The required length of the speed-limited stretch to resolve the jam depends on the density and flow associated with state 2 and the physical length of the detected jam. When the jam wave is resolved, there remains an area with the speed limits active (state 4) with a moderate density (higher than in free-flow, lower than in the jam wave). It was assumed that the traffic from area 4 can flow out more efficiently than a queue discharging from full congestion as in the shock wave (flow of state 2). This assumption was demonstrated in a later research by analyzing the data from SPECIALIST field test experiment (Hegyi and Hoogendoorn, 2010).

The similarity between these two VSL control mechanisms is that they both assume the traffic jams are associated with a capacity drop, and the major benefit of VSLs is to reduce travel delays by eliminating the capacity drop. The difference is that these two mechanisms eliminate capacity drop in different ways, which may lead to different consequences. The mechanism of VSLs against jam waves takes advantage of the transition flow created by VSLs, which only lasts for a relatively short period of time, just enough to resolve the jam. As it aims to keep the VSL-induced density at a moderate value, (e.g., area 4 in SPECIALIST), these VSLs can keep the traffic stable under the speed limits. It is assumed that the demand is always lower than the free-flow capacity so that the jam wave can be resolved without creating a new congestion. On the other hand, VSLs against standing queues do not need that assumption because even though the demand of the bottleneck exceeds the capacity, the traffic system still gets benefit from eliminating the capacity drop and maximizing the throughput. However, new congestion may be created when VSLs are applied to eliminate standing queues. For example, Papageorgiou et al. (2008) found that the VSL-induced capacity may be lower than the free-flow capacity. However, at a different site, Soriguera et al. (2017) could not identify any permanent flow reduction that could be attributed to VSLs, even when the speed limit value is as low as 40 km/h. Therefore, to create a sufficiently low flow to eliminate the standing queue under this circumstance, speed limits lower than 40 km/h will be needed. This will create new congestion at the upstream of the VSL-controlled area.

Most of the experiments (both simulations and field test) on VSLs against jam waves were performed in a homogeneous freeway stretch (Hegyi et al., 2008; Han et al., 2017b, 2021). For VSLs against standing queues, some strategies have been tested in larger sizes of freeway networks which include multiple on- and off-ramps via macroscopic simulations (Carlson et al., 2010b; Wang et al., 2020).

In this paper, we focus only on jam waves, and the mechanism of the VSLs follows the theory of SPECIALIST. From SPECIALIST field test experiment, it was summarized that some jam waves were not successfully resolved because the VSL-induced flows were not sufficiently low (Han et al., 2017a). In other failed cases, it was found that new jam waves were triggered at the upstream of the VSL-controlled area because the densities of this area were too high (Hegyi and Hoogendoorn, 2010). Therefore, an effective VSL control scheme to improve traffic efficiency against freeway jam waves should be able to (i) create sufficiently low flow to resolve the jam, and (ii) maintain the density of the VSL-controlled area at a moderate value so as to avoid triggering a new jam wave. In the next section we will formulate an RL controller, that is capable of both by using stretches of VSLs that are directly upstream of the jam and that can vary in length.

2.2. RL-based VSL control system

Reinforcement learning concerns the problem of a learning agent that interacts with its environment to achieve a goal (Sutton and Barto, 2018). The agent and the environment are generally interacting in discrete time steps. At each time step k, the agent takes an action $a(k)$ based on the state $s(k)$ received from the environment. The environment responds to the action by assigning a reward $r(k)$ to the agent and presenting a new state, $s(k+1)$. The agent's objective at time step k is maximizing the accumulative reward-to-go over a given time horizon,

$$G(k) = \sum_{\tau=k}^{K_T} \gamma^{\tau-k} r(\tau), \tag{1}$$

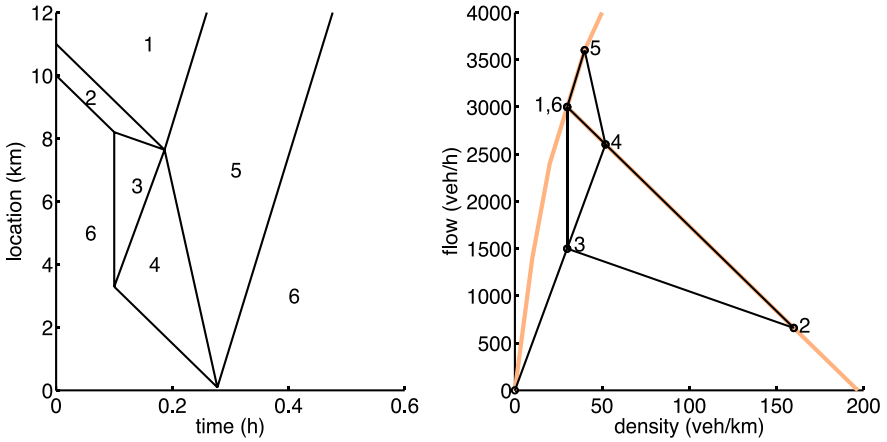

Fig. 3. Illustration of traffic evolution under the SPECIALIST (Hegyi et al., 2008). The left figure is the time–space graph and the right figure is the fundamental diagram. Areas 3 and 4 are the VSL-controlled areas.

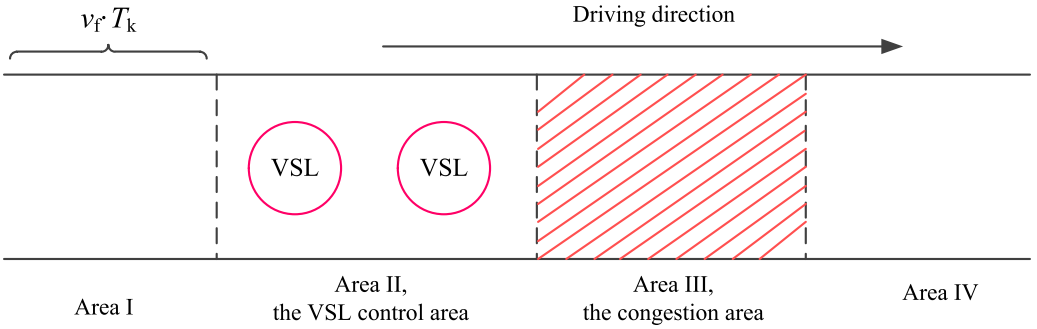

Fig. 4. Dividing the freeway stretch into four areas.

where K_T denotes the time index when the state of the environment reaches the terminal state; $r(\tau)$ the reward received at time τ; and $\gamma^{\tau-k}$ the discount factor ($0 \leq \gamma \leq 1$) that defines the relative importance of the reward at time τ.

In this paper, we consider an RL-based VSL control system, where the traffic dynamics on the freeway is the environment, and the VSL controller is the agent. More specifically, we consider a long homogeneous freeway stretch, which is suitable for applying VSLs to resolve jam waves. The agent decides about the speed limit values displayed to the drivers at different positions of the freeway. It is assumed that the freeway is equipped with fixed-location sensors, e.g., loop detectors, which divide the freeway into cells. Variable message signs (VMSs) which display the speed limit values are placed above the freeway.

The state, action, and reward of the RL system are defined considering the mechanism of VSLs as presented in the previous section. The freeway stretch is divided into four areas, which are indexed as I, II, III, and IV from upstream to downstream, as shown in Fig. 4. They represent the area upstream of VSL control (I), the VSL-controlled area (II), the jam area (III), and the area downstream of the jam (IV), respectively. Each area consists of a number of consecutive cells, so the area boundaries coincide with the cell boundaries. Area I has a length of $v_f \cdot T_k$, where v_f denotes the free-flow speed and T_k is the unit time step duration. Area II, the VSL-controlled area, denotes the freeway section that controlled by a number of consecutive VMSs, which display the speed limits. It is assumed that this area resides immediately upstream of the congestion area. Area III, the congestion area, consists of all the cells that are in congestion. Cell i is defined to be in congestion if $v_i \leq v_{jmax}$ and $q_i \leq q_{jmax}$ are both satisfied, where v_{jmax} and q_{jmax} are predefined speed and flow thresholds, respectively. Area IV covers the part where the discharging traffic recover to the free-flow speed. The length should be long enough for the acceleration, e.g., longer than 1 km. These four areas move along with the jam wave and their traffic states are updated in each control cycle accordingly. Note that the VSL controller is switched on when there is only one jam area on the freeway. When there are multiple, disconnected congestion areas, e.g., multiple jam waves, the VSL controller will not be activated.

As presented in the previous section, it is summarized that an effective VSL control scheme against freeway jam waves should be able to (i) create sufficiently low flow to resolve the jam, and (ii) maintain the density of the VSL-controlled area at a moderate value. Therefore, the state and action variables of the RL model should be able to capture the traffic dynamics of the jam and the VSL-controlled area. According to the conservation law, the dynamic evolution of a traffic jam is related to the size of the jam at the current time step, i.e., how many vehicles are in the jam, and the inflow and outflow of the jam. Likewise, the density variation of the VSL-controlled area is related to its original density, and the inflow and outflow of this area. Therefore, to resolve the jam

Table 1
The state and action variables of the proposed RL system.

State variables	$\bar{q}_{\mathrm{I}}(k)$ [veh/h]	The inflow to the VSL-controlled area, which is calculated as the arithmetic mean of the measured flow of all cells in area I.
	$\bar{\rho}_{\mathrm{V}}(k)$ [veh/km/lane]	The average density of the VSL-controlled area, which is calculated as arithmetic mean of the density of all cells in area II.
	$l_{\mathrm{jam}}(k)$ [km]	The length of the congestion area, i.e., area III.
	$\bar{v}_{\mathrm{jam}}(k)$ [km/h]	The average speed of the jam area, which is calculated as the arithmetic mean of the measured speed of all cells in area III.
	$P_{\mathrm{jam}}(k)$	The index of the most upstream cell of area III, the jam area.
Action variables	$V(k)$ [km/h]	The speed limit value
	$P_{\mathrm{V}}(k)$	The index of the most upstream cell of the VSL-controlled area

wave and also maintain the density of the VSL-controlled area at a moderate value, the state, action, and reward functions of the RL system should take all those variables into account.

To define the state, action, and reward, the VSL control system is discretized in time. The state of discrete time step k, $s(k)$, and the action, $a(k)$, are defined as:

$$s(k) = [\bar{q}_{\mathrm{I}}(k), \bar{\rho}_{\mathrm{V}}(k), l_{\mathrm{jam}}(k), \bar{v}_{\mathrm{jam}}(k), P_{\mathrm{jam}}(k)] \tag{2}$$

$$a(k) = [V(k), P_{\mathrm{V}}(k)], \tag{3}$$

where $V(k)$ denotes the speed limit value, and $P_{\mathrm{V}}(k)$ denotes the index of the most upstream cell of the VSL-controlled area. It is assumed that VSLs are applied directly upstream of P_{jam}, the most upstream cell of the jam area. Therefore, the variables, $V(k)$, $P_{\mathrm{V}}(k)$, and $P_{\mathrm{jam}}(k)$ can determine the speed limit value of every VMS. For other state variables, $\bar{q}_{\mathrm{I}}$ denotes the average flow of area I, which is considered as the arriving flow to the VSL-controlled area in one time step. $\bar{\rho}_{\mathrm{V}}$ represents the average density of the VSL-controlled area. l_{jam} and $\bar{v}_{\mathrm{jam}}$ are the length and average speed of the congestion area, respectively. These two variables represent the size of the jam wave. All the state and action variables are summarized in Table 1. Those state and action variables can effectively capture the traffic dynamics of the jam and the VSL-controlled area.

1. The state variable, $\bar{q}_{\mathrm{I}}(k)$, determines the inflow of the VSL-controlled area. The state variable, $\bar{\rho}_{\mathrm{V}}(k)$, and the action variable, $V(k)$, approximate the outflow of the VSL-controlled area. The state variable, $\bar{\rho}_{\mathrm{V}}(k)$, represents the density of the VSL-controlled area at the current time step. Therefore, the RL system captures the density variation of the VSL-controlled area based on those variables.
2. The state variables, $l_{\mathrm{jam}}(k)$ and $\bar{v}_{\mathrm{jam}}(k)$, represent the size of the jam at the current time step. The inflow to area III is equal to the outflow of area II. Thus, the state variable, $\bar{\rho}_{\mathrm{V}}(k)$, and the action variable, $V(k)$, approximate the inflow to the jam. According to the empirical study of Yuan et al. (2015), the outflow of a jam wave is dependent to the speed in the jam. Therefore, the state variable, $\bar{v}_{\mathrm{jam}}$, can capture the outflow of the jam. The RL system captures the dynamic evolution of the jam based on those variables.

