Transportation Research Part C 117 (2020) 102649

Contents lists available at ScienceDirect

Transportation Research Part C

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

# Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm☆

Yuankai Wu[a], Huachun Tan[b,*], Lingqiao Qin[c], Bin Ran[b,c]

[a] Department of Civil Engineering and Applied Mechanics, McGill University, 817 Rue Sherbrooke, Montreal, OR H3A0C3, Canada
[b] School of Transportation Engineering, Southeast University, Sipailou 2, Nanjing, 210096, China
[c] Civil and Environmental Engineering, University of Wisconsin-Madison, 2312 Engineering Hall, 1415 Engineering Drive, Madison, WI 53706, United States

ARTICLE INFO

*Keywords:*
Variable speed limit
Deep reinforcement learning
Connected and autonomous vehicles

ABSTRACT

Variable speed limit (VSL) control is a flexible way to improve traffic conditions, increase safety, and reduce emissions. There is an emerging trend of using reinforcement learning methods for VSL control. Currently, deep learning is enabling reinforcement learning to develop autonomous control agents for problems that were previously intractable. In this paper, a more effective deep reinforcement learning (DRL) model is developed for differential variable speed limit (DVSL) control, in which dynamic and distinct speed limits among lanes can be imposed. The proposed DRL model uses a novel actor-critic architecture to learn a large number of discrete speed limits in a continuous action space. Different reward signals, such as total travel time, bottleneck speed, emergency braking, and vehicular emissions are used to train the DVSL controller, and a comparison between these reward signals is conducted. The proposed DRL-based DVSL controllers are tested on a freeway with a simulated recurrent bottleneck. The simulation results show that the DRL based DVSL control strategy is able to improve the safety, efficiency and environment-friendliness of the freeway. In order to verify whether the controller generalizes to real world implementation, we also evaluate the generalization of the controllers on environments with different driving behavior attributes. and the robustness of the DRL agent is observed from the results.

## 1. Introduction

Variable speed limits (VSLs) or speed harmonizations, have been studied for a long time (Khondaker and Kattan, 2015; Lu and Shladover, 2014). Specifically, VSLs have been shown to resolve traffic breakdowns, improve traffic safety, and brought environmental benefits. For example, the application of VSLs in Germany has shown that VSLs typically result in lower crash rates and a 5%~10% increase in capacity (Weikl et al., 2013). In the United Kingdom, VSLs increased capacity by 7% and decreased the overall congestion time (Middelham, 2006). In Netherlands, 20%~30% traffic emission reductions of NOx and PM10 were reported in test locations where VSLs were applied (MacDonald, 2008). VSL control is thus regarded as a hot topic in intelligent transportation systems.

Prior studies in VSL control can be categorized into: hand-crafted rule-based strategies (Soriguera et al., 2013; Piao and McDonald, 2008), in which speed limits are controlled with pre-defined rules; and proactive approaches (Hegyi et al., 2005b; Kattan

☆ This article belongs to the Virtual Special Issue on "Machine learning".

* Corresponding author.
*E-mail address:* tanhc@seu.edu.cn (H. Tan).

https://doi.org/10.1016/j.trc.2020.102649
Received 10 May 2019; Received in revised form 25 March 2020; Accepted 11 April 2020
Available online 24 June 2020

