Integrated Traffic Control for Freeway Recurrent Bottleneck Based on Deep Reinforcement Learning

Chong Wang, *Member, IEEE*, Yang Xu, Jian Zhang, *Member, IEEE*, and Bin Ran

***Abstract*— Recent advances in deep reinforcement learning have shown promising results in solving sophisticated control problems with high dimensional states and action space. Inspired by this, we use the latest deep reinforcement learning (DRL) methods to improve freeway traffic mobility and alleviate recurring bottlenecks and congestion. More specifically, this paper proposes a centralized traffic control system that can coordinate multiple ramp metering (RM) and variable speed limit (VSL) traffic controllers on freeways to minimize the total travel time. The system uses a novel double-layer structure to synchronize different traffic controllers and introduces the actor-critic-based DRL methods to learn joint actions in a high-dimensional traffic environment. The reward function takes into account the waiting time of vehicles, the average speed of different road sections, and the on-ramp queuing limit to improve traffic mobility. We also proposed an integrated feedback controller as a benchmark. The simulation results show that the actor-critic-based methods are superior to other methods and can save more than 20% of the total travel time. We also analyzed the curse of dimensionality problem by comparing the performance of two scenarios in the simulation: one is a single-ramp interweaving area scenario; the other is a large freeway corridor with multiple on-ramps and off-ramps. The results show that our system can effectively handle these two situations without significant performance degradation, which means that the centralized control system can effectively control freeway corridors by directly guiding various traffic controllers. This also leads to the conclusion that we can use a centralized actor-critic-based control unit to manage medium-scale freeway traffic to save computing resources instead of using complex collaboration strategies.**

***Index Terms*— Integrated traffic control, deep reinforcement learning, deep actor-critic algorithm, freeway traffic management.**

I. Introduction

THE freeway bottleneck occurs when the traffic demand exceeds traffic capacity, resulting in capacity drop and traffic congestion. As a rule of thumb, the bottleneck often forms near the on-ramp sections, which is the merge area of the freeway. When a large volume of traffic comes from different directions, the on-ramp area is prone to traffic jams and accidents. Thus, many studies focus on relieving traffic congestion and improving traffic mobility around on-ramp sections. The most frequently discussed strategies are ramp metering (RM) [1] and the variable speed limit (VSL) [2] control. The RM restricts vehicles from entering the congested area, while the VSL limits the upstream speed to reduce the inflow traffic. The restrictions will be lifted after congestion is relieved. Although consider either RM or VSL may have good performance in some cases, there are some imperfections. For example, the RM may disturb the nearby road traffic when the on-ramp road is full of vehicles, while the speed limit may adversely affect the upstream traffic. Therefore, the integration of RM and VSL has practical value, as it can combine the merits of both strategies and reduce the side effect of the single control strategy.

Conventionally, two ways are proposed to integrate RM and VSL control near the on-ramp segment. First is the model predictive control (MPC) methods [3]–[5], second is the feedback (e.g., Proportion Integral) control methods [6], [7]. The MPC control methods can optimize traffic proactively and systematically. Meanwhile, the feedback control methods are easier for implementation and are reliable when the traffic fluctuates within a reasonable range. However, both of them have limitations. For example, complicated traffic models restrict the problem scale of MPC methods. Besides, real traffic's randomness leads to inevitable inconsistency between the model and reality, making long-term traffic unpredictable. On the other hand, the feedback methods have delayed responding to the control objectives, which may decay its performance when the traffic is heavy and rapid variation. Therefore, it is necessary to seek new solutions to these problems.

Recently, reinforcement learning (RL) based artificial intelligence (A.I.) defeated top human professionals in complicated multiplayer games [8], [9]. They are remarkable progress and demonstrate that the RL methods may have great potential. The traffic environment is similar to the game environment. Thus, the RL approaches can be transferred to intelligent traffic control systems. Rezaee *et al.* [10] studied the ramp metering with the classic RL approaches for a typical bottleneck near Toronto and found that the RL approaches could save 20% more travel time than the feedback approach. Li *et al.* [11] built a VSL control strategy with novel RL technology to relieve the "capacity drop" problem of the recurrent bottlenecks, and they improved the RL exploration techniques. The result showed

Manuscript received 7 August 2020; revised 30 June 2021; accepted 14 December 2021. Date of publication 19 January 2022; date of current version 12 September 2022. This work was supported in part by the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology (NUIST), and in part by the Key Research and Development Program of Jiangsu Province in China under Grant BE2020013. The Associate Editor for this article was J. Sanchez-Medina. *(Corresponding author: Chong Wang.)*

Chong Wang and Yang Xu are with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: chongwang@nuist.edu.cn; yangxu@nuist.edu.cn).

Jian Zhang is with the School of Engineering, Tibet University, Lhasa, Tibet 850011, China, and also with the School of Transportation, Southeast University, Nanjing, Jiangsu 210096, China (e-mail: jianzhang@seu.edu.cn).

Bin Ran is with the College of Engineering, University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: bran@engr.wisc.edu).

Digital Object Identifier 10.1109/TITS.2022.3141730

that the RL approach could quickly adapt to the changing environment and even drivers' behavior. Zhu *et al.* [12] proposed a novel dynamic speed limit RL control model in the stochastic traffic network environment. They found that the total travel time and emissions could reduce by near 20% compared with the no control cases. El-Tantawy *et al.* [13] developed an RL-based coordination control strategy for multiple intersection traffic lights in the large-scale city road network. The result showed that the average intersections delay could reduce by 39%, and the road travel time could reduce by 27%. In summary, RL algorithms have two advantages compared with the feedback and MPC methods: (1) they can control traffic flexibly with self-adaption and act proactively based on the historical experience. (2) They are model-free and do not require additional expert experience to train but to learn by themselves automatically. As a result, the number of RL studies for intelligent traffic control has proliferated in recent years [16].

The classic RL methods often suffer from the curse of dimensionality when facing large-scale traffic control problems. Therefore, researchers introduced deep reinforcement learning (DRL) methods to solve the problem. For freeway traffic control, Bellteti *et al.* [17] proposed the multi-agent share weight DRL method for the cooperative ramp metering control. They found that the DRL method could achieve performance similar to the ALINEA algorithm and save 20% travel time. Wu *et al.* [18] studied the differential variable speed limits control (DVSL) with deep reinforcement learning. The results indicated that the DRL-based DVSL control strategy could improve safety, efficiency, and vehicle emissions with different rewards. However, unlike the amount of urban traffic light control [16] studies, research on the freeway traffic control with DRL is still insufficient, especially for integration control (of RM and VSL controllers). Integration models require inspection on coordination details, and some challenges have to be considered first, which are listed as follows:

(1) How to decide the control period for a traffic controller: for example, a traffic controller refers to a VSL sign or an RM traffic signal light. In normal circumstances, the RM's control cycle (signal phase) could not keep pace with the VSL controller. The immediate solution is to make independent agents with different control cycles [7], but this may lead to a complicated joint control strategy.

(2) How to control multiple traffic controllers accurately and efficiently: Although the multi-agent DRL algorithms developed quickly in recent years [19], they are still quite complicated, and only some could transplant to the traffic control field. Therefore, it is necessary to find new ways for integration control. The ability of actor-critic-based DRL methods to control multiple traffic controllers is neglected. Their structures are straightforward and do not require complex collaboration.

(3) How to process the massive input traffic data: images or snapshots are used as the state input in previous studies [20], [21]. It worked fine on intersections but may not work fine as well for freeway traffic control. Besides, it requires vast storage resources to record every vehicle's data.

We proposed a novel integrated control system with the actor-critic-based DRL algorithms to solve the above challenges for freeway traffic control. The system allows dynamic speed limits on the upstream of the bottleneck area and adaptive ramp metering control to adjust the outflow of ramp roads. To the best of our knowledge, this is the first study to integrate multiple VSL and RM controllers with the latest DRL methods for freeway traffic management. The main contributions of this paper can be summarized as follows:

(1) This paper unified the control cycle with a double-layer structure to synchronize different traffic controllers. The upper layer optimizes the ramp outflow per minute to keep pace with the VSL controllers. The lower layer converts ramp outflow into RM signal phases.

(2) This paper proposed a generalized framework comparable to many actor-critic-based DRL algorithms. The framework is very flexible towards extensions. Therefore, our system is not plagued by the curse of dimensionality, and the performance is not noticeably decayed in a large road network.

(3) This paper divided the road network into segments according to the Courant-Friedrichs-Lewy (CFL) condition [22]. Besides, it merged diverse traffic data (e.g., average speed, density, traffic flow, etc.) to obtain traffic states. In addition, we introduced a normalization layer to merge different traffic data. The neural networks are much simpler without the image processing layers.

The rest of this paper is organized as follows: The review of related works is in Section II. The methodology is described in Section III. The problem statement is given in Section IV. The DRL model is proposed in section V. The control framework and improved algorithms are given in section VI. The simulation results are shown in Section VII. The conclusion is given in Section VIII.

II. Related Work

This section will discuss the DRL algorithms studied in previous works and the Vehicle to Infrastructure (V2I) technology for the DRL control system.

A. Deep Reinforcement Learning for Traffic Control

The essence of reinforcement learning is to learn through interaction with the environment and record the experiences (e.g., states, actions, rewards, etc.) to continuously improve the agent's behavior. The classic RL (e.g., Q learning) methods use the Q table to record experiences, which suffers from the curse of dimensionality for large-scale traffic control. As a result, single-agent RL methods can only solve local control problems [10], [11], for large-scale problems must use multi-agent reinforcement learning (MARL) with complicated coordination strategies [13]–[15].