For the presented VSL control system, the VSL controller is activated when a jam wave is detected and deactivated when it is resolved or considered as unresolvable. The jam wave is considered as being resolved if for every cell i, $v_i > v_{\mathrm{jmax}}$ and $q_i > q_{\mathrm{jmax}}$ are both satisfied. The jam wave is considered as unresolvable if the congestion has reached to the upstream boundary of the freeway stretch or multiple jam waves have been observed during the VSL control process.

It is assumed that for a single jam wave only one speed choice is applied to the entire VSL-controlled area over the control horizon. In other words, on detection of a jam wave the speed limit value is decided and it is kept unchanged until the speed limits are deactivated again. This setting is consistent with SPECIALIST, which has been implemented in practice. It requires less attention for drivers because they only need to decelerate once and accelerate after the jam wave being resolved. Therefore, this setting is more acceptable to the drivers and may also avoid possibly new breakdowns induced by a frequent acceleration and deceleration. For a different jam wave, however, the speed choice can be chosen from all available speed choices based on the learning result. Besides, when VSL control is activated, to avoid a sharp deceleration, the speed limit values are gradually reduced from the default speed limit to the target value.

The reward should reflect the improvement of traffic performance caused by VSLs. Intuitively, the reward should be a function of the freeway throughput, since the foremost improvement resulting from resolving jam waves is the elimination of capacity drop. Unfortunately, the throughput increment produced by the VSL control can hardly be observed until the jam wave is resolved. Thus, for a faster learning we define a reward function based on an artificial variable, $J(k)$, that represents congestion severity. The $J(k)$ is defined as follows:

$$J(k) = \frac{l_{\mathrm{jam}}(k)}{v_{\mathrm{jam}}(k)}. \tag{4}$$

The congestion severity decreases as the average speed in the congestion area increases and the length of the congestion area diminishes. The change of $J(k)$ may take place soon after the VSL is implemented, and before the jam wave is resolved. The reward,

$r(k)$, is defined as the reduction in congestion severity,

$$r(k) = J(k) - J(k+1). \tag{5}$$

2.3. The solution algorithm

The RL problem presented in the previous section can be solved by a number of methods (Sutton and Barto, 2018). In this section, we briefly introduce a model free Q-learning method, which has been extensively used in RL-based traffic control systems (Watkins and Dayan, 1992; Davarynejad et al., 2011; Li et al., 2017). To apply the Q-learning method, the variables in the state and reward functions, i.e., Eqs. (2) and (5) need to be discretized. The domain of each variable is divided into discrete intervals, and the value of each interval is represented by its midpoint.

The Q-learning method estimates the optimal value function Q^* using temporal-difference learning. The Q-value, $Q_{(s,a)}$, stores the value of a state–action pair, and it is updated according to:

$$Q_{(s,a)} \leftarrow Q_{(s,a)} + \kappa_{(s,a)}[r + \gamma \max_{a'} Q_{(s',a')} - Q_{(s,a)}] \tag{6}$$

where r is the observed reward of the transition from the current state s to the new state s' under action a; a' denotes the action chosen at state s'; $\kappa_{(s,a)}$ is the learning rate which controls how fast the Q-values are altered. Typically, the learning rate decreases over time to ensure convergence. Some studies, e.g., Li et al. (2017), defined the learning rate of a state–action pair as a function of the number of visits to that pair. In this paper we adopt the same method and define $\kappa(s,a)$ as:

$$\kappa_{(s,a)} = \left[\frac{1}{1 + C_{(s,a)}(1-\gamma)}\right]^{0.7} \tag{7}$$

where $C_{(s,a)}$ is the number of visits to the state–action pair (s,a).

For Q-learning, the selection rule for the action taken at a given state should consider the trade-off between exploitation and exploration. Even though using pure exploitation may greatly save the learning time, it may prohibit the discovery of new, potentially better actions. On the contrary, although pure exploration outperforms pure exploitation in the capability of discovering better actions, the former is quite time-consuming as it selects actions without making use of the learning results. In this paper, the method for the RL agent's exploration is referred to Li et al. (2017), in which the probability of selecting action a from state s is represented as:

$$p_s(a) = \frac{e^{Q_{(s,a)}/T}}{\sum_{a' \in A_s} e^{Q_{(s,a')}/T}} \tag{8}$$

where A_s is the set of available actions at state s; and T is the so-called temperature parameter. When T is large, each action would have approximately the same probability of being selected (more exploration). When T is small, actions would be selected in proportional to their estimated value (more exploitation).

The pseudocode of Q-learning is shown in Algorithm 1. $\mathbb{T}$ denotes the set of training data, where each training data slice is represented by a state transition tuple, $[s, a, s', r]$. $\mathbb{S}$ and $\mathbb{A}$ represent the set of states and the set of actions in the training data, respectively. ϵ denotes a very small positive value. The terminal state is defined as the state at the time the speed limit control is deactivated, i.e., when the jam wave is resolved or considered as unresolvable. Please be noted that any RL algorithm that can learn directly from data can be used in the proposed VSL control approach. In this paper, a simple Q-learning algorithm is used because we consider the amount of training data is relatively small.

Algorithm 1 The pseudocode of Q-learning.

Input: $\mathbb{T}$, $\mathbb{S}$, $\mathbb{A}$

Initialize $Q_{(s,a)} = 0$, $C_{(s,a)} = 0$, $\kappa_{(s,a)} = 1$, $\forall s \in \mathbb{S}$, $\forall a \in \mathbb{A}$;
repeat
  Initialize s;
  repeat
    choose a from s based on equation (8);
    $C_{(s,a)}$ += 1;
    update $\kappa_{(s,a)}$ based on equation (7);
    update $Q_{(s,a)}$ based on equation (4-6);
    $s \leftarrow s'$;
  until s is a terminal state
until convergence: $\sqrt{\sum_s \sum_a (Q_{(s,a)} - Q_{(s',a')})^2} \le \epsilon$

Output: $Q_{(s,a)}$, $\forall s \in \mathbb{S}$, $\forall a \in \mathbb{A}$

3. An iterative RL approach of VSLs

As explained earlier in this paper, the conventional training method for RL-based traffic control strategies, which solely relies on traffic simulators to generate the training data, is flawed because accurate traffic simulators are difficult to obtain (Papageorgiou,

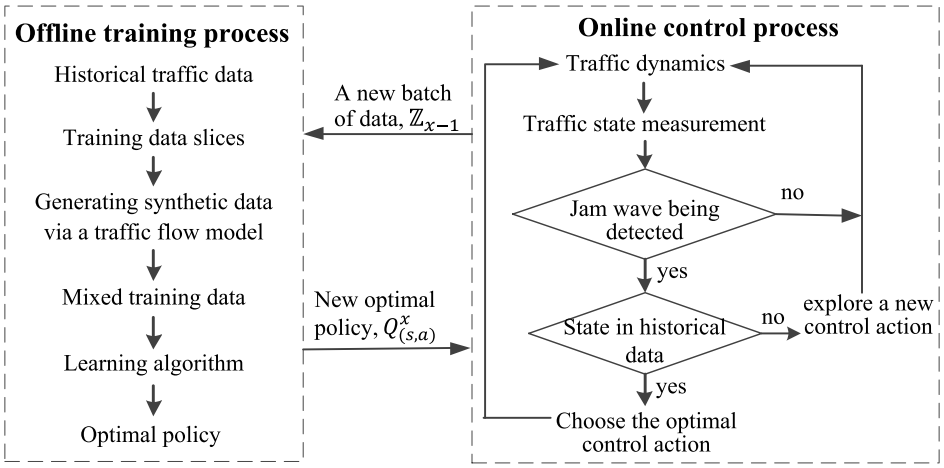

Fig. 5. The offline training process and online control process of the proposed approach in an iteration.

1998). In fact, the modeling errors of some well-known simulators were shown to be between 10%–20% (Spiliopoulou et al., 2014; Han et al., 2017a). Moreover, how the error of a traffic simulator affects the performance of the corresponding RL controller is still unclear. On the other hand, training the RL controller with field data is also infeasible because of random explorations. The control actions that are randomly explored may lead to very poor traffic performance. Furthermore, the training process with random exploration may require a large amount of training data, which may not be feasible to collect because the speed of data collection in real world is restricted by physical time.

In light of the above, our RL approach combines the two ways of training by using both offline simulation data and real data, where real data gradually replace simulation data. The proposed approach applies an iterative training framework, where the optimal control policy is updated by exploring new control actions from both online and offline in each training iteration. Section 3.1 presents the general framework of the proposed approach. Sections 3.2 and 3.3 explain the offline training and online control processes, respectively.

3.1. Framework of the iterative RL

The proposed iterative RL approach of VSLs consists of an offline training process and an online control process, which interact through the iterations. In each iteration, the interaction between those two processes is shown in Fig. 5. For the offline training process, the input includes historical data and a new batch of data collected from last iteration. Historical data are sliced in the form of state transition tuples, and added to the training dataset. To explore new control actions that may lead to better traffic performance, new synthetic data generated from a macroscopic traffic flow model based on the new batch of real data, are also added to the training dataset. The process of the offline synthetic data generation is presented in Section 3.2. The output of the offline training process is the Q-table that contains the Q-values, $Q_{(s,a)}$, of all available state–action pairs in the data.

After training, the optimal control policy is fed into the online control process. In each iteration, the VSL control policy associated with a fixed state-value table is implemented in the online process for a period of time. This duration is determined considering the trade-off between the control performance and the learning rate. If the time duration of each stage is too long, it would take more time for the VSL agent to improve the traffic performance. If the time duration is too short, the data gathered from the online process may be too limited for the RL agent to improve. A new control action is explored online only if the RL state is not in the Q-table. The online exploration that follows a certain of rules is presented in Section 3.3.

Exploration in RL is always the price to pay to improve the system performance. However, in real traffic system, there are many restrictions that limit the exploration of new control actions. For example, a poor exploration method, e.g., random exploration as in many existing RL systems, may lead to very poor traffic performance or even unsafe traffic situations. Furthermore, an inefficient exploration method may not lead to any improvement in real-world simply because of limited physical time. The presented offline/online exploration method prevents poor control actions being explored in real traffic process to some extent, so as to reduce the exploration and learning costs. With the interaction between the offline training and the online control, the optimal policy is updated iteratively. During the course of iterations, the traffic performance is expected to be improved with the updating of the optimal policy because the model mismatch is alleviated via replacing knowledge from the models by knowledge from the real process.

3.2. Offline training

The offline training process in an iteration is shown in the left block of Fig. 5. In the offline training process of iteration x, the set of training data slices, $\mathbb{T}_x$, which include both real data and synthetic data, are gathered and fed into Algorithm 1 to obtain the

Table 2
The notation of variables in Algorithm 2.

$\mathcal{T}$	A training data slice
$\widetilde{\mathcal{T}}$	A synthetic training data slice
$\mathbb{T}$	The set of training data slices
$\mathbb{T}^{\text{real}}$	The set of real training data
$\mathbb{T}^{\text{start}}$	The initial training data
x	Index of the iterations
m	Index of traffic data slice
$\mathbb{Z}$	The set of traffic data slices
z_m	Traffic state data slice m, $z = [\hat{\rho}, \hat{q}, \hat{v}]$
$\hat{\rho}, \hat{q}, \hat{v}$:	Vectors of density, flow, and speed of all the cells in the freeway network
s_m	The RL state of data slice m
$\tilde{A}_m$	The set of feasible synthetic control actions for data slice m
$\tilde{a}$	A synthetic control action
$\tilde{z}_m^{\text{next}}$	The synthetic future traffic state predicted on state z^m
$\tilde{s}_m^{\text{next}}$	The synthetic future RL state predicted on state z^m
$\mathcal{F}$	The operator of predicting traffic state using a traffic flow model
$\mathcal{H}$	The operator of transforming a traffic state to a RL state

Q-table. A training data slice is represented as state transition tuples in the form of $[s, a, s', r]$. In the training data, the real data is represented as $\mathbb{T}_x^{\text{real}}$, and,

$$\mathbb{T}_x^{\text{real}} = \mathbb{T}_x^{\text{start}} \bigcup_{n=1}^{x-1} \mathbb{T}_n^{\text{real}} \tag{9}$$

where $\mathbb{T}_x^{\text{start}}$ is the initial training data, e.g., the training data collected from the previous implementations of other VSL control strategies in the target site. If no VSL control strategy was implemented before, $\mathbb{T}_x^{\text{start}}$ is empty.