Unlike the classic RL methods, the DRL methods use neural networks to record learning experiences [23] and have various optimizations for further improvement [24]. Since the nonlinear neural network better describes real-world traffic, the DRL methods have become prevailing in urban traffic control research [25]. Study [20], [26] built the policy gradient (PG) and deep Q network (DQN) models for the traffic light control

TABLE I
SUMMARY OF REPRESENTATIVE DEEP REINFORCEMENT LEARNING STUDIES IN TRAFFIC CONTROL

Reference	DRL algorithm	State	Reward	Action	Simulation Result
F. Belletti et al. [17]	shared weight PPO/TRPO	Density Queue length	Total outflow	Multiple RM	Reduce 20% total vehicle hour during congestion
Y. Wu et al. [18]	DDPG	Occupancy	Flow, Speed Emissions	DVSL	Reduce 11% travel time and 6.5% CO_2 emissions
Mousavi, et al. [20]	Policy-Gradient/DQN	Queue length	Wait time	Signal phase	Reduced near two-thirds of intersection wait time
T. Wu et al. [21]	MADDPG with LSTM	Position, Speed Phase	Wait time Queue length	Signal phase	Pedestrian and vehicle waiting time reduced by 18% and 48%, respectively
X. Liang et al. [26]	3DQN with prioritized replay	Position Speed	Wait time Queue length	Signal phase	Reduced 20% more intersection wait time compared with baseline
Y. Gong et al. [27]	3DQN with Convolutional Layers	Queue length Position	Wait time	Signal phase	Reduce 10% travel time and 46% total delay
T. Chu et al. [27]	A2C with LSTM Layers	Queue length Inflow	Wait time Queue length	Signal phase	Average 20.8% travel delay and 65% intersection delay, increase 12% speed

TABLE II
MEANING AND REFERENCE OF ACRONYMS

Acronym	Meaning
RL/DRL	(Deep) reinforcement learning
DQN	Deep Q Network [20]
3DQN	Dueling Double Deep Q Network [26]
A2C	advantage actor-critic [28]
LSTM	long short term memory [28]
MADDPG	multi-agent deep deterministic policy-gradient [21]
PPO	Penalized Policy Optimization [17]
TRPO	Trust Region Policy Optimization [17]
DDPG	deep deterministic policy-gradient [18]
RM	Ramp metering [17]
DVSL	Differential variable speed limit [18]

(TLC) of independent intersections. The simulation results showed that the intersection waiting time could be visibly reduced. Study [27], [28] discussed multi-agent DRL control strategies for large-scale urban traffic control, proposed different cooperative solutions including traffic state sharing [27], advantage actor-critic (A2C) algorithm with long short term memory [28] (LSTM), and multi-agent deep deterministic policy-gradient (MADDPG) [21] algorithm. The representative studies are summarized in Table I.

The meaning of acronyms in Table I is given in Table II.

From TABLE I, we can see three limitations in previous works. Firstly, most current studies focus on urban traffic control, while research on freeway traffic is insufficient. Secondly, current multi-agent systems have many control units, which might not be necessary. Thirdly, DRL methods develop fast, many new algorithms [29], [30] not studied in the traffic control field may play an essential part in the future. In this paper, we make the following improvements: Firstly, we designed a centralized traffic control system to combine the merits of different control strategies. Secondly, we proposed a generalized framework upon the novel actor-critic structure to handle high-dimensional control problems effectively. Thirdly, we employed the latest actor-critic DRL algorithms with improved techniques to achieve good performance.

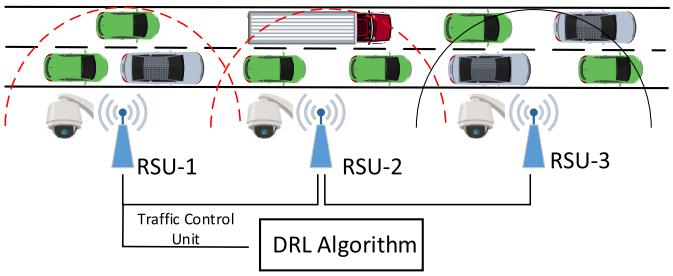


Fig. 1. Conceptual design of V2I based freeway traffic control system.

B. Vehicle to Infrastructure for Deep Reinforcement Learning

V2I is critical for an intelligent transportation system. With V2I technology, the road can collect traffic information more efficiently, and the vehicles can respond to road instructions more quickly. In previous papers, V2I plays a key role in improving the performance of MPC methods [31], [32] and the cooperative on-ramp merging models [33], [34]. Study [35] pointed out that the control effect can be significantly improved under the V2I environment.

Although our control system is implemented via the SUMO simulation environment, we could consider how to realize it in the real world. V2I is the bridge from simulation to reality. The traffic state can be collected through various detectors in the simulation, and then the instructions are immediately sent to the vehicles. Similar functions can be done via the V2I technology. Fig. 1 proposed the conceptual design of a freeway traffic control system with V2I. The system consists of roadside units (RSU), wireless transceivers, and a traffic control unit (TCU). RSU and transceiver are the perceptrons of the system, and the TCU is the brain. The freeway is divided into segments, and each segment has an RSU. The RSU equips devices (e.g., traffic detector, camera, etc.) to collect vehicle data of the segment. Then, the vehicle data is aggregated and sent to the TCU via the transceiver. Next, the TCU uses the DRL algorithms to calculate the optimal results as the control instructions. Finally, the control instructions are sent to the vehicles through the transceivers, and the vehicles can adjust their behaviors according to the instructions.

Based on the above discussion, we assume that in the simulation scenario, the traffic control system can access the freeway traffic state, and the vehicles will obey the instructions of the control system. The detailed design of the control system is given in sections V and VI.

III. Methodology

A. Reinforcement Learning

The traffic control problem can be treated as a Markov Decision Process (MDP) for the RL agents. The RL agents learn by interacting with the dynamic environment. Each step, the agents perceive the environment and take some actions. Then the environment transits into a new state, and a reward is generated. The process can be described with a five-tuple (S, A, P, R, γ), where:

S : denotes the traffic state space. $s \in S$ is a specific state;

A : denotes the action space. $a \in A$ is a specific action;

$P = S \times A \times S$: denotes the transmission probability among the space sequences, the relation of state s and subsequent state s' is characterized as $s' = P^a_{ss'}$;

R : denotes the reward space. $r \in R$ is the immediate reward of an action;

$\gamma \in [0, 1)$: denotes the discount factor that defines the relative importance of the immediate and historical rewards.

To gradually improve the environment to the ideal state, RL agents select actions according to an optimal policy π. The goal of policy π is to maximize the cumulative expected rewards starting from the initial state. If agents know the optimal cumulative reward of a particular state, they can choose actions with the highest reward [26]. The cumulative reward can be obtained recursively with the Bellman equation. For instance, the agent at certain state s takes an action a to reach state s' and gets a reward r, which is denoted by tuple (s, a, r, s'), then the cumulative reward $Q^\pi(s, a)$of policy π can be calculated by the following equation

$$Q^\pi(s, a) = E_{s'}\left[r + \gamma \max_{a'} Q^\pi\left(s', a'\right) | s, a\right] \tag{1}$$

where $a' \sim \pi\left(s'\right)$ is the action selected according to the policy π at state s', $E_{s'} = \sum_{s' \in S} p^a_{ss'}$ is the expectation of states that could transfer from (s, a), γ is the discount factor, and is the best possible action.

Conventionally, $Q^\pi(s, a)$ can be updated via a Q table [10]–[15]. When the number of states becomes large, the Q table could consume a considerable amount of computing resources. Hence, DQN [23] methods are proposed to approximate the $Q^\pi(s, a)$ with a neural network $Q_\phi(s, a)$. However, DQN methods have difficulties processing large action space or continuous actions [35]. Therefore, the deep actor-critic algorithm emerges as a new solution [37].

B. Deep Actor-Critic Algorithm

Similar to the classic reinforcement learning, the objective of the deep actor-critic algorithm is to find a parametric policy π_θ to maximize the expected return $J(\pi_\theta)$. The policy π_θ is a neural network with parameters θ known as the actor. It can be updated by taking the gradient of the expected return $\nabla_\theta J(\pi_\theta)$ [28], which can be written as:

$$\nabla_\theta J(\pi_\theta) = E_{s \sim p^\pi}\left[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s, a)\right], \quad a \sim \pi_\theta(s) \tag{2}$$

where $E_{s \sim p^\pi}$ is the expected possible states decided by the transmission probability, $\log \pi_\theta(a|s)$ is the probability of taking action a under state s following policy π_θ, $Q^\pi(s, a)$is the expected return of state-action pair (s, a) following policy π. Eq. (2) increases the probability of the actions with high $Q^\pi(s, a)$ values and decreases others vice versa. Note that Eq. (2) could be more straightforward if updated following the chain rule in the deterministic form [35] as:

$$\nabla_\theta J(\pi_\theta) = E_{s \sim p^\pi}\left[\nabla_a Q^\pi(s, a) \nabla_\theta \pi_\theta(s)\right], \quad a = \pi_\theta(s) \tag{3}$$

Action a of Eq. (3) is determined by the state s and policy π_θ. $Q^\pi(s, a)$ is known as the critic or value function. For large state space, $Q^\pi(s, a)$ can be estimated with a neural network approximator $Q_\phi(s, a)$ with parameters ϕ [37]. To make the training process more stable, target networks $\pi_{\theta'}$ and $Q_{\phi'}$ are introduced [35]. The estimation equation can be written as:

$$\min_\phi E_{s \sim p} \pi\left(r + \gamma Q_{\phi'}\left(s', a'\right) - Q_\phi(s, a)\right)^2, \quad a' = \pi_{\theta'}(s') \tag{4}$$

where r is the immediate reward, γ is the discount factor, $Q^\pi(s, a) = r + \gamma Q_{\phi'}\left(s', a'\right)$ is the value function. $E_{s \sim p^\pi}$ is the expectation of state occurrence rate under policy π. The intuition form of Eq. (4) is to minimize the difference between $Q_\phi(s, a)$ and $Q^\pi(s, a)$ during training. Training can be applied off-policy, sampling random mini-batches of experiences from an experience replay buffer [29].

Eq. (3) and Eq. (4) constitute the general form of deep actor-critic algorithms. In general, the policy π_θ could be updated by the gradient of the expected return $\nabla_\theta J(\pi_\theta)$, rather than the random selection policy of classic RL methods. This can save many computing resources and is the secret of the actor-critic algorithms to deal with the high dimensional action space. Since our study cases are integrated traffic control, the action space is ample, and it is necessary to use actor-critic architecture. Note that there are various actor-critic algorithms [37]; for convenience, the deep deterministic actor-critic algorithms are chosen as the brain of the control system.

C. Integrated Feedback Control Strategy

The feedback traffic control strategies (e.g., ALINEA [38]) are proved effective by many studies. Thus, we proposed an integrated feedback control strategy modified from the study [7] as the benchmark. The framework is shown in Fig. 2, which coordinates the RM and VSL controllers. The RM controller is a proportional-integral controller that can be written as:

$$r_k = r_{k-1} + (K_I + K_P) e_{o,k} - K_P e_{o,k-1} \tag{5}$$

where r_k is the ramp metering rate based on the desired occupancy at time step k, K_I and K_P are the controller

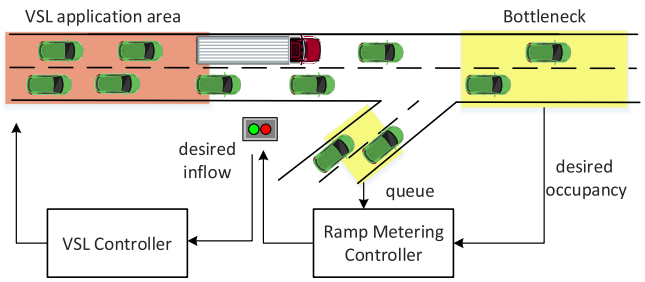


Fig. 2. The framework of the integrated feedback control strategy.

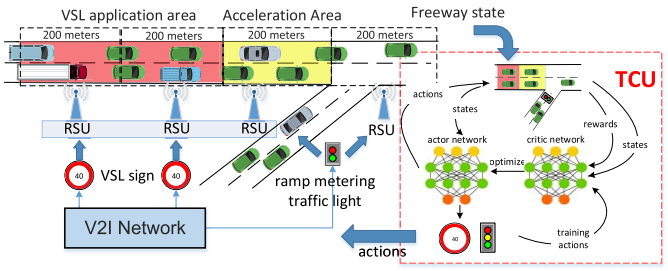


Fig. 3. Integration control system with the deep actor-critic-based TCU.

parameters, $e_{o,k} = \hat{o} - o_k$ is the bias between the desired occupancy $\hat{o}$ and the dynamic occupancy o_k. The intuition form of Eq. (5) is to adjust the ramp metering rate according to the congestion of the bottleneck area. The on-ramp vehicle queue also restricts the ramp metering rate. If the queue is too long, it will spill to the adjacent road. Hence, it is necessary to maintain a minimum ramp metering rate according to Eq. (6):

$$q_k = -\frac{1}{T_{RM}}\left[\hat{\mathrm{w}} - \mathrm{w}_{k-1}\right] + d_{k-1} \tag{6}$$

where $\hat{\mathrm{w}}$ represents the desired queuing limit. w_{k-1} represents the dynamic on-ramp waiting vehicles, and d_{k-1} represents the on-ramp traffic demand, respectively. The ramp metering rate $\tilde{q}_{RM}$ is the maximum value of Eq. (5) and Eq. (6):

$$\tilde{q}_{RM} = \max\left(r_k, q_k\right) \tag{7}$$

When $\tilde{q}_{RM} \geq r_k$, it means that ramp metering has to tackle the over-length queue, hence the VSL controller is activated to limit the upstream inflow from entering the bottleneck. The desired mainline inflow $\hat{q}_{in}$ is given as:

$$\hat{q}_{in} = q_{Cap} - \tilde{q}_{RM} \tag{8}$$

where q_{Cap} is the road capacity. The control objective is to restrict $\hat{q}_{in}$ via an integral controller:

$$b_k = b_{k-1} + K_I e_{q,k} \tag{9}$$

where $b_k \in [b_{min}, 1]$ is the upstream inflow rate. Similarly, $e_{q,k} = \hat{q}_{in} - q_{in,k}$ is the desired flow bias. b_k can be converted into the speed limits with a linear approximator. In this paper, $K_I = 15$, $K_p = 40$, $\hat{o} = 30\%$, $\hat{\mathrm{w}} = 15$ vehicles.

IV. Problem Statement

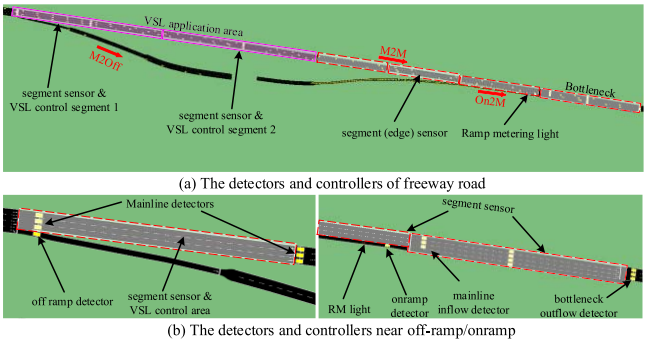
Our goal is to optimize the freeway bottleneck traffic via integrating the RM and VSL controllers. For convenience, the freeway road is divided into a series of segments, and the segment length should satisfy $\Delta x \geq v_{seg}\Delta t$ according to the CFL condition [22], where Δx is the segment length, v_{seg} is the average speed of the segment, and Δt is the time step. In the V2I environment, Δt can be updated every 5 seconds, and the free flow speed is 100 km/h. Hence, we set the segment length to 200 meters. A typical example is given in Fig. 3, the control system combines several RSU, VSL, and RM traffic controllers. The RSUs are deployed along the road, collecting traffic information via different sensors. The VSL controllers are built upstream of the bottleneck, sending the speed limitations to the vehicles. Thus, the upstream traffic is slowed down in the VSL application area when there is congestion downstream. Then the vehicles could accelerate in the acceleration area to pass through the bottleneck more effectively. The RM signal has green and red phases, when the traffic is congested near the on-ramp section, the red phase stops the vehicles from entering the congested area, while the green phase ensures that the vehicles do not spill back to the adjacent road. The TCU is the brain of the system and commands all the VSL and RM traffic controllers.

The first problem is how to improve the freeway traffic mobility by dynamically changing the speed limits of VSL signs and the phase's duration of RM traffic lights via learning from historical experiences. This paper builds a novel actor-critic DRL framework to estimate the Q-value and select action, respectively. First, we need to synthesize the global states by data from different sensors and set up the reward to minimize the total travel time of the entire road network. Then the critic network can self-training by continuously receiving states and rewards from the environment. The model is demonstrated on the right side of Fig. 3. The critic network self-updates after a batch of state data are received. Note that the training can be done offline, which means the critic network can use historical data for pre-training.

Another problem is how to synchronize all the VSL and RM controllers. To keep pace with the speed limits, RM actions are set to the ramp outflow per minute and then transferred into RM signal phase duration, and then the actor (network) could treat the outflow and the speed limits as numeric actions. The actor can choose multiple actions based on the current state and the score (Q-value) of the critic (network). The critic can continuously optimize the actor according to its latest training results to control freeway traffic adaptively. Finally, the actions are sent to traffic controllers through the V2I network. In practice, large-scale road networks can combine several small networks with similar control systems. We have tested two scenarios in the simulation: a typical bottleneck network similar to Fig. 3 and a more complicated large-scale network.

V. Deep Reinforcement Learning Model

In this section, the details of the integrated control system are proposed. We first build the study scenarios and then define our DRL model's three essential elements: states, actions, and



(a) The detectors and controllers of freeway road

(b) The detectors and controllers near off-ramp/onramp

Fig. 4. The detector and controllers of scene one.

rewards. The improved deep actor-critic algorithms are in the next section.

A. The Study Scenes of the Control System

This study aims to evaluate deep reinforcement learning algorithms in integration traffic control. To test control performance, we select two scenarios with ramp weaving sections. The open-source software Simulation of Urban Mobility (SUMO) [39] is chosen for the experiments. SUMO can perfectly simulate different freeway traffic scenes and various control strategies with the Traffic Control Interface package. We made the simulation environment as accurate as possible to evaluate the DRL models.

1) The Single Bottleneck Scene: A 2km freeway with on and off-ramps near Drechttunnel, Netherlands, is selected as the single bottleneck scene, as shown in Fig. 4. The road network is divided into segments, with 200 meters each. The control system has one RM controller and two continuously placed VSL controllers. The traveling traffic has three routes: the mainline to mainline (M2M) traffic, the mainline to off-ramp (M2Off) traffic, and the on-ramp to mainline (On2M) traffic. Krauss model [40] is used as the microscopic model for the simulation, which guarantees safe driving. There are two types of vehicles: 90% of cars and 10% of trucks. The length of a car is 4 meters, while the length of a truck is 8 meters. One simulation episode takes two hours to cover the entire rush hour period. The traffic demands of the three routes are in Table III.