The synthetic training data of iteration x, is generated based on the set of traffic data slices, $\mathbb{Z}_{x-1}$, collected from the online control process in iteration $x-1$. Each traffic data slice is represented by the vectors of density, flow, speed of all the cells in the freeway network. For data slice $z_m \in \mathbb{Z}_{x-1}$, we use a traffic flow model to predict its future traffic state for one step ahead under all feasible new VSL control actions, represented by the set $\tilde{A}_m$. We define $\mathcal{F}$ as the operator of predicting future traffic state using the traffic flow model, and,

$$z_m^{\text{next}} = \mathcal{F}(z_m, \tilde{a}) \tag{10}$$

where $\tilde{a} \in \tilde{A}_m$. z_m^{next} represents the predicted traffic state. The predicted reward, $\tilde{r}$, can be obtained based on the predicted traffic state, according to (4–5). z_m^{next} is transformed to the corresponding RL state, $\tilde{s}_m^{\text{next}}$, according to the definition of the variables in (2). We define $\mathcal{H}$ to represent the operator of transforming a traffic state to a RL state, and,

$$\tilde{s}_m^{\text{next}} = \mathcal{H}(z_m^{\text{next}}). \tag{11}$$

We define $\mathbb{S}_x$ as the set of all the RL state observed in real data, $\mathbb{T}_x^{\text{real}}$. If the predicted RL state is in the set of RL state, i.e., $\tilde{s}_m^{\text{next}} \in \mathbb{S}_x$, then the synthetic data slice, $[s_m, \tilde{a}, \tilde{r}, \tilde{s}_m^{\text{next}}]$ is added to the training data set, $\mathbb{T}_x$. s_m is the RL state corresponding to traffic state z_m.

Therefore, in the offline process, new actions are generated by which the process can go from state s to state s', where both state s and state s' have been observed in real data but the transition has not yet been observed yet. We use this data generation method for two reasons. Firstly, the reliability of the explored control actions are dependent to the accuracy of the traffic prediction. Since the proposed method predicts traffic state transitions for only one step ahead, the prediction accuracy should be better than the prediction for multiple steps, in which the prediction error will be accumulated. Secondly, this method restricts the ratio of synthetic data in the training dataset. If the offline model also produces new states, the fraction of synthetic data may remain large and may remain dominant in the training data. Consequently, the model mismatch would not be alleviated.

By adding the information of this possible transition to the training data, new actions can be explored in the offline training process. In addition, the new action leading to s' will only be chosen in the online control process if the associated Q-value is high enough (based on earlier experiences), which will prevent choosing actions that lead to very poorly performing states. The pseudocode of the offline training process is shown in Algorithm 2, where the notation of variables can be found in Table 2.

In the training dataset, the ratio of real data increases with the number of iterations because only real data are accumulated. Similar to many heuristic exploration methods such as softmax, the proposed method also has a higher probability of exploration at the beginning of the training than at the end when the policy is close to the greedy policy. At the early stage of the iterations, the training data set contains a high proportion of synthetic data, enabling the RL agent to explore more actions. With an increasing number of iterations, the real data become dominant in the training data set, and the RL agent explores fewer control actions, to guarantee the improvement of traffic performance.

Algorithm 2 The pseudocode of the offline training process in iteration x.

Input: $\mathbb{T}_x^{\text{real}}$, $\mathbb{S}_x$, $\mathbb{A}_x$, $\mathbb{Z}_{x-1} = \{z_1, z_2, ..., z_M\}$;
$\mathbb{T}_x = \mathbb{T}_x^{\text{real}}$
for m=1, 2,...,M **do**
  $s_m = \mathcal{H}(z_m)$
  for $\tilde{a} \in \tilde{A}_m$ **do**
    $z_m^{\text{next}}, \tilde{r} = \mathcal{F}(z_m, \tilde{a})$
    $\tilde{s}_m^{\text{next}} = \mathcal{H}(z_m^{\text{next}})$
    if $\tilde{s}_m^{\text{next}} \in \mathbb{S}_x$ **then**
      $\widetilde{\mathcal{T}} = [s_m, \tilde{a}, \tilde{r}, \tilde{s}_m^{\text{next}}]$
      $\mathbb{T}_x \leftarrow \mathbb{T}_x \bigcup \{\widetilde{\mathcal{T}}\}$
    end if
  end for
end for
function Algorithm 1($\mathbb{T}_x$, $\mathbb{S}_x$, $\mathbb{A}_x$)
  return $Q_{(s,a)}$
end function
$Q_{(s,a)}^x = Q_{(s,a)}$
Output: $Q_{(s,a)}^x$, , $\forall s \in \mathbb{S}_x$, $\forall a \in \mathbb{A}_x$

3.3. Online VSL control

The online control process is shown in the right block of Fig. 5. First, the control system detects jam waves based on traffic flow measurements, as per the criteria presented in Section 2.2. The VSL controller is then activated once a jam wave is detected. For the first control step, k^*, if the RL state is in the RL state data set, i.e., $s(k^*) \in \mathbb{S}_x$, then the control action is decided by:

$$a(k^*) = \arg\max_a Q_{\left(s(k^*),a\right)}, \text{ if } s(k^*) \in \mathbb{S}_x, \tag{12}$$

where $a(k^*) = [V(k^*), P_V(k^*)]$. If the RL state is not in the RL state data set, then the control action of the first step will be determined by an existing VSL control strategy, e.g., SPECIALIST. For the subsequent control steps, the speed limit value $V = V(k^*)$ will be kept unchanged and only the boundaries of the VSL-controlled area are allowed to change. For step k, if the RL state, $s(k)$, is in the state data set, the controller exploits the optimal policy to give the optimal control action. Among all the state–action pairs associated with that state, the action that produces the largest Q-value is chosen and implemented to the traffic process:

$$a(k) = \arg\max_a \left\{Q_{\left(s(k),a\right)} \mid a = [V, P_V], V = V(k^*)\right\}, \text{ if } s(k) \in \mathbb{S}_x \tag{13}$$

If the RL state is not in the state value table, a new control action will be explored and implemented in the traffic process. For the new control action, it is assumed that the speed limit value is the same as it was in the previous control step, i.e., $V(k+1) = V(k)$, and the index of the most upstream cell of the VSL control area changes no more than 1, i.e., $|P_V(k) - P_V(k+1)| \le 1$. Note that this constraint not only prevents frequent acceleration and deceleration of drivers caused by VSLs, but also reduces the exploration space in the RL. If the exploration space is too large, finding the actions that improve the system performance may take unrealistically long time. For these states, that do not exist in the state value table, we apply a simple method to determine $P_V(k)$. The method intends to keep the VSL-controlled area at a moderate value. Specifically, we define a tuning parameter ρ_V^{cr}, which represents the critical density of the VSL-controlled area, and use ρ_V^{up} to represent the density of the most upstream cell of the VSL-controlled area. For ρ_V^{up} in two consecutive steps, $\rho_V^{\text{up}}(k-1)$ and $\rho_V^{\text{up}}(k)$, there are four possible situations:

① The density is lower than the critical value and it is decreasing: $\rho_V^{\text{up}}(k) \le \rho_V^{\text{cr}}$ and $\rho_V^{\text{up}}(k) \le \rho_V^{\text{up}}(k-1)$;

② The density is lower than the critical value and it is increasing: $\rho_V^{\text{up}}(k) \le \rho_V^{\text{cr}}$ and $\rho_V^{\text{up}}(k) > \rho_V^{\text{up}}(k-1)$;

③ The density is higher than the critical value and it is decreasing: $\rho_V^{\text{up}}(k) > \rho_V^{\text{cr}}$ and $\rho_V^{\text{up}}(k) \le \rho_V^{\text{up}}(k-1)$;

④ The density is higher than the critical value and it is increasing: $\rho_V^{\text{up}}(k) > \rho_V^{\text{cr}}$ and $\rho_V^{\text{up}}(k) > \rho_V^{\text{up}}(k-1)$.

For situation ①, the upstream boundary of the VSL-controlled area moves one cell downstream. For situations ② and ③, that upstream boundary remains at the same position. For situation ④, that upstream boundary moves upstream by one cell. Therefore, $P_V(k)$ is determined as:

$$P_V(k) = \begin{cases} P_V(k) - 1, \text{ if } ①,\ s(k) \notin \mathbb{S} \\ P_V(k), \text{ if } ② \text{ or } ③,\ s(k) \notin \mathbb{S} \\ P_V(k) + 1, \text{ if } ④, s(k) \notin \mathbb{S}. \end{cases} \tag{14}$$

4. Simulation experiment design

This section presents the simulation experiments for testing the proposed VSL control approach. The purpose of the simulations is to show that the proposed approach (i) can effectively eliminate jam waves and reduce travel delays, (ii) performs better than those approaches affected by the model mismatch, and (iii) has less exploration and learning costs compared to a RL method with random explorations. The following experiment scenarios are designed.

1. Testing the proposed iterative RL approach of VSLs using macroscopic traffic simulation. The purpose of this scenario is to investigate the performance of the proposed approach in reducing travel delays during the iterative training process. As it is impossible to directly test the proposed approach in the field, the METANET model is used as the process model to represent the real-world traffic flow dynamics. For this scenario, the overall framework is presented in Section 4.1. The process model and simulation settings are presented in Section 4.2. The parameter settings of the RL controller are presented in Section 4.3.
2. Compare the proposed approach to SPECIALIST. In SPECIALIST, the traffic state transitions under VSLs are predicted based on kinematic wave theory. The accuracy of the prediction is influenced by the tuning parameters and some external disturbances such as demand fluctuations. Therefore, the mismatch between the prediction results and real process may affect the control performance. The purpose of this scenario is to demonstrate the proposed approach can outperform SPECIALIST in terms of reducing travel delays by eliminating the model mismatch. Parameter settings of SPECIALIST are presented in Section 4.4.
3. Compare the proposed approach to an existing MPC approach against freeway jam waves (Han et al., 2017b). The MPC approach was developed based on the extended CTM (Han et al., 2016). As the prediction model of the MPC is different from the process model (METANET), the performance will be affected by the model mismatch. This scenario intends to demonstrate that the proposed approach can outperform the MPC approach in terms of reducing travel delays by alleviating the model mismatch,
4. Compare the proposed approach to an existing RL-based VSL control approach with random online exploration. In this scenario, the RL model is directly trained in the real traffic process using the DDQN algorithm. As the DDQN explores control actions randomly, the exploration and learning costs during the training may be very high. For example, a randomly explored control action may lead to very poor traffic performance and even increase the travel delay. The purpose of this scenario is to demonstrate that the proposed approach has much less exploration and learning costs than the random exploration method.
5. Compare the proposed approach to an existing RL-based VSL control approach with zero-shot policy transfer. In this scenario, an existing deep reinforcement learning algorithm, namely Double DQN (DDQN, Van Hasselt et al. (2016)), is used as the training algorithm. The same extended CTM model is used as the training environment. After training, the optimal policy is directly transferred to the real traffic process. As the training environment is different from the real traffic process, this RL-based approach is also affected by the model mismatch. The purpose of this scenario is to demonstrate that the proposed approach can outperform this RL-based approach by alleviating the model mismatch.
6. Compare the proposed approach to an existing RL-based VSL control approach with continually online learning. In this scenario, the DDQN-based VSL control strategy in scenario 5 is assumed to continually learn from the online environment after the offline optimal policy was transferred. This scenarios intends to investigate if the DDQN can continually improve traffic performance in the online environment, and also to quantify the online learning cost of the DDQN.

The simulation results of those five experiment scenarios are presented in Sections 5.1–5.5, respectively.

4.1. Overall framework of experiment scenario 1

The simulation experiment for testing the proposed approach includes the following steps.