As shown in Fig. 4, there are two types of sensors in the simulation to collect traffic information. The first type is the traditional detectors to collect fixed point data (e.g., traffic flow). The second type is the area sensors that collect traffic data of the entire segment (e.g., density and average velocity). Sufficient sensors are deployed to simulate the V2I environment. In addition, two VSL controllers to command vehicles' speed are placed upstream of the bottleneck (purple area), and an RM controller is placed at the front of the on-ramp road. The on-ramp weaving segment is the bottleneck of the network.

2) The Multiple Bottlenecks Scene: A 6.5km freeway network with multiple on and off-ramps from the Alicante to Murcia freeway open data source [41] is selected as the large-scale scene; the details are shown in Fig. 5. The road network

TABLE III
TRAFFIC DEMAND (VEHICLES PER ROUTE) OF SCENE ONE

Route \ Time	7:00 am-7:30 am	7:30 am-8:30 am	8:30 am-9:00 am
M2M	4026	6596	3023
On2M	1009	1331	1005
M to Off2	823	1200	795

TABLE IV
TRAFFIC DEMAND (VEHICLES PER ROUTE) OF SCENE TWO

Route \ Time	7:00 am-7:30 am	7:30 am-8:30 am	8:30 am-9:00 am
M to Off1	605	1031	823
M to M	1501	2060	1000
M to Off2	523	1105	538
On1 to M	800	501	1200
On2 to M	500	510	505
On2 to Off2	531	500	585
On3 to M	1065	2533	1532

contains two bottleneck areas, about 3.5km apart. The road network is also divided into a series of 200 meters segments. The control system has three RM signal controllers, and four VSL controllers placed in two continuous VSL control segments. The seven traveling routes of traffic are given in Table IV.

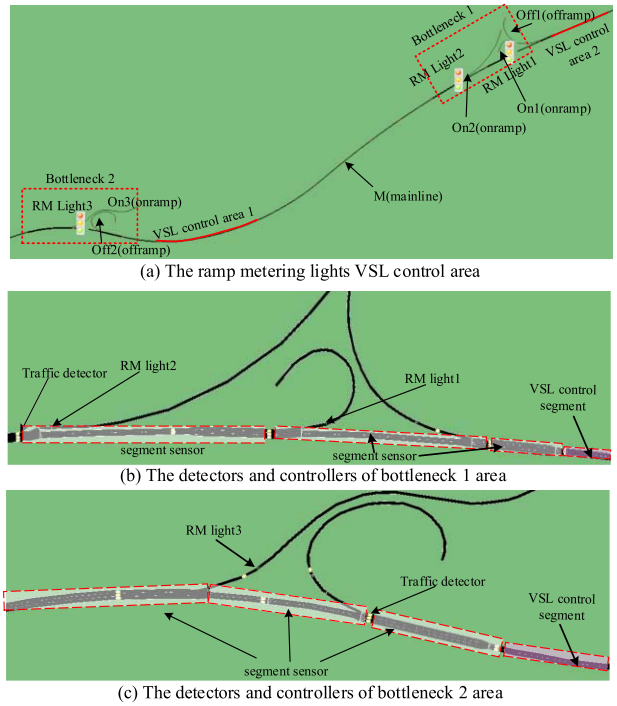
Each segment has a five-tuple state s_i, each traffic controller has an action a_i in the traffic control unit. In the first scene, we have 58 states (including the elements of s_i) and three actions (1 RM and 2 VSL actions). In the second scene, we have 99 states and seven actions (3 RM and 4 VSL actions).

B. The State of the DRL Model

States are the input of the actor-critic DRL model. As mentioned above, the freeway network can be divided into N segments, state s_i represents segment i, then the global state $\boldsymbol{s}$ is denoted as $\boldsymbol{s} = (s_1, \ldots, s_N)$. s_i is denoted by a five-tuple $s_i = \left(f_{i,in}, d_i, v_i, w_i, f_{i,out}\right)$, where $f_{i,in}$ is the segment inflow, $f_{i,out}$ is the segment outflow, d_i is the segment density, v_i is the average speed, and w_i is the waiting vehicles (speed less than 0.1 m/s) in the segment. All the tuple elements can be retrieved via different sensors, as shown in Fig. 6. Note that the upstream outflow $f_{i,out}$ is equal to downstream inflow $f_{i+1,in}$, so the duplicated state elements are deleted in practice. The state information is refreshed every second.

C. The Action of the DRL Model

Actions are the output of the DRL model. In each step, the DRL model receives a global state $\mathbf{s}$ and produces a joint action. $\boldsymbol{a} = (VSL_1, \ldots, VSL_n, RM_1, \ldots, RM_n)$ represents the joint action of all the controllers, where VSL_i represents the speed limit of ith VSL controller, and RM_i represents the ith ramp outflow. Both VSL_i and RM_i have a range. $VSL_i \in [VSL_{min}, VSL_{max}]$, where the $VSL_{min} = 8.33$m/s, and the



(a) The ramp metering lights VSL control area

(b) The detectors and controllers of bottleneck 1 area

(c) The detectors and controllers of bottleneck 2 area

Fig. 5. The detectors and controllers of scene two.

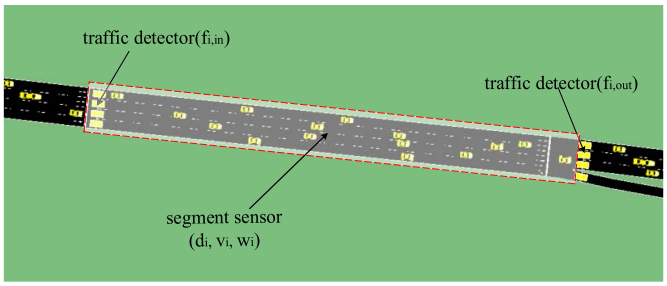


Fig. 6. The segment state retrieved by different sensors in SUMO.

VSL_{max} = 27.78m/s. Similarly, $RM_i \in [RM_{min}, RM_{max}]$, where the RM_{min} = 4 veh/min, and the RM_{max} = 30 veh/min. The RM_i has the same magnitude with VSL_i so the actors could output them correctly.

A double-layer architecture is proposed to solve the asynchronous problem of VSL and RM controllers. Instead of directly using the signal phases as actions, the upper layer uses the ramp outflow RM_i as an action. The lower layer can convert it to the RM signal phases in the next step. For instance, if the control period of VSL_i is one minute, then RM_i can also set to outflow per minute, and then turn into traffic phase duration with Algorithm 1:

Fig. 7 illustrates how the ramp metering and VSL controllers keep in pace. VSL speed limitations and ramp outflow changes once per minute. Then the ramp flow can convert into the green and red phases via Algorithm 1. The different outflow has different duration for green and red phases. When the ramp outflow is large, the green duration becomes long. Otherwise, the green duration becomes short. In this way, RM signals can easily keep pace with the VSL controllers,

Algorithm 1 RM Signal Phase Conversion

```
Initialize: time step (TS), control period (CP), ramp outflow
(RO), green-time per vehicle (GPV), vehicle number (VN)
for TS = 1, N, do:
  if TS mod CP = 0: //begin a control period
    Delete previous green or red phase
    Convert RO to VN/minute
    Green phase duration = GPV×VN/minute
    Red phase duration = CP-green phase duration
    Convert duration to phases in CP
  else //Regular step of CP
    if TS in green phase:
      RM signal phase set green
    else TS in red phase:
      RM signal phase set Red
    end if
  end if
end for
```

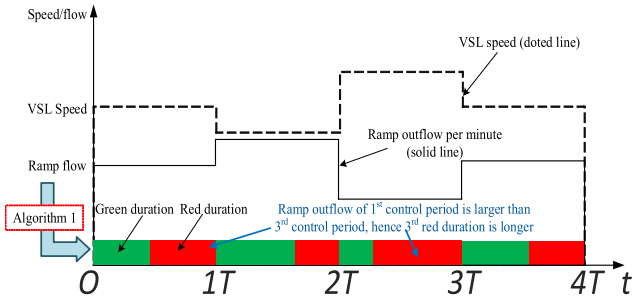


Fig. 7. The control period of VSL and the signal phase of ramp metering.

and the phase duration can adjust flexibly. Finally, the control system decomposes joint action into independent strategies for traffic controllers.

D. The Reward of the DRL Model

The immediate reward $r \in \mathbb{R}$ is a scalar value the DRL model receives each step after taking a specific action. With a proper reward r the agent can gradually converge to the optimal action. Traffic in the microscopic simulation environment is sophisticated, and the reward is often disturbed by stochastic noise. Hence, a stable reward that can be easily distinguishable is required. We consider three key factors of freeway traffic mobility: the traveling speed on the road, the on-ramp queue restriction, and the waiting (halting) vehicles in each segment.

1) The Average Traveling Speed on the Road: The average traveling speed on the road is considered the major term to measure the traffic mobility of the freeway network. Furthermore, if the traveling speed of each segment is evenly distributed as time goes, it implies that the traffic is stable. The average traveling speed reward can be described as:

$$\mathrm{V}_t = \frac{1}{N}\sum_i^N \bar{v}_{i,t} \tag{10}$$

where $\bar{v}_{i,t}$ is the average speed of ith segment at time step t, and N is the total segments of the freeway network.