1. Implementing the starting VSL control approach. We assume SPECIALIST as the starting VSL approach, which was applied before implementing the proposed approach. Therefore, in (9), $\mathbb{T}_x^{\text{start}}$ is the set of training data collected from SPECIALIST implementation. The time period of implementing SPECIALIST is represented by 100 online simulations where in each simulation one jam wave is artificially created.
2. The offline–online interaction process. The iterations start from the offline training process. In the offline, the synthetic data are generated from the extended CTM, which is briefly presented in Section 4.5. In each iteration, the optimal VSL control policy associated with a fixed state-value table is implemented in the online process for a period of time, represented by 100 online simulations. Other parameters of the RL controller are specified in Section 4.4.
3. Stop criterion. In the online control process, if the RL state is in the state-value table, i.e., the state has appeared in historical traffic data, the action that produces the largest Q-value is implemented by the process. If the RL state is not in the state-value table, a new control action will be implemented. We define the actions selected from the state-value table as RL control actions. In each stage, the total number of RL control actions (N_x^{RL}) and the total number of all the control actions (N_x) are recorded. The ratio between N_x^{RL} and N_x, denoted as η_x, represents the percentage of the states that has appeared in historical data. In general, η_x should be higher with the increment of the number of iterations and the expansion of training data. The experiment ends if η_x is larger than 0.8, when a large percentage of the states has appeared in historical data.

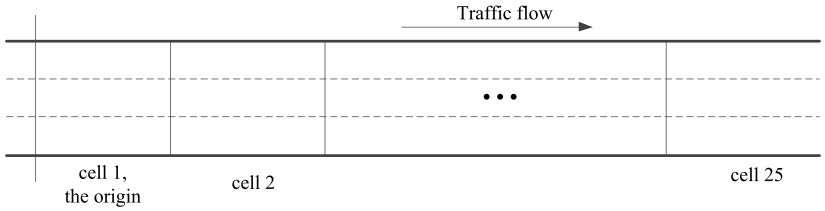

Fig. 6. A graphical representation of the synthetic freeway stretch.

Note that two different macroscopic traffic flow models are used in the simulation experiments. The METANET model is used as the process model which represents the real-world traffic flow dynamics. The extended CTM is used as the offline data generation model. Therefore, the simulations using METANET are referred to as online simulations and the simulations using the extended CTM are referred to as offline simulations.

The stochasticity of traffic flow is considered in the experiment, by incorporating noises to the process model for different jam waves. Detailed settings about the process model is presented in Section 4.2. The simulation experiment is repeated for 20 times to avoid getting unreliable results due to the stochasticity of the simulation environment.

4.2. The METANET model and simulation settings

The second-order macroscopic traffic flow model METANET, proposed by Messmer and Papageorgiou (1990), Kotsialos et al. (2002b), has been extensively used for freeway traffic simulation. The METANET model predicts the dynamic evolution of traffic speeds based on a steady speed–density relation and some heuristic terms that express driver behavior. Hegyi et al. (2005a) has extended the METANET model to account for the effect of VSLs. The model with VSLs extension has been validated using field data (Han et al., 2017b; Frejo et al., 2019). In this simulation test, the model presented in Hegyi et al. (2005a) is utilized as the process model. The simulation test uses the METANET model to represent real-world traffic dynamics. The reason we choose METANET as the process model is that it has been validated to reproduce the propagation of jam waves with a reasonable accuracy (Han et al., 2017a; Frejo et al., 2019). Furthermore, it runs much faster than microscopic simulations.

In the METANET model, the freeway is divided into cells which have a uniform geometric structure. For cell i, the desired speed at time t is calculated as:

$$V(\rho_i(t)) = \min\left(V_{\mathrm{C},i}(t), v_{\mathrm{f},i}\cdot\exp\left(-\frac{1}{a_m}\left(\frac{\rho_i(t)}{\rho_{\mathrm{cr},i}}\right)^{a_m}\right)\right), \tag{15}$$

where the first term, $V_{\mathrm{C},i}$, is the speed limit of cell i. We assume that the drivers fully comply with the speed limit control. The second term describes the steady speed–density relation of the model, which is characterized by three parameters, namely a_m, $v_{\mathrm{f},i}$ and $\rho_{\mathrm{cr},i}$. In the fundamental diagram, $v_{\mathrm{f},i}$ and $\rho_{\mathrm{cr},i}$ represent the free-flow speed and the critical density, respectively. For the sake of compactness, the equations that describe the traffic dynamics of METANET are shown in Appendix A.

Most of the experiments on VSLs against jam waves (both simulations and field test) are performed on a homogeneous freeway stretch. In the experiments, a three-lane synthetic freeway stretch is used as the test bed for the proposed VSL control approach. The homogeneous freeway stretch is 7.5 km in length, and it is divided into 25 cells. A graphical representation of the synthetic freeway is shown in Fig. 6. The parameter values of the process model are taken from Kotsialos et al. (1999), Hegyi et al. (2005a), Han et al. (2017b). Specifically, $\rho_{\mathrm{cr}} = 27.6$ veh/km/lane, $a_m = 2.5$ for every cell, and $v_{\mathrm{f}} = 108$ km/h.

In practice, traffic flow conditions (e.g., traffic demand and capacity) may vary from day to day. To reproduce the stochastic feature of traffic flow, we assume that parameters v_{f}, a_m, and ρ_{cr}, which influence the shape of the fundamental diagram, are stochastic. Each of the three parameters is assumed to follow a Gaussian distribution, where the mean is equal to the referred value and the standard deviation is 2% of the mean. Therefore, in each online simulation run, a sample of these parameters is taken. This gives us (slightly) different sizes of fundamental diagrams for different simulation runs. Fig. 7(a) shows the free-flow capacities obtained from 100 random online simulation runs. In general, the free-flow capacity in most of the online simulation runs ranges from 1900 veh/h/lane to 2100 veh/h/lane. Furthermore, to reproduce traffic demand fluctuations in reality, the demands in the online simulation runs are assumed to follow Gaussian distribution. Specifically, each online simulation run lasts for 2 h, including one hour of peak time and one hour of off-peak time. The mean of peak hour demands and mean off-peak hour demands are set to 90% of the capacity (which varies in different simulation runs) and 4000 veh/h respectively. The standard deviations are set to 5% of the mean.

Jam waves in reality usually form at a relatively fixed location of a site (Hegyi and Hoogendoorn, 2010). In the simulations, jam waves are artificially triggered at the downstream boundary of the freeway stretch. The densities downstream of the freeway stretch are set to 100 veh/km/lane at min. 32–34. To give an impression of the resulting stochasticity, we run the simulation for 100 times applying the presented demand and parameter settings. The density-flow plot, taken from the data of every cell in every minute, is shown in Fig. 7(b). The length of congestion area in those created jam waves varies from 0.9 km to 2.4 km, which is consistent with empirical observation. Fig. 8 shows an example of the simulated jam waves.

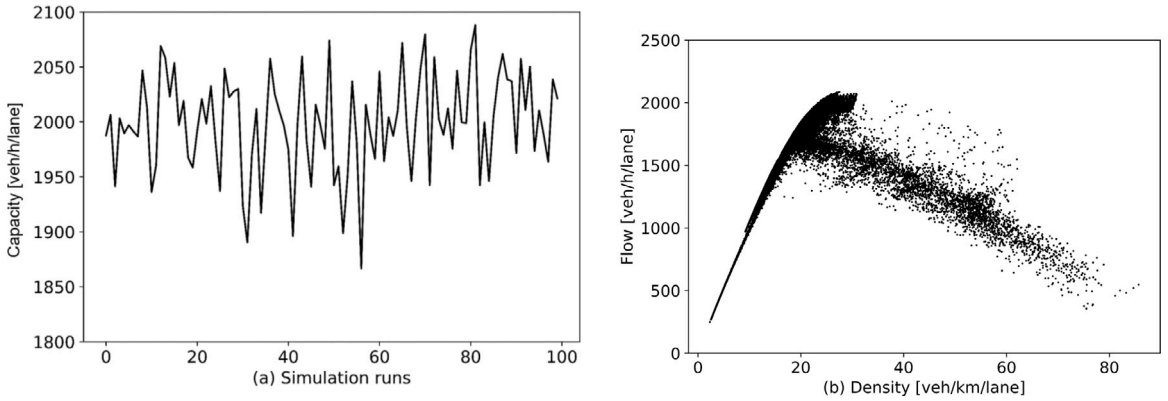

Fig. 7. Results of 100 online simulation runs: (a) The road capacity of each online simulation run. (b) The density-flow plot of all cells.

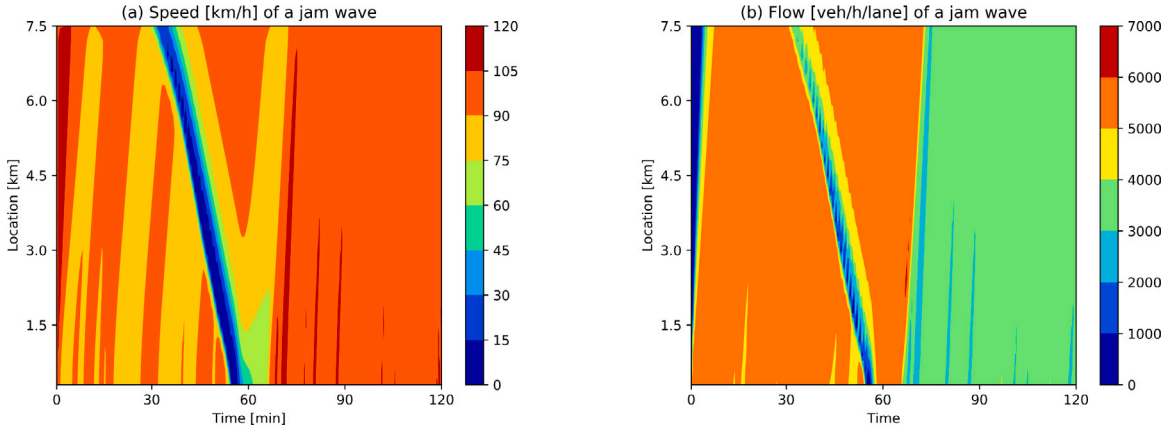

Fig. 8. (a) Speed and (b) Flow contour plots of an example of the simulated jam waves.

Table 3
The discrete intervals and the upper and lower bounds of the state and action variables.

Variables	Discrete intervals	Upper bounds	Lower bounds
$\tilde{q}_{\mathrm{I}}$ [veh/h]	100	2000	1000
$\tilde{\rho}_{\mathrm{V}}$ [veh/km/lane]	2	100	10
l_{jam} [km]	0.3	3	0.3
$\tilde{v}_{\mathrm{jam}}$ [km/h]	5	50	5
P_{jam}	1	25	1
P_{V}	1	24	1
V	10	60	50

4.3. Settings of the RL controller

In the offline training process of the proposed VSL control approach, the state and reward variables need to be discretized. The domain of each variable is divided into discrete intervals, and the value of each interval is represented by the midpoint. The discrete interval sizes of q_{I}, ρ_{V}, l_{jam}, v_{jam}, and P_{jam} are set to 100 veh/h/lane, 2 veh/km/lane, 0.3 km, 5 km/h, and 1 cell respectively, which considers the trade-off between data resolution and variable space. The discrete intervals and the upper and lower bounds of the state and action variables are summarized in Table 3. A penalty of −200 min is added to the terminal state, if the jam wave is not successfully resolved. In the Q-learning, the convergence threshold ϵ is set to 0.01 min.

In the proposed control system, we assume that two values of speed limit are used: 50 km/h and 60 km/h. Those two values are chosen based on both empirical evidence and trial-and-error tuning. From extensive simulation tests it is found that (i) a speed limit lower than 50 km/h would result in a higher density in the VSL-controlled area, which increases the risk of inducing new traffic breakdowns, and (ii) a speed limit value higher than 60 km/h may not be able to trigger a sufficiently low flow that can resolve the jam waves. For reader's reference, the displayed speed limit value in SPECIALIST system is 60 km/h (Hegyi and Hoogendoorn, 2010).

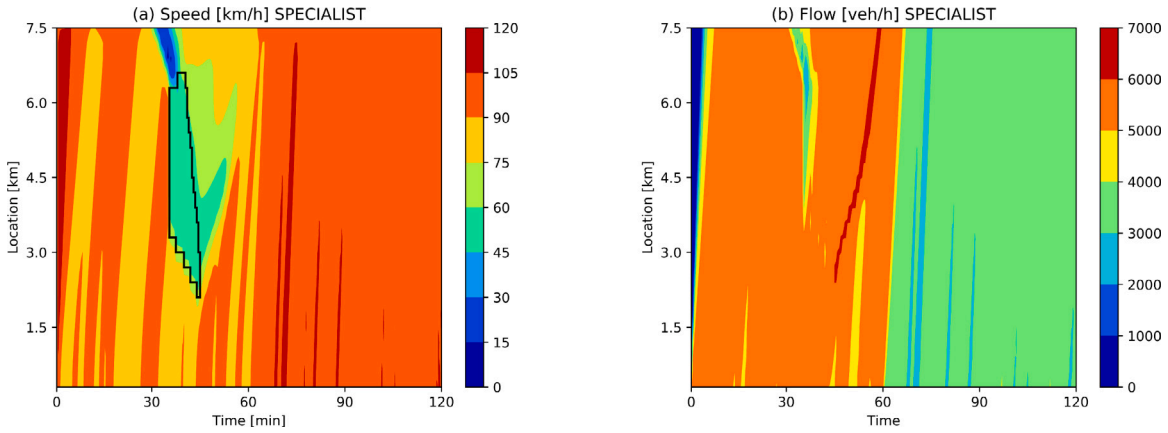

Fig. 9. (a) Speed and (b) Flow contour plots of an example in the simulation in which the jam wave is successfully resolved by SPECIALIST. In (a), the VSL-controlled area is enclosed by black lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For some traffic situations, the traffic performance may be further improved if more speed limit values can be displayed. However, the solution space of the RL increases exponentially with the size of action space, which may require an impractical amount of time to gather sufficient training data for the RL agent to improve the traffic performance. Therefore, the number of speed limit values is determined by considering the trade-off between potential traffic improvement and the time required to achieve the improvement.

In the online control process, the duration of a control time step, T_k, is set to 30 s. When VSL control is activated, to avoid a sharp reduction of speed limit, i.e., from the free-flow speed to 50 km/h or 60 km/h, 100 and 80 km/h are used for the lead-in. The same approach was used in Hegyi and Hoogendoorn (2010). The critical speed of the VSL control region, ρ_V^{cr}, is set to 30 veh/h/lane. The VSL control is deactivated when the jam wave is resolved.

4.4. The starting VSL control approach

We assume the SPECIALIST algorithm as the original VSL control approach before the RL-based VSL control approach is implemented. SPECIALIST has multiple tuning parameters, which have clear physical interpretations. These parameters can be tuned based on heuristic tuning rules using offline traffic data. In this simulation test, we mimic the implementation of SPECIALIST in the METANET simulation. A brief introduction of SPECIALIST and the tuning rules of parameters are presented in Appendix B. Fig. 9 shows an example in the simulation in which the jam wave is successfully resolved by SPECIALIST.

4.5. Model mismatch

In scenario 1 of the simulation experiments, we use the extended CTM model, proposed by Han et al. (2016), as the offline data generation model. The model extends the original CTM to reproduce capacity drop and the propagation of jam waves. Since there is always mismatch between real traffic process and a traffic simulation model, we choose the extended CTM model, which has a different mechanism as the METANET model, as the offline synthetic data generation model to reproduce such mismatch.

Although the process model (the METANET model) and the offline data generation model (the extended CTM) have some similarities, e.g., both of them assume a fundamental diagram for homogeneous traffic state, their mechanisms are still quite different. For example, the METANET model considers driver behavior in traffic speed dynamics such as anticipation to spatially increasing or decreasing densities, while the extended CTM does not. In the simulation experiment, the extended CTM model is calibrated with the simulation data from the METANET model. For a detailed presentation about the extended CTM, readers are referred to Han et al. (2016, 2017b).

Furthermore, in scenario 3 of the simulation experiments, the extended CTM is used as the prediction model of an MPC controller of VSLs for comparison. In scenario 4, the training environment of an existing RL-based VSL control strategy, which is used for comparison, is also developed based on the extended CTM. In those two scenarios, the model mismatch can be reproduced as a result of the difference between METANET and the extended CTM.

5. Simulation results and analysis

This section presents the results of the simulation experiments, and each sub-section corresponds to one of the experiment scenarios described in Section 4.

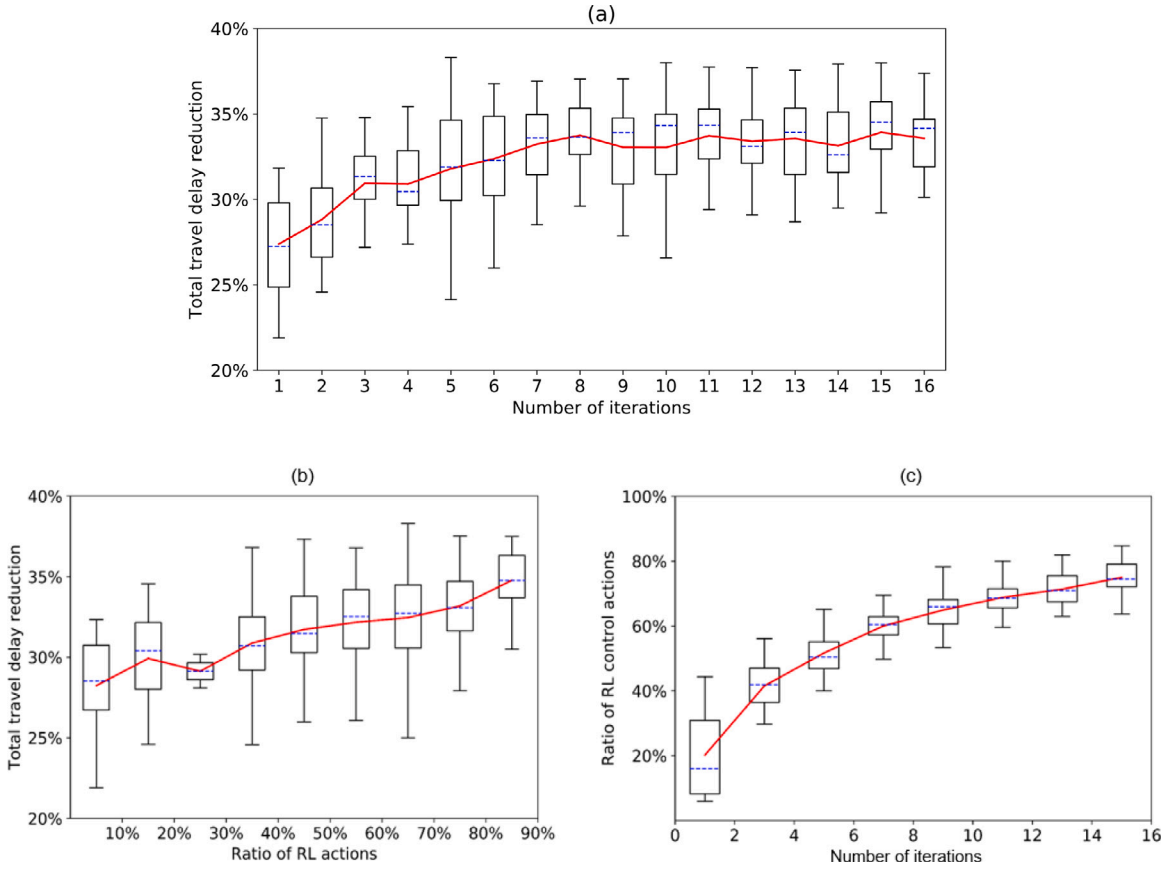

Fig. 10. (a): The whisker plot of total travel delay reduction (compared to those without VSL control) for different iterations; (b) total travel delay reduction for different ratios of RL actions; (c): the share of the ratio of RL control actions for different iterations.

5.1. Performance of the proposed approach

This section presents the results of the simulation experiment in testing the proposed approach, described as scenario 1 in Section 4. In the simulation experiment, 100 online simulations are performed in each iteration. The traffic performance at each iteration is evaluated using the average total travel delay as the performance indicator, which is calculated as the difference between the total time spent by all vehicles in the freeway stretch and the sum of all the vehicles' free-flow travel time. The simulation experiment is repeated for 20 times to avoid getting unreliable results due to the stochasticity of the simulation environment. A whisker plot that depicts the traffic performance of the proposed VSL control approach is shown in Fig. 10(a). The average total travel delay saving of the proposed approach is 31.3% during the entire training process.

Fig. 10(b) shows the total travel delay improvement of the presented VSL control approach with different values of η. In the figure, each box represents a 10 percent interval of that ratio. The dashed blue line in each box indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points. The red line represents the average total travel delay reduction for different intervals of η. In general, the average total travel delay saving increases with η, except for the interval [20%, 30%], where only three data points are observed. Moreover, it can be observed that the lower bound of the total travel delay reduction also increases when η is higher than 50%, which indicates that the presented VSL control approach becomes more robust as η grows. These results are as expected, because with the increment of η, more control actions are explored and more data are utilized by the RL. Hence, the actions selected by the RL controller becomes more reliable, because the RL controller takes the stochasticity of traffic environment into account.

Fig. 10(b) shows the change of η with the increment of the number of iterations. The average number of iterations in the simulation experiment is 15.9. The offline training time of the RL agent varies from less than one minute to 5 min. During earlier stages when the amount of training data is less, it takes less time for the Q-learning to converge.

5.2. Comparison with SPECIALIST

This section presents the proposed approach and SPECIALIST, described as scenario 2 in Section 4. SPECIALIST is utilized as the starting VSL control strategy in scenario 1 of the simulation experiments. The average total travel delay reduction of SPECIALIST

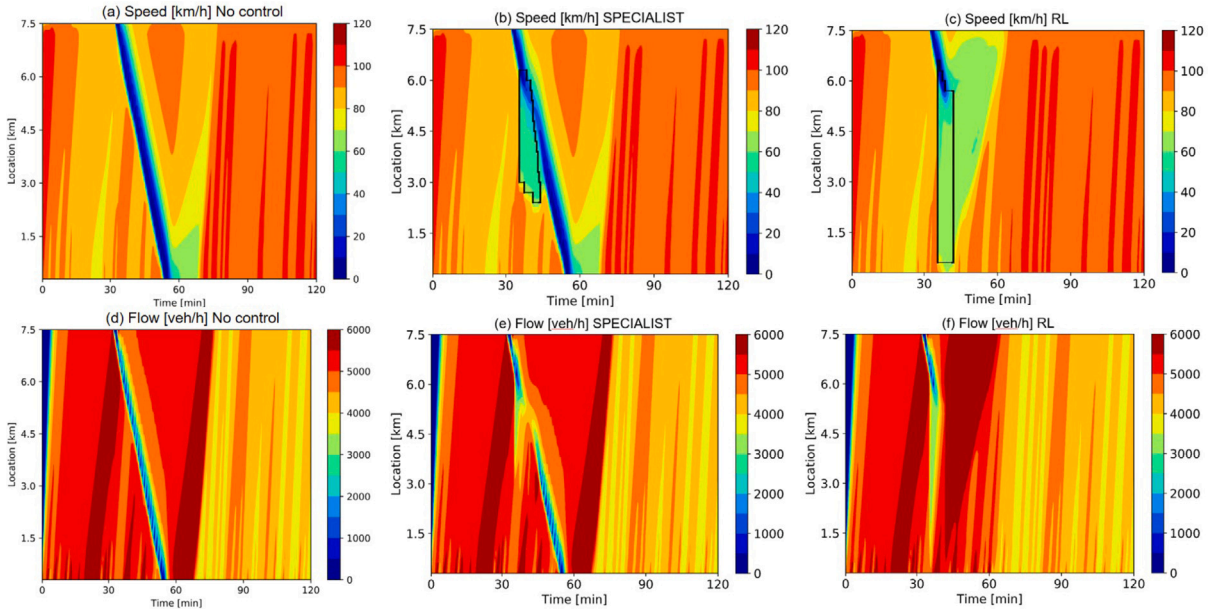

Fig. 11. Comparison between SPECIALIST and the proposed VSL control approach in an example. In this example, (a) and (d) are the simulated speed (km/h) and flow (veh/h) contour plots without VSL control; (b) and (e) are the simulation results under SPECIALIST; (c) and (f) are the simulation results under the proposed VSL control approach. In (b) and (c), the VSL-controlled areas are enclosed by black lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is 15.5%. In all the simulation runs, about 70% of the jam waves are classified as resolvable and the VSL schemes generated from SPECIALIST are implemented in those cases. Among the cases where VSLs are implemented, over 60% of the jam waves are successfully resolved. Some of the failures are attributed to the mismatch between the predicted traffic dynamics and real traffic process, for example, a sudden demand increment.