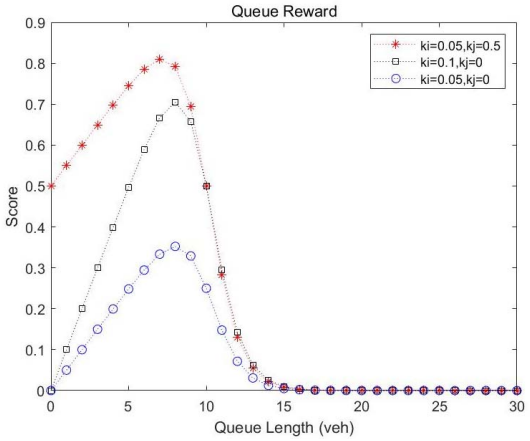


Fig. 8. Reward function of on-ramp queue restriction.

2) On-Ramp Queue Restriction: On-ramp queue restriction q_{RM} is proposed because of the ramp road capacity. Previous researches seldom considered the waiting vehicles on the ramp road. However, they cannot ignore since they may spread to the adjacent highway and cause congestion. Therefore, the on-ramp queue restriction is defined as a positive value if not exceed the maximum desired length, otherwise dived to zero. We selected the combination of sigmoid function and the linear function, which can be written as:

$$q_{RM} = \frac{k_i q_t + k_j}{1 + e^{q_t - q_{max}}} \tag{11}$$

where q_t is the dynamic queuing limit of the on-ramp, q_{max} is the longest desired queue. $k_{i,j}$ refers to the coefficient of reward. Eq. (11) indicated that when $q_t \leq q_{max}$, there is no penalty; when $q_t > q_{max}$, q_{RM} would dive to zero. Hence, the vehicle queue can be restricted. In this study, we set q_{max} as 15 vehicles for each on-ramp. Fig. 8 has shown the curves of different $k_{i,j}$. Considering the balance of the ramp outflow and the waiting queue, we set $k_i = 0.05$, $k_j = 0.5$.

3) The Waiting Vehicles: Waiting vehicles include the vehicles waiting on the ramp road and the mainstream during congestion. Since the objective is to minimize traveling time, waiting time is essential for evaluating freeway mobility. To avoid the overlong queues in the road network, the waiting vehicles played as the penalty term, which is given in Eq. (12):

$$W_t = -\frac{1}{N}\sum_i^N k_i(w_{i,t} + \Delta w_{i,t}) \tag{12}$$

where $w_{i,t}$ is the dynamic waiting vehicles on segment i on time step t, k_i is the weight of each segment, $\Delta w_{i,t} = w_{i,t} - w_{i,t-1}$ is the increment of waiting vehicles. With $\Delta w_{i,t}$, an extra incentive is received when the waiting vehicle number is decreasing, an additional penalty is received vice versa. Combining Eq. (10) $\sim$ (12), we propose the reward function as:

$$r_t = W_1 * \mathrm{V}_t + W_2 * q_{RM} + W_3 * W_t \tag{13}$$

where W_i is the weight of different terms, in this paper we define $W_1 = 0.25$, $W_2 = 0.25$, $W_3 = 0.5$.

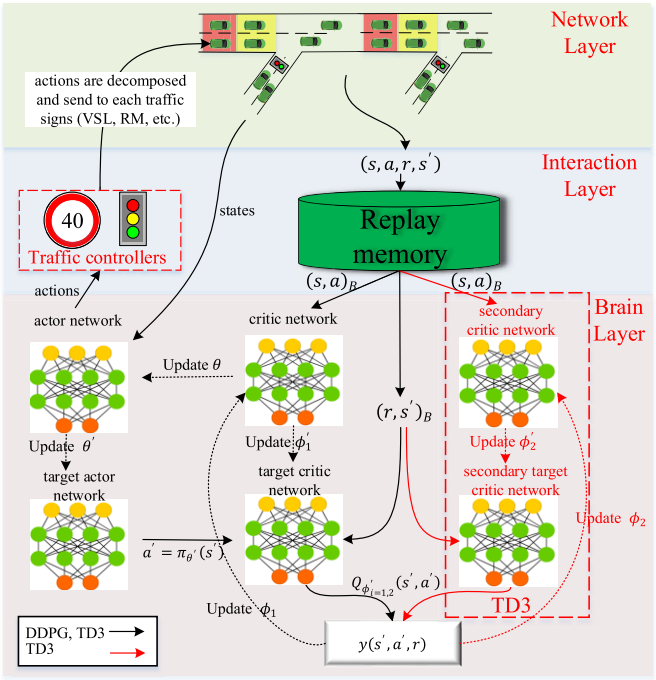


Fig. 9. A generalized framework for deterministic actor-critic algorithms.

VI. Deep Reinforcement Learning Algorithms

A. The Framework for Deep Actor-Critic Algorithm

Our control system provides a generalized framework for deep actor-critic algorithms. The framework is designed based on the similarity of deep deterministic actor-critic algorithms. Therefore, our framework is very flexible towards extensions. This paper has implemented the most prevailing algorithms on our framework, the DDPG [36] and the TD3 [29], with improved exploration techniques and the prioritized experience replay. The framework is in Fig. 9.

The framework has three layers: First is the Network Layer, where the control system retrieves traffic states via the RSU and controls vehicles with different traffic signs as mentioned in section III. The second is the Interaction Layer. It contains two modules: 1. the traffic controllers, including the VSL and RM signal controllers, which turn the output of actor into specific traffic commands; 2. the replay memory, which stores a vast of historical traffic state data from various sensors each step, the states are aggregated as experience (s, a, r, s'). The experiences are randomly sampled and sent to the critic networks for training.

The third is the Brain Layer. As shown in Fig. 9, different algorithms have similar and separate networks, marked with different colors in the Brain Layer. Generally speaking, a deep actor-critic algorithm has an actor with parameter θ and a critic with parameter ϕ. As mentioned in section II, the actor generates the optimal policy π_θ based on the critic network Q_ϕ with Eq. (3). To provide the stable update in each iteration, separate target networks with the same architecture but different parameter θ' and ϕ' are used. In addition, to solve the overestimate problem, another critic network is used in the TD3 algorithm. To train Q_ϕ with Eq. (4), a batch of

TABLE V
NOTATIONS

Notation	Meaning
π_θ	policy (actor) with parameter θ
$\pi_{\theta'}$	target actor with parameter θ'
Q_ϕ	critic with parameter ϕ
$Q_{\phi'}$	target critic with parameter ϕ'
s	certain state
a	certain action according to state s
s'	consequent state of s
a'	action according to s' following policy π_θ
r	the reward of state s
B	size of a mini-batch replay memory
$y(s', a', r)$	learning target, Q_ϕ is updated by minimizing loss with $y(s', a', r)$

experiences $\{(s, a, r, s')\}_B$ are sent to the critic networks for evaluation. Then, action a' can be calculated according to target policy $\pi_{\theta'}(s')$. Finally, Eq. (4) can be calculated with (s, a, r, s', a'). The meaning of notations in Fig. 9 are shown in Table V.

B. Training With the Deterministic Actor-Critic Algorithms

1) The DDPG Algorithm: Eq. (3) and (4) are the major equations of the DDPG algorithm. In practice, however, more details required consideration. The first problem is how to balance exploration and exploitation. Since the actions of DDPG are deterministic, stochastic noise must add for exploration. In this paper, we propose two techniques to improve explore efficiency:

- Uniform exploration

With the first few episodes, the agent selected actions with the uniform distribution U (or random selection).

- Reward Memorizing

The agent could remember the best reward R_{max} and the worst reward R_{min} during training, as the reference of its latest strategy (with the reward of r_{last}) in Eq. (14):

$$\mathcal{N}_{exp} = \mathcal{N}_{basic} + \xi \frac{R_{max} - r_{last}}{R_{max} - R_{min}} \tag{14}$$

where $\mathcal{N}_{exp}$ is the total exploration noise, $\mathcal{N}_{basic}$ is the basic noise with Gaussian distribution. ξ is the coefficient to measure the importance of the agent's latest action performance. In this paper, we set $\mathcal{N}_{basic} = 0.1$ and $\xi = 0.4$. If $r_{last} \ll R_{max}$, the second term becomes prominent, and the agent will do more exploration in the following episodes, which is good for the agent to overstep the local optimum.

The second problem is how to calculate $E_{s\sim p^\pi}[\cdot]$ of Eq. (3). $E_{s\sim p^\pi}[\cdot]$ can be estimated with a batch of randomly sampled experiences $(s, a, r, s')_B$ from replay memory, where B is the batch size. To improve the efficiency of parameter updating, a prioritized experience replay strategy is selected [42], in which the temporal difference error δ_i ranks the priority of an experience sample i:

$$\delta_i = \left| r + \gamma\, Q_{\phi'}\left(s', \pi_\theta\left(s'\right)\right) - Q_\phi\left(s, a\right)\right| \tag{15}$$

The following equation calculates the probability P_i of sampling experience i:

$$P_i = \frac{p_i^\tau}{\sum_{k\in B} p_k^\tau}, \quad p_i = \frac{1}{rank\left(\delta_i\right)} \tag{16}$$

where τ represents how much prioritization is used, when τ becomes zero, it is random sampling. With the replay examples, we can train the critic by minimizing the loss function $L(\phi)$:

$$L\left(\phi\right) = \frac{1}{|B|}\sum\nolimits_{(s_i, a_i, r_i, s_i')\in B}\left(y_i - Q_\phi\left(s_i, a_i\right)\right)^2 \tag{17}$$

where

$$y_i = r_i + \gamma\, Q_{\phi'}\left(s_i', a_{i|a_i'=\pi_{\theta'}(s_i')}'\right) \tag{18}$$

Here the i index refers to the ith sample. y_i is the optimal target, computed from reward r_i and outputs of the target actors and critics, with parameters $\pi_{\theta'}$ and $Q_{\phi'}$. In practice, parameter ϕ is updated with the gradients obtained from the loss function with the Adaptive moment estimation (Adam) optimizer [43].