In contrast, the proposed VSL control approach reduces the average total travel delay by 35.1% when η is larger than 0.8. About 80% of the jam waves are resolved by VSLs, which is much higher than the SPECIALIST algorithm. Its better performance is attributed to two main reasons. First, the RL controller has a feedback structure. It determines the VSL control actions based on the online measured traffic states. It is thus able to handle disturbances such as demand increases. Second, the RL controller does not rely on online traffic prediction, because optimal control actions are obtained mainly from real traffic data.

Fig. 11 shows an example of comparison between the SPECIALIST and the proposed VSL control approach. In this example, both VSL control approaches are tested using the same demand profile and the same parameter values for the process model. Under SPECIALIST, the VSL-controlled area is too short to generate a transition flow that lasts long enough to resolve the jam wave. The reason is that the outflow of the jam (flow of area 1 in Fig. 3) is overestimated. Moreover, as SPECIALIST has a feed-forward control structure, it is very sensitive to the errors of traffic flow prediction. By contrast, the RL-based controller successfully resolves the jam wave.

It is worthy to be noted that we have tried a different set of SPECIALIST parameters. Although the performance of SPECIALIST with the new parameters is inferior, the performance of the proposed approach using the inferior tuning of SPECIALIST, is not affected. It still reduces the total travel delay by 35% at the end of the training. The reason is that the proposed approach explores new control actions and evaluates them in every stage. The actions that lead to a good traffic performance are kept, and the actions that lead to a worse traffic performance are discarded by the RL model. When the amount of training data becomes sufficiently rich, SPECIALIST data are only a small proportion of the training data and overruled by the real data. Therefore, the performance of the proposed approach is not sensitive to the tuning parameters of SPECIALIST.

5.3. Comparison with a MPC approach

This section presents the results of the comparison between the proposed VSL control approach and the MPC approach, described as scenario 3 in Section 4. The same extended CTM is used for traffic prediction in the MPC. The MPC has a feedback control structure, and the optimal VSL control scheme is calculated in every control step based on traffic state feedback. The prediction horizon is set to 20 min. The duration of a control step is set to 30 s. Model parameters are calibrated with the online simulation data. The minimum VSL value is set to 50 km/h in the optimization of the MPC. At each optimization iteration, the traffic demand in the prediction is set to a constant value for the entire prediction horizon, and the value is predicted as the measured average demand of last 15 min. For a full presentation of the MPC, readers are referred to Han et al. (2017b).

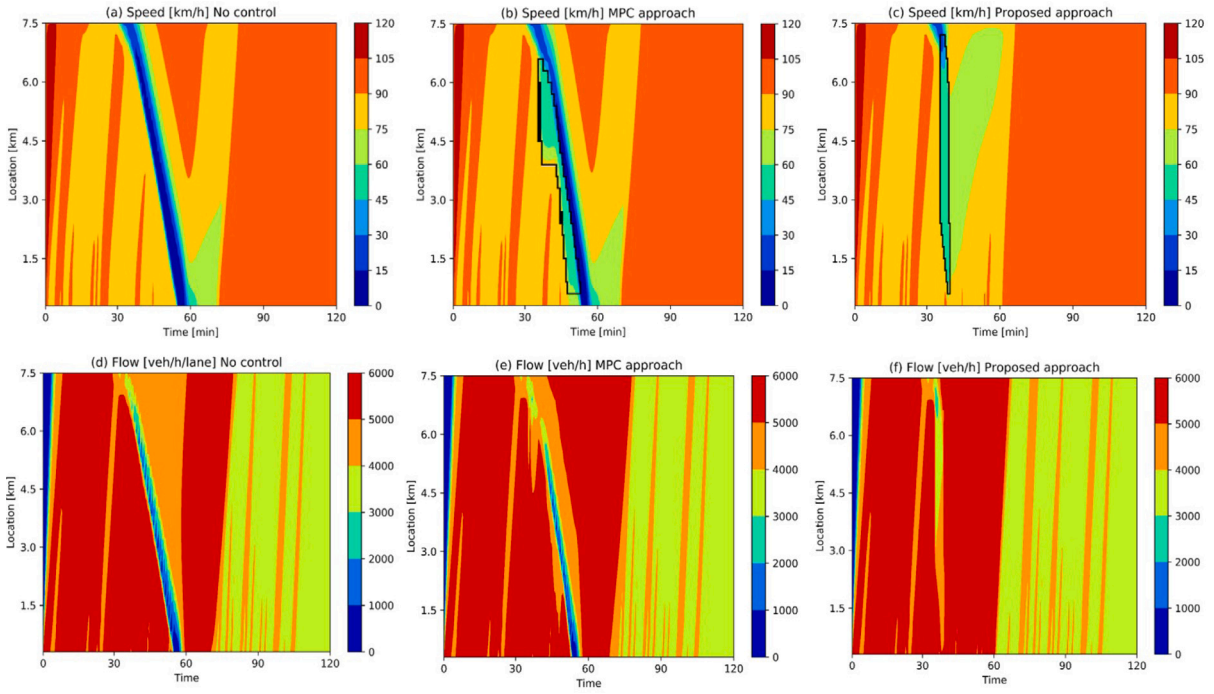

Fig. 12. Comparison between the MPC approach and the proposed VSL control approach in an example. In this example, (a) and (d) are the simulated speed (km/h) and flow (veh/h) contour plots without VSL control; (b) and (e) are the simulation results under the MPC control approach; (c) and (f) are the simulation results under the proposed VSL control approach. In (b) and (c), the VSL-controlled areas are enclosed by black lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The MPC controller is run with the online process model for 100 simulation runs. It reduces the average total travel delay by 25.9%, which is higher than SPECIALIST but lower than the proposed VSL controller. The performance of the MPC controller depends on the accuracy of traffic prediction. It may generate ineffective control schemes if the predicted traffic dynamics are not consistent with the simulated traffic process. Fig. 12 shows an example, in which the MPC controller fails to resolve the jam wave because of inaccurate traffic prediction. In this example, the capacity of the process model is set to 1950 veh/h/lane, which is slightly lower than that of the prediction model, 2000 veh/h/lane. At minute 35, when the MPC controller is activated, the predicted traffic demand is 4900 veh/h but the actual traffic demand is about 5400 veh/h. Therefore, the congestion severity of the jam wave is underestimated by the MPC. As a result, the MPC only narrows the jam wave but it is unable to completely resolve it. Using the same demand profile and parameter values, the proposed VSL control approach can successfully resolve the jam wave, as showing by the speed and flow contour plots in Fig. 12(c) and (f).

In this simulation experiment, the prediction model of the MPC controller is the same as the data generation model in the proposed RL-based VSL control approach. However, the performance of the MPC controller is restricted by the accuracy of the prediction model, as evidenced by the above example. On the other hand, the performance of the proposed VSL control approach is not restricted by the accuracy of that model, because the explored actions produced from the data generation model are evaluated in the online process (i.e., the reality), and the actions that lead to worse traffic performances are discarded.

5.4. The exploration and learning costs

This section compares the proposed approach with an existing RL-based VSL control approach using random exploration in terms of exploration and learning costs. The RL model with random exploration is directly trained in the online simulation environment using the DDQN algorithm. The performance curves of the random exploration approach are shown in Fig. 13. We use data from the first 10 000 simulation runs to evaluate the exploration and learning costs, as the performance of the random exploration approach stabilizes after 10 000 simulation runs. For the proposed approach, data from all the online simulations are used for evaluation.

The exploration cost is represented by the performance in terms of travel delay. During the first 10 000 simulation runs, the average total travel delay for the random exploration approach is 168.5 h. In 32.6% of the online simulation runs, VSLs lead to worse traffic performance, i.e., increase the total travel delay. For the proposed approach, the average travel delay during the training phase is 142.2 h. Only in 17.9% of the simulation runs, VSLs lead to worse traffic performance. Furthermore, for the random exploration approach, the average total travel delay saving after 10 000 simulation runs is 28.1%. For the proposed approach, it achieves a comparable performance only using less than 200 simulation runs, as shown in Fig. 10. Therefore, the exploration cost of the proposed approach is much less than that of the random exploration approach (see Fig. 13).

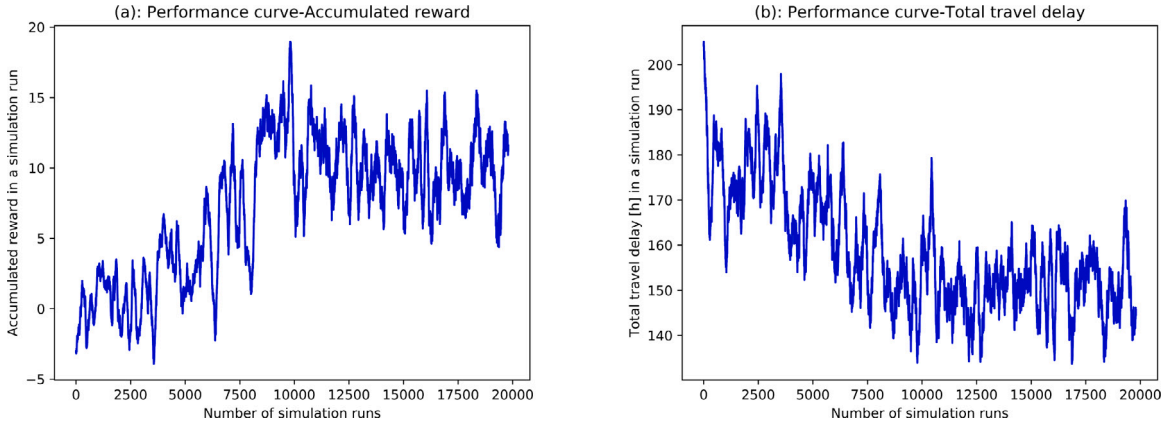

Fig. 13. Performance curves of the random exploration approach in the online training.

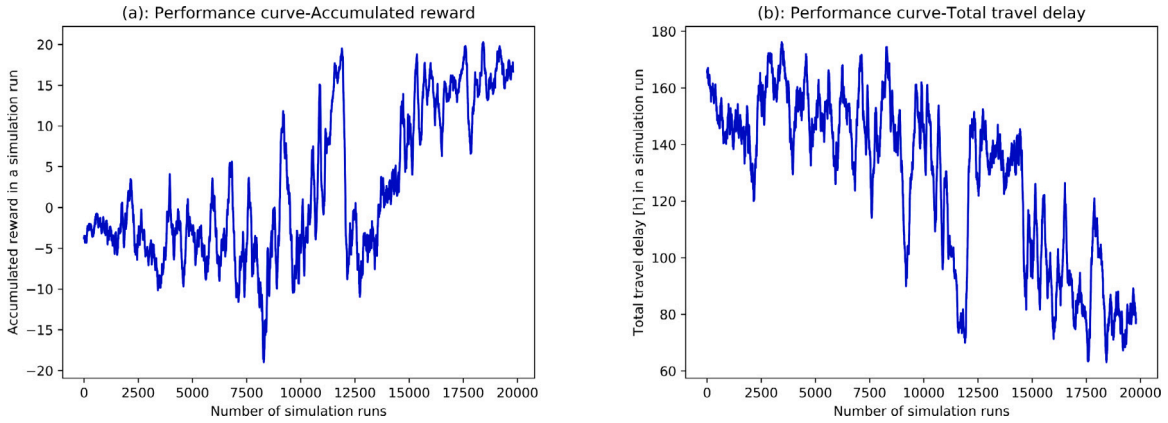

Fig. 14. Performance curves of the training with DDQN in the extended CTM environment.

5.5. Comparison with an existing RL approach

In this section, we compare the proposed approach with an existing RL-based VSL control approach with zero-shot transfer, described as scenario 5 in Section 4. Specifically, the same extended CTM is used as the training environment. An existing deep reinforcement learning algorithm, namely Double DQN (DDQN, Van Hasselt et al. (2016)), is applied as the training algorithm. DDQN has been successfully applied to RL-based traffic signal control systems in multiple studies, such as Zeng et al. (2018), Liang et al. (2019). During the training process, the RL agent receive states and rewards from the environment while the environment implements actions taken by the agents. After training, the optimal policy is directly transferred to the online simulations.

In the training environment, the settings of traffic demand and model parameters are the same as those in Section 4.2. The state, action, and reward are the same as those defined in Sections 2.1 and 4.5. This RL model is trained using data from 20 000 offline simulation runs. Fig. 14 shows the performance curves of the training. The control policy at the end of the training is implemented to the online simulations for 100 runs. It reduces the average total travel delay by 22.4%, which is not as good as the proposed control approach.