The actor is updated by maximizing the future reward, or the value Q_ϕ of critic networks, hence the updating of an actor can be written as:

$$\nabla_\theta J\left(\pi_\theta\right) = \frac{1}{|B|}\sum\nolimits_B\left[\nabla_a Q_\phi\left(s, a\right)\nabla_\theta \pi_\theta\left(s\right)\right], \quad a = \pi_\theta\left(s\right) \tag{19}$$

where $\nabla_\theta J(\pi_\theta)$ denotes the gradient of policy π_θ, B denotes the size of the sampled batch, $Q_\phi(s, a)$ denotes the evaluations from the critic, $\nabla_\theta J(\pi_\theta)$ can be solved by the chain rule [35]. The input of DDPG is the states received from the detectors in the simulation, and the output of the DDPG is the joint action of the RM outflow and VSL limitation values. The state and action space have been discussed in the previous section.

2) The TD3 Algorithm: The TD3 algorithm has three major improvements. First is the Target Policy Smoothing, where a stochastic noise is added to $\pi_{\theta'}(s_i')$ in Eq. (18) as follows:

$$a_i^{''} = clip\left(\pi_{\theta'}\left(s_i'\right) + clip\left(\epsilon, -c, c\right), a_{Low}, a_{High}\right) \tag{20}$$

where $\epsilon \sim \mathcal{N}(0, \sigma)$ is a stochastic noise subject to the normal distribution. $clip$ is a function to limit the variables to a specific range. e.g., $a_{Low} \le a_i^{''} \le a_{High}$. The limit of noise clip $c = 0.5$. Target Policy Smoothing essentially serves as a regularization term. Since the DDPG is deterministic, it may fall into local optimum. The regularization term of TD3 can help the agent evade the local optimum.

The second improvement is the Double Q Learning, where a secondary critic network is employed to solve the overestimation problem [29], the minimal Q value will be taken as the result. Thus, Eq. (18) can be rewritten as:

$$y_i = r_i + \gamma \min_{i=1,2} Q_{\phi_{i=1,2}'}\left(s_i', a_i^{''}\right) \tag{21}$$

Note that the two critics are updated with $L\left(\phi_{i=1,2}\right)$ in Eq. (17). The actor is updated in Eq. (19) with Q_{ϕ_1}.

The third improvement is the Delayed Policy Updates, which means the actor does not update as frequently as the critic network, which will add the training stability.

The action exploration techniques and prioritized experience replay are also used in the TD3 algorithm. To sum up the above equations, the deep actor-critic traffic control algorithms are proposed with pseudocode in Algorithm 2.

Algorithm 2 DDPG and TD3 Algorithm for Freeway Traffic Control

Initialize critic networks $Q_{\phi_{i=1,2}}$, actor network π_θ
Initialize target networks with $\phi'_{1,2} \leftarrow \phi_{1,2}$, $\theta' \leftarrow \theta$
Initialize replay memory R, random noise N_{exp}
for episode = 1 to M, do:
  Initialize simulation environment s_0 in SUMO
  for simulation step = 1 to T, do:
    Receive state s_t from detectors and sensors in SUMO
    Select action a_t according to $\pi_\theta(s_t)$ with noise $\epsilon \sim \mathcal{N}_{exp}$
    Transfer a_t to traffic commands in SUMO
    Observe reward r_t and new state s_{t+1}
    $R \leftarrow (s_t, a_t, r_t, s_{t+1})$
    Select B samples from R based on the sampling priorities
    if DDPG:
      Update Q_ϕ by minimizing the loss $L(\phi)$ with (17) and (18)
      Update π_θ by sampled gradient with (19)
    else: //TD3
      Update $Q_{\phi_{i=1,2}}$ by minimizing the loss $L\left(\phi_{i=1,2}\right)$ with (17) and (21), update a_{t+1} with target policy smoothing (20)
      Update π_θ by sampled gradient with (19) with policy decay
    end if
    Update transition priorities with (15) and (16)
    Update the target network:
      $\theta' = \tau\theta + (1-\tau)\theta'$
      $\phi'_{(1,2)} = \tau\phi_{(1,2)} + (1-\tau)\phi'_{(1,2)}$//the TD3 has two critics
  end for
end for

C. The Deep Neural Network Structure

Fig. 10 shows the structure of the actor and critic networks. Both networks contain the input (output), batch normalization (BN), and dense layers. The BN layer normalizes the scale differences from different sensors' inputs. The concatenation layer in the critic network merges the observations and actions into one layer. The dense layers contain many trainable units that are updated during training. The layer-wise propagation rule can be written as:

$$\boldsymbol{H}_N^{l+1} = \alpha_N\left(\boldsymbol{W}_{NM}\boldsymbol{H}_M^l + \boldsymbol{b}_N^l\right) \tag{22}$$

where $\boldsymbol{H}_M^l$ represents the output of lth layer with M neurons, N is the number of next layer's neurons. α represents the activation function. Most layers use the Rectified Linear Unit function (Relu) as the activation function. $\boldsymbol{W}$ represents the trainable weights, and $\boldsymbol{b}$ represents the bias, respectively. Each dense layer has 256 neurons. Thus, $\boldsymbol{W} \in R^{256\times256}$, $\boldsymbol{H} \in R^{256}$, and $\boldsymbol{b} \in R^{256}$. Note that the last layer of the actor network uses the $tanh$ activation function to restrict the action range.

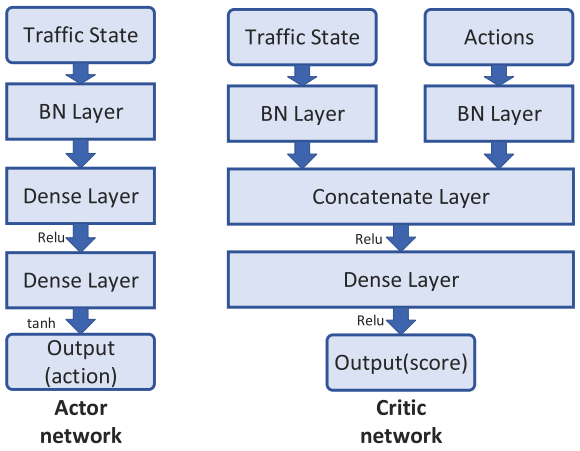


Fig. 10. Structure of the actor and critic network.

TABLE VI
SIMULATION PARAMETERS SETTING

Parameters	Value
Max speed	27.78m/s (100km/h)
Max acceleration	2.6m/s^2
Max deceleration	-4.5m/s^2
Simulation time per episode	7200s (2 hours)
Max episodes	200
Control period	60s
Discount factor γ	0.99
Training batch size B	300
Memory length	100000
Target network update rate τ	0.001
Learning rate α	0.0001

VII. EXPERIMENTAL RESULTS

A. Experiment Set-Up

This section evaluates our methods through simulation experiments in two different scenes. In the first scene, we focus on comparing different algorithms, while in the second scene, we focus on the performance of proposed DRL algorithms in a larger scale road network. The details of the two simulation environments are in section V. The environment parameters' configurations are in Table VI.

B. Compared Methods

To test the effectiveness of our control system, we compared the following algorithms. All algorithms use the same states, actions, and rewards defined in section V.

(1) Fixed-time ramp metering control (FTC): Ramp metering signal circulates in a fixed phase sequence, with 30 seconds of green/red phase. The fixed-time ramp metering control is activated when the mainline traffic flow is above 1000 vehicles/lane/hour.

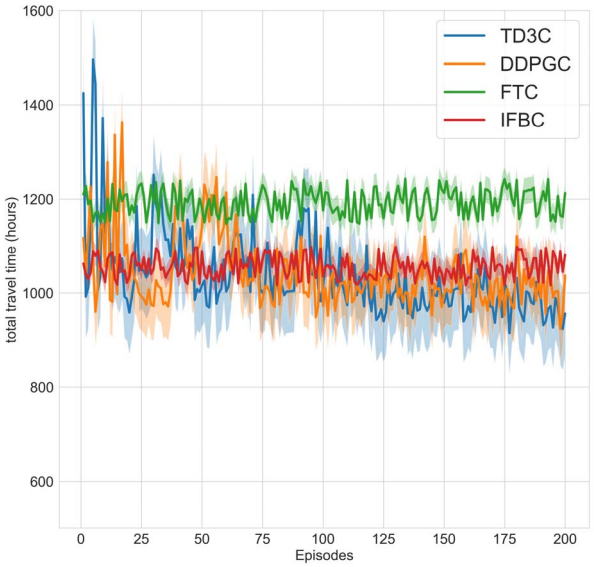


Fig. 11. Total travel time comparison of algorithms in scene one.

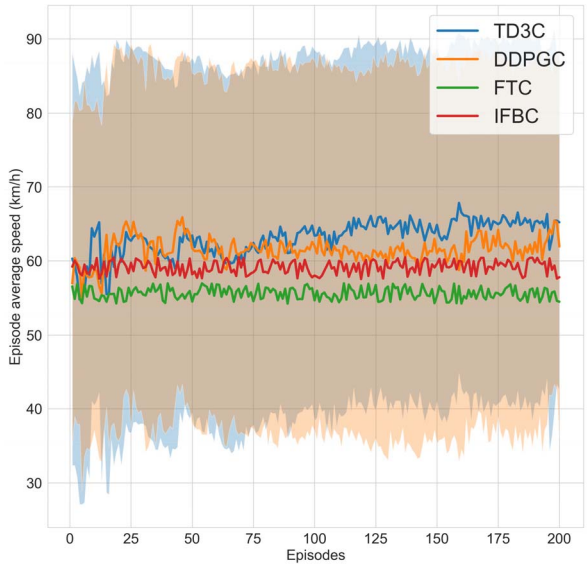


Fig. 12. The average speed of the road network.

(2) Integrated feedback control (IFBC): The VSL signs and ramp metering signal automatically change according to bottleneck traffic occupancy and the ramp queue length.
(3) DDPG control with improved exploration and prioritized replay (DDPGC): A traffic control unit controls all the VSL and RM controllers with the DDPG algorithm to find the optimal control policy.
(4) TD3 control with improved exploration and prioritized replay (TD3C): A traffic control unit controls all the VSL and RM controllers with the TD3 algorithm to find the optimal control policy.

C. Result in Scene One

In this section, all the algorithms aim to minimize the total travel time with the restriction of the on-ramp queue. The total travel time includes vehicle traveling time and waiting time on the road during simulation. Three key indicators, including the total travel time, the average speed, and the episodes' reward, are considered. The results after 200 episodes of training (which are 400 simulation hours) are demonstrated from Fig.11 to Fig. 13.

The comparison of total travel time is shown in Fig. 11. The solid line indicates the mean value of each episode, and the shallowed area indicates the standard deviation. From Fig.11, we can see that the IFBC and DRL methods can significantly reduce total travel time during rush hours. The reason is that the IFBC and DRL methods are adaptive, they can maintain a stabilized outflow to evacuate oversaturated traffic more efficiently. In contrast, the fixed time control cannot adjust its behavior according to traffic variation and may even lead to congestion due to improper actions. From the result, the DRL methods slightly outperformed the IFBC method, this is achieved by continuous self-learning. Fig.11 showed that the total travel time of FTC is about 1200 hours, the total travel time of IFBC is about 1050 hours, and the total travel time of DRL algorithms is around 950 hours. It can be found that the

Fig. 13. Reward comparison of the TD3C and DDPGC.

IFBC can save 12.5% of travel time, while the DRL algorithms can save 20.83% of travel time, indicating the effectiveness of DRL algorithms. Fig. 12 showed that the average speed of the road network is also increased, from 55 km/h of FTC to 65km/h of TD3C. This indicated that the increased traveling speed and the reduced waiting vehicle number contribute to the reduced total travel time. Moreover, results showed that the deep actor-critic algorithms could outperform the IFBC and FTC methods through continuous learning.

We further compared the cumulative rewards of DDPGC and TD3C in Fig. 13. Our goal is to maximize the reward in Eq. (13), and the higher reward implies less waiting time or higher traveling speed. We can see that both DDPGC and TD3C have an upward reward trend during training, but the fluctuation is significant. The reason is that the saturated traffic

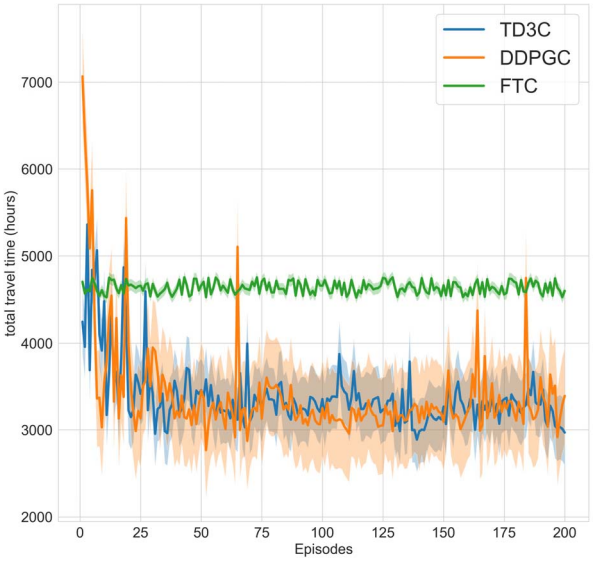


Fig. 14. Total travel time of TD3C, DDPGC, and FTC in scene two.

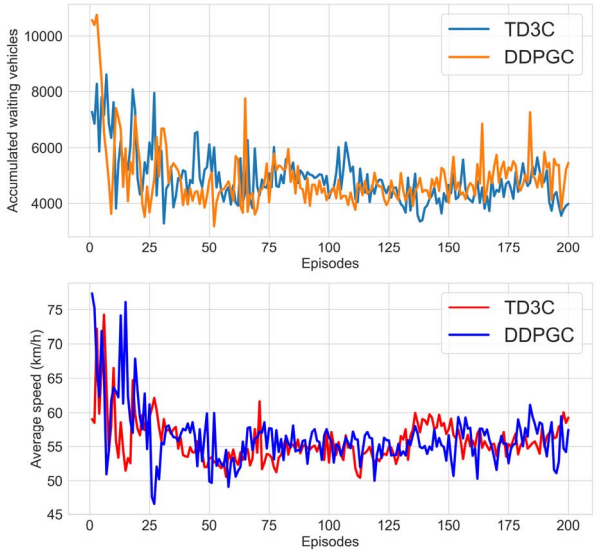


Fig. 15. Accumulated waiting vehicles and the average speed of the road network of TD3C and DDPGC during training.

is unstable and sensitive to disturbances, while under training, there are random actions (trial and error) that may lead to performance degradation. From the result, it can be seen that although the fluctuation of TD3C was more significant in the first few episodes, it quickly found the optimal actions and then had minor fluctuations and better results. Obviously, the TD3C is more stable, especially for the traffic speed. Harmonious speed is beneficial when the traffic flow is saturated.

D. Result in Scene Two

1) The Performance on an Extensive Network: he result of simulation scene two is shown in Fig. 14. Compared with the FTC, the total travel time of deep actor-critic algorithms (i.e., DDPGC and TD3C) decreases from 4620 hours (FTC) to 3000 hours on average, saving 35% more time than the FTC strategy. Fig.15 illustrates that the waiting time contributes to the major part of the traveling time saved, where the number of the waiting vehicles has reduced by nearly one-half. It implies that the freeway corridor can bear more vehicles on the road with the adaptive control of DRL methods. However, since the traffic on the road is more saturated, the average speed decreases slightly from 65 km/h to 60 km/h. Nevertheless, the overall traveling time is reduced.

In our cases, comparisons between TD3C and DDPGC have shown that both algorithms can achieve optimal results. However, the training process of TD3C is more stable, and the outcome of TD3C is slightly better than the DDPGC. We believe that the Target Policy Smoothing and the Delayed Policy Update techniques contributed to the stability of TD3C, and the double critic networks contributed to the better result since they could evaluate the policy more accurately. Generally speaking, the freeway traffic volume is much heavier than urban traffic, and the average speed is much higher. Hence, even tiny imperfections may lead to obvious decay. This can explain the pulses of DDPGC during training. Therefore, the stability of TD3C is important for freeway traffic management.

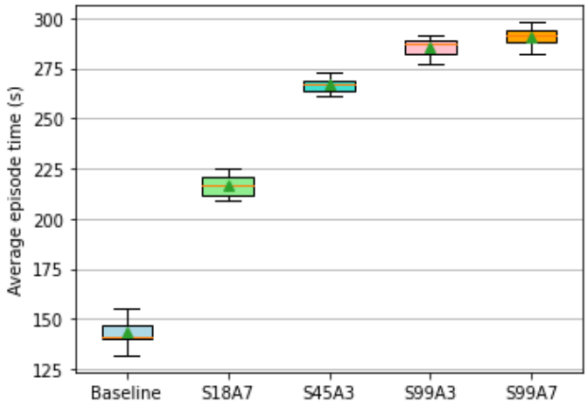


Fig. 16. Execution time of an episode for different problem scales.

In summary, the proposed system with TD3C can satisfy the stability and accuracy requirements and achieve good performance under rush hours.

2) The Scalability and Computational Performance: The scalability and computational performance is key issue for reinforcement learning. Classic RL methods suffer from the curse of dimensionality when the state and action spaces become large. Thus, we have investigated the performance of the proposed system from two aspects. First is the training episodes to converge. Compared Fig. 11 with Fig. 14, we could see that the total travel time became stable within 200 episodes in both scenes, indicating that the training episodes did not increase for the large road network.

The second is the execution time of an episode. When the state and action space grows large, the deep neural network is more complicated and requires more time for training. To test the execution time variation, we designed an experiment in Fig. 16, where the "S18A7" denotes the control system that receives 18 states and outputs seven actions.

In other words, the system controls seven traffic controllers simultaneously. Similarly, "S45A3" denotes the control system receives 45 states and outputs three actions. All the cases were tested in scene two using TD3C, and the baseline was a pure simulation without traffic control. Every case ran ten times to collect the average performance data. Fig. 16 illustrates that as the state and action space grows large, the execution time rises in a logarithmic curve, which means the computational performance decreases slower than the increment of the problem scale. The test results are promising: the proposed system does not suffer from the curse of dimensionality and the execution time is acceptable for large-scale network control. In the worst-case "S99A7", two hours of compressed simulation data can be trained within five minutes. Note that the results were tested with the Intel i7-10710U CPU and 16 GB RAM; the execution time can be further decreased with more powerful devices.