Fig. 15 shows an example that highlights the comparison between the proposed approach and the DDQN-based approach. In this example, the proposed approach successfully resolves the jam wave, but the DDQN-based VSL control approach fails. The DDQN-based approach chooses speed limit value 60 km/h at the beginning of VSLs activation, as shown in Fig. 15(b) and (j). As time advances, although the upstream of the VSL-controlled area nearly reaches to the upstream boundary of the freeway stretch, the VSL control still cannot create a transition flow that is sufficiently low to fully resolve the jam, as shown in Fig. 15(e). As a comparison, the proposed approach chooses speed limit value 50 km/h at the beginning of VSLs activation, so the created transition flow is sufficiently low to resolve the jam wave, as shown in Fig. 15(a), (d), and (i).

The performance of the DDQN-based VSL control approach is also tested in the training environment, i.e., the extended CTM, using the same traffic demand of the aforementioned example. In the training environment, the DDQN-based VSL control approach successfully resolves the jam wave and achieves a higher downstream throughput, as shown in Fig. 15(c) and (f). The different performances in the offline training environment and the online simulation indicate that the DDQN-based approach is affected by

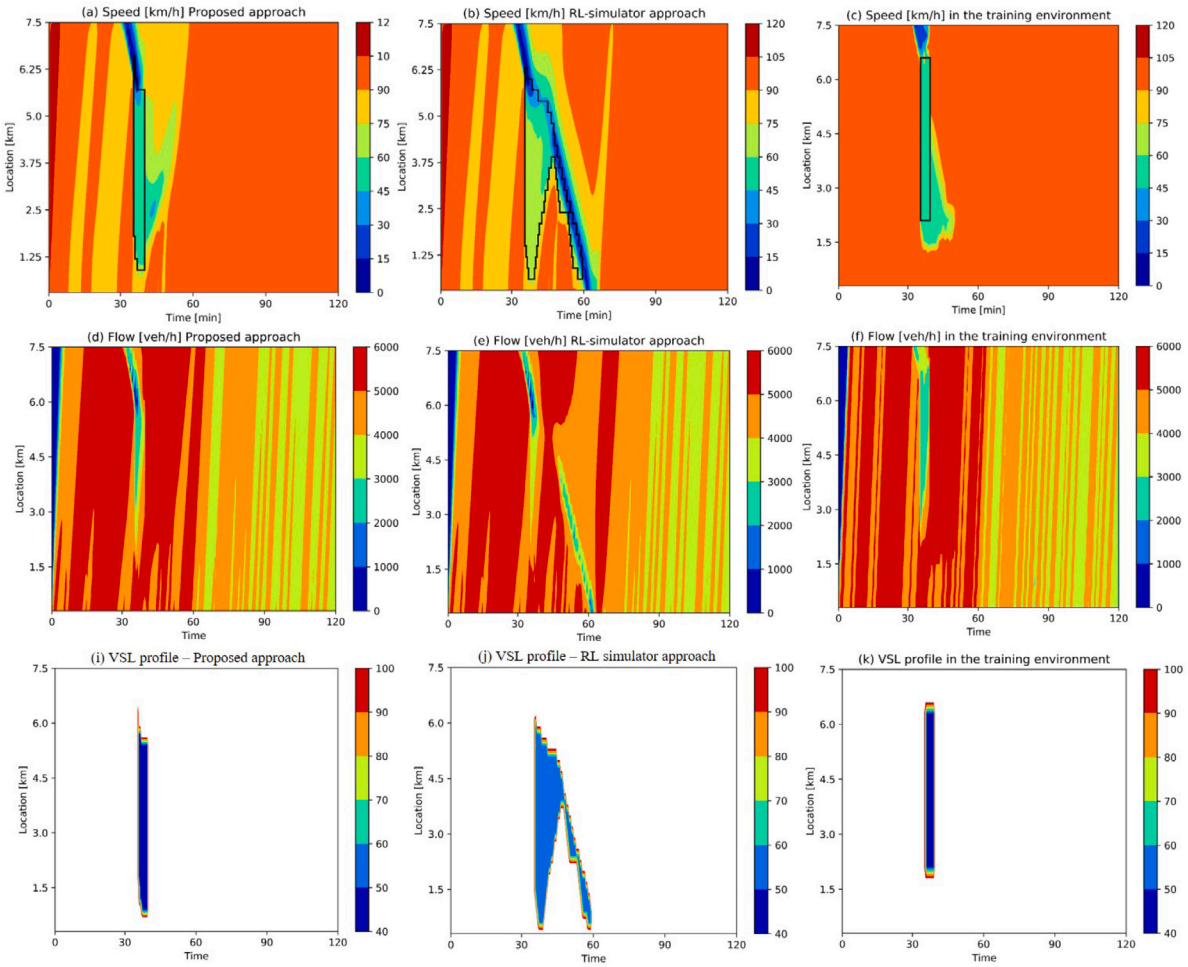

Fig. 15. Comparison between the proposed VSL control approach and the DDQN-based approach in an example. In this example, (a) and (d) are the simulated speed (km/h) and flow (veh/h) contour plots under the proposed approach; (b) and (e) are the simulation results under the DDQN-based approach; (c) and (f) are the simulation results under the DDQN-based approach in the training environment. (i-k) are the corresponding VSLs profiles. In (i), the speed limit is chosen as 50 km/h while in (j), the speed limit is chosen as 60 km/h.

the model mismatch, i.e., the difference between the training environment and the online simulation. Even though a well-trained RL strategy performs well in the training environment, it is not guaranteed that the strategy will perform equally well in real traffic process, where there is always a mismatch.

5.6. DDQN with continual online learning

This section presents the results of scenario 6, the DDQN with continual online learning. Two sub-scenarios are tested in this section. In sub-scenario 1, it is assumed that there is no online exploration after the offline optimal policy being transferred to the online environment. Therefore, the DDQN adopts the greedy policy to update the parameters. In sub-scenario 2, it is assumed that there are still online exploration after the offline optimal policy being transferred. The DDQN adopts the ϵ-greedy policy to update the parameters. The performance curves of both scenarios are shown in Fig. 16.

In those two sub-scenarios, the DDQN with ϵ-greedy policy reduces the total travel delay substantially more than the DDQN with greedy policy. To quantify the learning cost, we use the average delay of the proposed method during the entire training period as a comparison. For the DDQN with ϵ-greedy policy, the average travel delay during the first 2000 simulation runs is 155.9 h, which is 9.6% higher than the average of the proposed method, shown as the red lines in Fig. 16. While the DDQN eventually achieves a similar performance as the proposed method, but at a significantly higher learning cost during the online process.

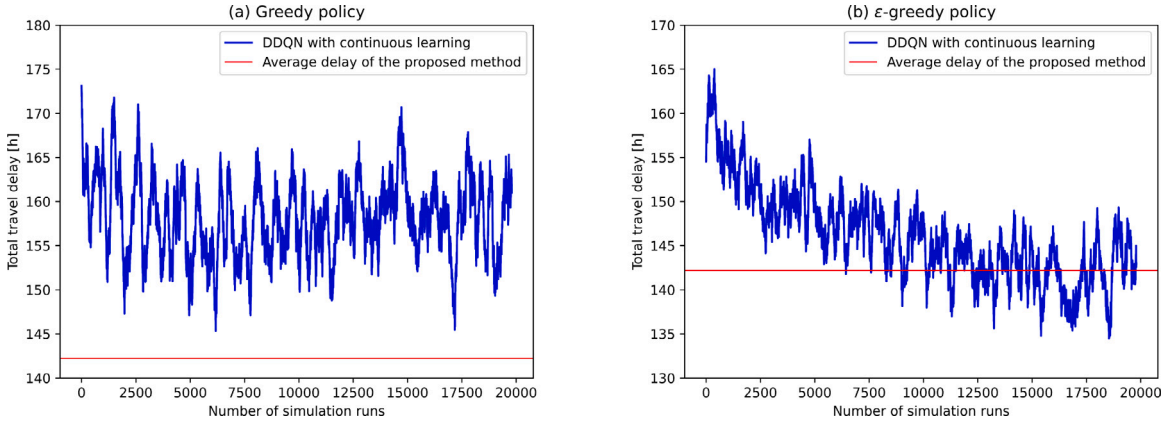

Fig. 16. Performance curves of the DDQN with continual learning.

6. Discussion and conclusions

Reinforcement learning has attracted extensive attentions in traffic control areas. Most of existing RL-based traffic control approaches explore control actions randomly, which may induce high exploration and learning costs. For those approaches, the RL learning cannot be purely based on real-world explorations. Furthermore, the training process with random exploration may require a large amount of training data, which may not be feasible to collect because the speed of data collection in real world is restricted by the "slowness" of the traffic process. Therefore, to date most of existing RL-based traffic control approaches train their RL models solely using traffic simulators. However, The mismatch between the training simulators and the real traffic process affects the performance of those approaches.

In this paper we have proposed a new reinforcement learning-based VSL control approach to resolve freeway jam waves. The proposed VSL control approach applies an iterative training framework, where the optimal control policy is updated by exploring new control actions both online and offline in each iteration. The offline/online exploration method often prevents poor control actions being explored in real traffic process so as to reduce the exploration and learning costs. The explored control actions are evaluated in the real traffic process. Thus the proposed approach avoids letting the RL model learning only from a traffic simulator, and alleviates the impact of the model mismatch by replacing knowledge from the model by knowledge from the real process.

The proposed VSL control approach has been tested using a macroscopic traffic simulation model, namely METANET, which represents real world traffic flow dynamics. The simulation results have shown that the RL controller decreases the total travel delay as more control actions are explored and more training data are fed into the RL. The proposed approach has also been compared with several existing VSL control approaches to demonstrate its advantages. Due to the alleviation of model mismatch errors, the proposed approach performed better in reducing travel delays, than SPECIALIST, the MPC-based approach, and the approach based on an existing RL method. The advantage in reducing the exploration and learning costs has been demonstrated by the comparison with an existing RL-based approach with random exploration.

Although the proposed approach has been demonstrated to alleviate the impact of the model mismatch, it is not guaranteed that it will lead to a system optimal performance. In the proposed method, actions are mainly explored in a smaller space created from the offline model rather than in the entire action space. Therefore, the policy of the RL can be suboptimal if the optimal control actions are out of the exploration space. In future research, we will further investigate if there are better training methods which can incorporate random online explorations and lead to a system optimal performance.

The proposed VSL control approach is designed to resolve freeway jam waves based on the VSL control mechanism against jam waves. In future research, we will extend the proposed approach to eliminate infrastructural bottlenecks such as on-ramp bottleneck and lane-drop bottleneck. The test bed will also be extended to larger sizes of freeway networks. Other methods that can more efficiently deal with the scarcity of real data in RL-based traffic control problems will also be investigated.

CRediT authorship contribution statement

Yu Han: Conceptualization, Methodology, Formal analysis, Software, Writing – original draft. **Andreas Hegyi:** Conceptualization, Methodology, Formal analysis, Writing – review & editing. **Le Zhang:** Conceptualization, Methodology, Writing – review & editing. **Zhengbing He:** Conceptualization. **Edward Chung:** Conceptualization, Writing – review & editing. **Pan Liu:** Conceptualization, Resources, Writing – review & editing.

Acknowledgments

This research is jointly supported by the Natural Science Foundation of Jiangsu (No. BK20200378), and the National Natural Science Foundation of China (No. 52002065, No. 52131203).

Appendix A. METANET model

In the METANET model, the following equations describe the evolution of freeway traffic dynamics over time. The outflow of each cell is equal to the density times the mean speed and the number of lanes of that cell (represented by λ_i):

$$q_i(t) = \rho_i(t) v_i(t) \lambda_i, \tag{A.1}$$

The density of a cell follows the vehicle conservation law, which is represented as:

$$\rho_i(t+1) = \rho_i(t) + \frac{T_s}{l_i \lambda_i} \left(q_{i-1}(t) - q_i(t) \right), \tag{A.2}$$

where l_i is the length of cell i. The mean speed of segment i at time step $t+1$, $v_i(t+1)$, depends on the mean speed at time step t, the speed of the inflow of vehicles, and the density downstream. Specifically,

$$v_i(t+1) = v_i(t) + \frac{T_s}{\tau_M}(V(\rho_i(t)) - v_i(t)) + \frac{T_s}{l_i} v_i(t)(v_{i-1}(t) - v_i(t)) - \frac{\vartheta T_s}{\tau_M l_i} \frac{\rho_i(t+1) - \rho_i(t)}{\rho_i(t) + \kappa}, \tag{A.3}$$

where τ_M, ϑ, κ are model parameters. In the experiment, τ_M is set to 18 s, κ is set to 40 veh/km/lane, and ϑ is set to 30 km^2/h.