The reason behind this is the structure of the deep actor-critic algorithm. Unlike the classic RL methods, the actor network's output (last) layer can directly produce the actions for traffic controllers. Each output layer neuron can produce an independent numerical action. The results from different neurons can easily combine to the joint action. The scalability of state (input) is similar to the action. In a word, the size of the neural network grows linearly as the size of action and state space increases, which is acceptable for large-scale control.

VIII. Conclusion

This paper proposed a centralized traffic control system that integrates multiple ramp metering and VSL traffic controllers to relieve traffic congestion on the freeway. The centralized system has a straightforward structure and is robust. Besides, we have introduced deep actor-critic algorithms to solve the curse of the dimensionality problem for the control system. Simulation results indicated that the time complexity of the deep actor-critic algorithms increases linearly when controlling more traffic controllers on a larger network. Therefore, our system could effectively manage complicated freeway bottleneck sections with various traffic controllers using a centralized traffic control unit.

In addition, we developed a generalized framework that can implement different deep reinforcement algorithms for further extension. The framework uses a double-layer architecture to synchronize multiple traffic controllers. The upper layer focuses on the cooperation between traffic controllers, while the lower layer decomposes the collaboration into independent strategies for different traffic controllers. We also improved the exploration strategy. The new design allows more explorations when the reward is relatively low so that there is a higher chance to evade local optimum. The SUMO simulation is utilized to evaluate the system performance. Two scenes are discussed: a single ramp weaving section and a freeway corridor with multiple on-ramps and off-ramps. Results have demonstrated that our system can learn a good policy in both scenes and remarkably reduce the total travel time. In the first scene, the total travel time has decreased by 20%, and in the second scene, the total travel time has reduced by 35%, compared with the fixed-time control. The TD3 algorithm outperforms other control methods in performance and is more stable compared with the DDPG control method. Results also pointed out that the proposed centralized control system could guide multiple traffic controllers effectively in a medium-scale freeway network. This concludes that we could save computing resources by controlling multiple traffic controllers with a centralized traffic control unit without noticeable performance decay.

In the future, we will expand our research to a large-scale road network, focusing on finding the equilibrium point of the traffic controllers' scalability and the control algorithms' effectiveness. The cooperation method will also upgrade to allow more advanced control strategies.

References

[1] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: An overview," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 4, pp. 271–281, 2002.

[2] X.-Y. Lu *et al.*, "Review of variable speed limits and advisories: Theory, algorithms, and practice," *Transp. Res. Rec.*, 2018.

[3] A. Hegyi, B. De Schutter, and H. Hellendoorn, "Model predictive control for optimal coordination of ramp metering and variable speed limits," *Transp. Res. C, Emerg. Technol.*, vol. 13, no. 3, pp. 185–209, 2005.

[4] R. C. Carlson *et al.*, "Optimal motorway traffic flow control involving variable speed limits and ramp metering," *Transp. Sci.*, vol. 44, no. 2, pp. 238–253, 2010.

[5] Y. Liu, G.-L. Chang, and J. Yu, "An integrated control model for freeway corridor under nonrecurrent congestion," *IEEE Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1404–1418, 2011.

[6] I. Papamichail *et al.*, "Heuristic ramp-metering coordination strategy implemented at monash freeway, Australia," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2178, pp. 10–20, 2010.

[7] R. C. Carlson, I. Papamichail, and M. Papageorgiou, "Integrated feedback ramp metering and mainstream traffic flow control on motorways using variable speed limits," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 209–221, 2014.

[8] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019.

[9] O. Vinyals *et al.*, "Grandmaster level in starcraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[10] K. Rezaee, B. Abdulhai, and H. Abdelgawad, "Self-learning adaptive ramp metering: Analysis of design parameters on a test case in Toronto, Canada," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2396, pp. 10–18, 2013.

[11] Z. Li *et al.*, "Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3204–3217, 2017.

[12] F. Zhu and S. V. Ukkusuri, "Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 41, pp. 30–47, 2014.

[13] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "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.*, vol. 14, no. 3, pp. 1140–1150, 2013.

[14] C. Wang *et al.*, "A new solution for freeway congestion: Cooperative speed limit control using distributed reinforcement learning," *IEEE Access*, 2019.

[15] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graph," in *Proc. 19th Eur. Conf. Mach. Learn.*, 2008, pp. 656–671.

[16] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*, 2016, pp. 47–66.

[17] F. Belletti *et al.*, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1198–1207, 2018.

[18] Y. Wu *et al.*, "Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm," *Transp. Res. C, Emerg. Technol.*, vol. 117, 2020, Art. no. 102649.

[19] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, 2020.
[20] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intell. Transp. Syst.*, vol. 11, no. 7, pp. 417–423, 2017.
[21] T. Wu *et al.*, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8243–8256, 2020.
[22] S. Lin *et al.*, "On a spatiotemporally discrete urban traffic model," *IET Intell. Transp. Syst.*, vol. 8, no. 3, pp. 219–231, 2014.
[23] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[24] M. Hessel *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. AAAI*, vol. 32, no. 1, Apr. 2018, pp. 3215–3222.
[25] F. Rasheed, K.-L. Yau, R. M. Noor, C. Wu, and Y.-C. Low, "Deep reinforcement learning for traffic signal control: A review," *IEEE Access*, Oct. 2020.
[26] X. Liang *et al.*, "'A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, 2019.
[27] Y. Gong *et al.*, "Decentralized network level adaptive signal control by multi-agent deep reinforcement learning," *Transp. Res. Interdiscipl. Perspect.*, vol. 1, 2019, Art. no. 100020.
[28] T. Chu *et al.*, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–10, 2019.
[29] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Mach. Learn. Res.*, 2018, p. 80.
[30] T. Haarnoja *et al.*, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 1861–1870.
[31] B. Khondaker and L. Kattan, "Variable speed limit: A microscopic analysis in a connected vehicle environment," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 146–159, 2015.
[32] M. Wang *et al.*, "Connected variable speed limits control and car-following control with vehicle-infrastructure communication to resolve stop-and-go waves," *J. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 559–572, 2016.
[33] Y. Wang *et al.*, "Automated on-ramp merging control algorithm based on internet-connected vehicles," *IET Intell. Transp. Syst.*, vol. 7, no. 4, pp. 371–379, 2013.
[34] C. Rong, L. S. Jin, and J. Yu, "Research on safety driving assistant system of expressway on-ramp merging area based on wireless network communication," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2009, pp. 3139–3144.
[35] E. R. Müller *et al.*, "Microsimulation analysis of practical aspects of traffic control with variable speed limits," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 512–523, 2015.
[36] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
[37] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016.
[38] M. Papageorgiou, H. Hadj-Salem, and J. M. Blosseville, "ALINEA: A local feedback control law for on-ramp metering," *Transp. Res. Rec.*, vol. 1320, no. 1, pp. 58–67, 1991.
[39] D. Krajzewicz *et al.*, "Recent development and applications of SUMO-simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, nos. 3–4, 2012.
[40] S. Krauß, "Towards a unified view of microscopic traffic flow theories," *IFAC Proc. Volumes*, vol. 30, no. 8, pp. 901–905, 1997.
[41] GitHub. *Alicante-Murcia-SUMO-Scenario*. Accessed: Jan. 12, 2022. [Online]. Available: https://github.com/jjgonde/Alicante-Murcia-SUMO-Scenario
[42] Y. Hou *et al.*, "A novel DDPG method with prioritized experience replay," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, 2017, pp. 316–321.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Chong Wang (Member, IEEE) received the B.S. degree from the Nanjing University of Information Science and Technology, Nanjing, China, in 2009, the M.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, in 2012, and the Ph.D. degree from Southeast University, Nanjing, in 2019.

He was a Software Engineer with Nanjing Fujitsu Nanda Software Technology Company Ltd., from 2012 to 2014. He is currently a Lecturer with the School of Computer and Software, Nanjing University of Information Science and Technology. His research interests include machine learning, connected vehicles, control theory with the Internet of Things, and intelligent transportation systems.

Yang Xu received the B.S. degree in instrument science from the East China University of Technology, Nanchang, China, in 2012, the M.S. degree in instrument science from the Hefei University of Technology, Hefei, China, in 2015, and the Ph.D. degree in instrument science from Southeast University, Nanjing, China, in 2019.

He is currently a Lecturer with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing. His research interests include machine learning and data science.

Jian Zhang (Member, IEEE) received the B.S. degree in applied mathematics and the Ph.D. degree in transportation engineering from Southeast University, Nanjing, China, in 2006 and 2011, respectively. From 2009 to 2010, he was a Visiting Scholar with the University of Wisconsin, Madison. He is currently a Professor with the School of Engineering, Tibet University, and the Vice Director of the Research Center for Internet of Mobility and the Research Center for National Transport Strong Strategy, Southeast University. He has authored or coauthored over 80 articles in journals. His research interests include intelligent transportation systems, connected and automated transportation, vehicle scheduling, and machine learning.

Bin Ran received the Ph.D. degree from the University of Illinois Chicago, Chicago, USA, in 1993.

He is currently a Professor with the Department of Civil and Environmental Engineering, University of Wisconsin–Madison, Madison, WI, USA, and the Director of the Research Center for Internet of Mobility, Southeast University, Nanjing, China. He has authored or coauthored more than 90 articles in international journals, including *Transportation Science*, *Transportation Research Part B: Methodological*, and *Transportation Research Part C: Emerging Technologies*. He is one of the co-founders of the Chinese Overseas Transportation Association (COTA), and he was the first chairperson.