Appendix B. SPECIALIST

There are multiple tuning parameters for the SPECIALIST algorithm, which correspond to the traffic states in Fig. 3. The control scheme can be constructed given the measured and calculated traffic states 1–6. The densities, speeds, and flows for the six states are denoted as $\rho_{[j]}, v_{[j]}, q_{[j]}, j \in 1, \ldots, 6$. In the experiments, these parameters are determined using the same method as in Hegyi and Hoogendoorn (2010). One of the most important tuning parameters is the density associated with state 4. The speed of state 4 is determined by the speed limits, however the choice of the density is a design variable that influences the shape of the control scheme. Based on trial-and-error tuning, $\rho_{[4]}$ is set to 30 veh/km/lane, and $\rho_{[5]}$ and $q_{[5]}$ are set to 27 veh/km/lane and 2000 veh/h/lane, respectively.

After the construction of the control scheme, the resolvability is assessed. If the constructed control scheme satisfies certain conditions, the jam wave is considered to be resolvable and the control scheme is applied. These conditions include: (i) the heads and tails of areas 2 and 4 should converge; (ii) the speed of area 6 should be higher than the speed limits; and (iii) the necessary length of the speed-limited stretch is smaller than the available upstream free-flow area. In the experiment, it is assumed that the SPECIALIST can choose one speed limit value from 50 km/h and 60 km/h. If both values satisfy the conditions of resolvability, the higher value 60 km/h will be chosen. The VSL control is activated at minute 35, when the jam wave has already formed.

References

Arel, I., Liu, C., Urbanik, T., Kohls, A., 2010. Reinforcement learning-based multi-agent system for network traffic signal control. IET Intell. Transp. Syst. 4 (2), 128–135.

Belletti, F., Haziza, D., Gomes, G., Bayen, A.M., 2017. Expert level control of ramp metering based on multi-task deep reinforcement learning. IEEE Trans. Intell. Transp. Syst. 19 (4), 1198–1207.

Carlson, R.C., Papamichail, I., Papageorgiou, M., 2011. Local feedback-based mainstream traffic flow control on motorways using variable speed limits. IEEE Trans. Intell. Transp. Syst. 12 (4), 1261–1276.

Carlson, R.C., Papamichail, I., Papageorgiou, M., 2014. Integrated feedback ramp metering and mainstream traffic flow control on motorways using variable speed limits. Transp. Res. C 46, 209–221.

Carlson, R.C., Papamichail, I., Papageorgiou, M., Messmer, A., 2010a. Optimal motorway traffic flow control involving variable speed limits and ramp metering. Transp. Sci. 44 (2), 238–253.

Carlson, R.C., Papamichail, I., Papageorgiou, M., Messmer, A., 2010b. Optimal mainstream traffic flow control of large-scale motorway networks. Transp. Res. C 18 (2), 193–212.

Chen, D., Ahn, S., 2015. Variable speed limit control for severe non-recurrent freeway bottlenecks. Transp. Res. C 51, 210–230.

Chen, D., Ahn, S., Hegyi, A., 2014. Variable speed limit control for steady and oscillatory queues at fixed freeway bottlenecks. Transp. Res. B 70, 340–358.

Davarynejad, M., Hegyi, A., Vrancken, J., van den Berg, J., 2011. Motorway ramp-metering control with queuing consideration using Q-learning. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1652–1658.

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

Frejo, J.R.D., Camacho, E.F., 2012. Global versus local MPC algorithms in freeway traffic control with ramp metering and variable speed limits. IEEE Trans. Intell. Transp. Syst. 13 (4), 1556–1565.

Frejo, J.R.D., Núñez, A., De Schutter, B., Camacho, E.F., 2014. Hybrid model predictive control for freeway traffic using discrete speed limit signals. Transp. Res. C 46, 309–325.

Frejo, J.R.D., Papamichail, I., Papageorgiou, M., De Schutter, B., 2019. Macroscopic modeling of variable speed limits on freeways. Transp. Res. C 100, 15–33.

Hadiuzzaman, M., Qiu, T.Z., 2013. Cell transmission model based variable speed limit control for freeways. Can. J. Civil Eng. 40 (1), 46–56.

Hadiuzzaman, M., Qiu, T.Z., Lu, X.-Y., 2013. Variable speed limit control design for relieving congestion caused by active bottlenecks. J. Transp. Eng. 139 (4), 358–370.

Han, Y., Hegyi, A., Yuan, Y., Hoogendoorn, S., 2017a. Validation of an extended discrete first-order model with variable speed limits. Transp. Res. C 83, 1–17.

Han, Y., Hegyi, A., Yuan, Y., Hoogendoorn, S., Papageorgiou, M., Roncoli, C., 2017b. Resolving freeway jam waves by discrete first-order model-based predictive control of variable speed limits. Transp. Res. C 77, 405–420.

Han, Y., Wang, M., He, Z., Li, Z., Wang, H., Liu, P., 2021. A linear Lagrangian model predictive controller of macro-and micro-variable speed limits to eliminate freeway jam waves. Transp. Res. C 128, 103–121.

Han, Y., Wang, M., Li, L., Roncoli, C., Gao, J., Liu, P., 2022. A physics-informed reinforcement learning-based strategy for local and coordinated ramp metering. Transp. Res. C 137, 103584.
Han, Y., Yuan, Y., Hegyi, A., Hoogendoorn, S.P., 2016. New extended discrete first-order model to reproduce propagation of jam waves. Transp. Res. Record: J. Transp. Res. Board (2560), 108–118.
Hegyi, A., De Schutter, B., Hellendoorn, H., 2005a. Model predictive control for optimal coordination of ramp metering and variable speed limits. Transp. Res. C 13 (3), 185–209.
Hegyi, A., De Schutter, B., Hellendoorn, J., 2005b. Optimal coordination of variable speed limits to suppress shock waves. IEEE Trans. Intell. Transp. Syst. 6 (1), 102–112.
Hegyi, A., Hoogendoorn, S., 2010. Dynamic speed limit control to resolve shock waves on freeways-Field test results of the SPECIALIST algorithm. In: 2010 International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 519–524.
Hegyi, A., Hoogendoorn, S., Schreuder, M., Stoelhorst, H., Viti, F., 2008. SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. In: 2008 International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 827–832.
Kerner, B.S., 2002. Empirical macroscopic features of spatial-temporal traffic patterns at highway bottlenecks. Phys. Rev. E 65 (4), 046138.
Kerner, B.S., Rehborn, H., 1996. Experimental features and characteristics of traffic jams. Phys. Rev. E 53 (2), R1297.
Kotsialos, A., Papageorgiou, M., Diakaki, C., Pavlis, Y., Middelham, F., 2002a. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. IEEE Trans. Intell. Transp. Syst. 3 (4), 282–292.
Kotsialos, A., Papageorgiou, M., Mangeas, M., Haj-Salem, H., 2002b. Coordinated and integrated control of motorway networks via non-linear optimal control. Transp. Res. C 10 (1), 65–84.
Kotsialos, A., Papageorgiou, M., Messmer, A., 1999. Optimal coordinated and integrated motorway network traffic control. In: 14th International Symposium on Transportation and Traffic Theory.
Li, Z., Liu, P., Xu, C., Duan, H., Wang, W., 2017. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. IEEE Trans. Intell. Transp. Syst. 18 (11), 3204–3217.
Li, L., Lv, Y., Wang, F.-Y., 2016. Traffic signal timing via deep reinforcement learning. IEEE/CAA J. Autom. Sin. 3 (3), 247–254.
Liang, X., Du, X., Wang, G., Han, Z., 2019. A deep q learning network for traffic lights' cycle control in vehicular networks. IEEE Trans. Veh. Technol. 68 (2), 1243–1253.
Lighthill, M.J., Whitham, G.B., 1955. On kinematic waves. II. A theory of traffic flow on long crowded roads. In: Proceedings of the Royal Society of London a: Mathematical, Physical and Engineering Sciences. 229, The Royal Society, pp. 317–345.
Lu, X.-Y., Qiu, T.Z., Varaiya, P., Horowitz, R., Shladover, S.E., 2010. Combining variable speed limits with ramp metering for freeway traffic control. In: Proceedings of the 2010 American Control Conference. IEEE, pp. 2266–2271.
Lu, X.-Y., Shladover, S.E., Jawad, I., Jagannathan, R., Phillips, T., 2015. Novel algorithm for variable speed limits and advisories for a freeway corridor with multiple bottlenecks. Transp. Res. Rec. 2489 (1), 86–96.
Messmer, A., Papageorgiou, M., 1990. METANET: A macroscopic simulation program for motorway networks. Traffic Eng. Control 31 (9).
Muralidharan, A., Horowitz, R., 2015. Computationally efficient model predictive control of freeway networks. Transp. Res. C.
Ozan, C., Baskan, O., Haldenbilen, S., Ceylan, H., 2015. A modified reinforcement learning algorithm for solving coordinated signalized networks. Transp. Res. C 54, 40–55.
Papageorgiou, M., 1998. Some remarks on macroscopic traffic flow modelling. Transp. Res. A 32 (5), 323–329.
Papageorgiou, M., Hadj-Salem, H., Blosseville, J.-M., 1991. ALINEA: A local feedback control law for on-ramp metering. Transp. Res. Rec. 1320 (1), 58–67.
Papageorgiou, M., Kosmatopoulos, E., Papamichail, I., 2008. Effects of variable speed limits on motorway traffic flow. Transp. Res. Record: J. Transp. Res. Board (2047), 37–48.
Prashanth, L., Bhatnagar, S., 2010. Reinforcement learning with function approximation for traffic signal control. IEEE Trans. Intell. Transp. Syst. 12 (2), 412–421.
Richards, P.I., 1956. Shock waves on the highway. Oper. Res. 4 (1), 42–51.
Roncoli, C., Papageorgiou, M., Papamichail, I., 2015. Traffic flow optimisation in presence of vehicle automation and communication systems–part II: Optimal control for multi-lane motorways. Transp. Res. C 57, 260–275.
Schmidt-Dumont, T., van Vuuren, J.H., 2015. Decentralised reinforcement learning for ramp metering and variable speed limits on highways. IEEE Trans. Intell. Transp. Syst. 14 (8), 1.
Schönhof, M., Helbing, D., 2007. Empirical features of congested traffic states and their implications for traffic modeling. Transp. Sci. 41 (2), 135–166.
Soriguera, F., Martínez, I., Sala, M., Menéndez, M., 2017. Effects of low speed limits on freeway traffic flow. Transp. Res. C 77, 257–274.
Spiliopoulou, A., Kontorinaki, M., Papageorgiou, M., Kopelias, P., 2014. Macroscopic traffic flow model validation at congested freeway off-ramp areas. Transp. Res. C 41, 18–29.
Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.
Van Hasselt, H., Guez, A., Silver, D., 2016. Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, No. 1.
Wang, Y., Yu, X., Zhang, S., Zheng, P., Guo, J., Zhang, L., Hu, S., Cheng, S., Wei, H., 2020. Freeway traffic control in presence of capacity drop. IEEE Trans. Intell. Transp. Syst. 22 (3), 1497–1516.
Watkins, C.J., Dayan, P., 1992. Q-learning. Mach. Learn. 8 (3–4), 279–292.
Wu, Y., Tan, H., Qin, L., Ran, B., 2020. Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm. Transp. Res. C 117, 102649.
Yuan, K., Knoop, V.L., Hoogendoorn, S.P., 2015. Capacity drop: Relationship between speed in congestion and the queue discharge rate. Transp. Res. Rec. 2491 (1), 72–80.
Zeng, J., Hu, J., Zhang, Y., 2018. Adaptive traffic signal control with deep recurrent Q-learning. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1215–1220.
Zhang, Y., Ioannou, P.A., 2016. Combined variable speed limit and lane change control for highway traffic. IEEE Trans. Intell. Transp. Syst. 18 (7), 1812–1823.
Zhang, Y., Ioannou, P.A., 2018. Stability analysis and variable speed limit control of a traffic flow model. Transp. Res. B 118, 31–65.