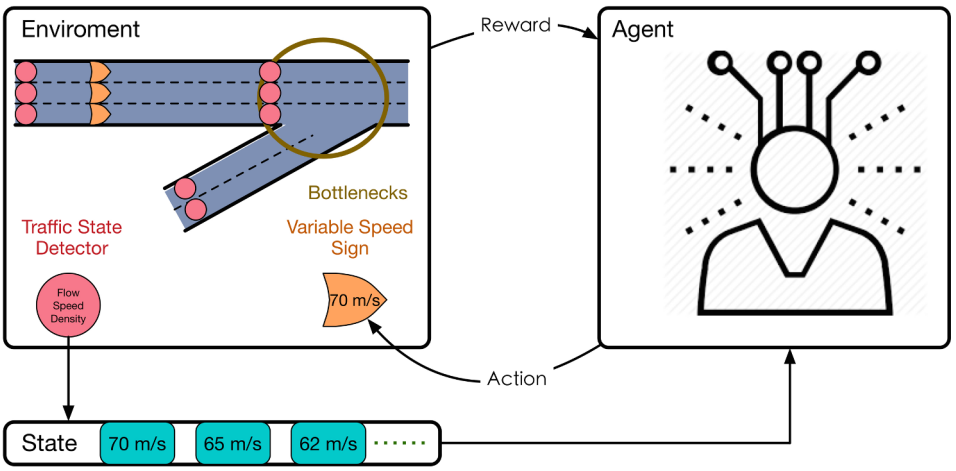



**Fig. 1.** Reinforcement learning framework for variable speed limit control.

et al., 2015), in which future traffic is predicted in such a way that traffic bottlenecks are anticipated. In the first case, the VSL decisions are specifically determined either by pre-defined time plans or by pre-selected thresholds of traffic state; in the second, proactive case, the controller automatically adjusts speed limits to avoid traffic breakdowns. There are some limitations in previous studies of these cases. For example, the rule-based approach requires expert knowledge from experienced engineers. Additionally, its robustness and generality cannot be guaranteed. The performance of proactive approaches is heavily dependent on the accuracy of traffic state prediction algorithms Zhang et al. (2019a,b). More specifically, the VSL controller takes actions shortly after a prediction time lag. If traffic state is largely varying over time, the controller may not be able to achieve the desired performance in real time.

Reinforcement learning (RL) provides great potential for addressing the limitations associated with the state-of-the-art VSL control strategies (Li et al., 2017; Zhu and Ukkusuri, 2014). A well-trained RL agent can theoretically achieve a proactive control scheme to optimize its benefits. The framework of RL-based VSL control is given in Fig. 1. For RL-based VSL control, the environment is composed of a transportation network with bottlenecks, traffic state detectors, and variable speed signs. State is a feature representation of the traffic state collected from detectors. Agent takes the state as an input and learns a model to change the speed limits. The speed limits are sent to the variable speed signs and the reward (e.g., the traffic state of bottlenecks) is sent back to the agent.

Two challenges arise in applying traditional RL to VSL control: (i) The difficulties in state summarization and representation; and (ii) the limited capability for learning correlations between a given environment and its corresponding optimal speed limit. In related fields including ramp metering (Belletti et al., 2018) and traffic signal control (Van der Pol and Oliehoek, 2016; Wei et al., 2018), recent studies have applied deep reinforcement learning (DRL) techniques to address these two challenges. In a DRL framework, massive traffic information collected from many different traffic detectors can be described as a vector or an image, and can be directly taken as input for neural networks of DRL. By the powerful function approximation properties of neural networks, the DRL agent can learn the optimal phase of traffic lights, succeeding controllers that used to be hand-engineered.

Another objective of this study is to allow VSL control to accommodate the emergence of connected and autonomous vehicles (CAVs) (Roncoli et al., 2015). Though CAVs have not yet been implemented in real world transportation system, it is believed that the CAV technology will alleviate congestion, improve traffic safety, and reduce vehicular emissions. In the authors' opinion, the CAV technology will bring two major benefits to VSL controls. The first is that CAVs naturally facilitate vehicle communication with infrastructures such as traffic state detectors, traffic lights, and variable speed signs. The second is that CAVs will ultimately lead to more rational and safe driving behavior, since human drivers will be replaced with artificial intelligence and automation. The benefits of modern VSL control strategies such as differential variable speed limits (DVSLs) among lanes, are dramatically limited by driver compliance (Schick, 2003). The impact of VSLs, in terms of safety and travel time, is quite sensitive to levels of driver compliance (Hellinga and Mandelzys, 2011). In a CAV environment, the message of dynamic speed limits can be sent to each individual vehicle, and these CAVs can be compelled to drive within these limits. As a result, the problem of driver compliance should be readily solved under a highly developed CAV environment.

In light of advances in DRL and CAV technologies, this study proposes a novel DRL approach to learn highly efficient VSL control strategies for recurrent freeway bottlenecks. The DRL-based VSL controller allows distinctive and dynamic speed limits among lanes, and is used to dynamically set different speed limits across multiple lanes to reduce congestion, accidents, and emissions. The controller is able to reason with a large number of possible speed limits at every time step. This is achieved by a novel deep actor-critic framework. The agent first produces a continuous action, and then find the closest possible speed limits. We test the proposed DRL-based DVSL controllers on a simulated freeway recurrent bottleneck. Results show that the DVSL control strategy is able to improve the efficiency and reduce the emissions of the freeway bottleneck. Moreover, numerous reward signals are used to train the actor-critic-based VSL controller and comparison between different reward signals is conducted.

The importance and contribution of this paper can be summarized as follows:

- We empirically demonstrate the limitation of having to rely on a single speed limits in the variable speed limits control for a multi-lane freeway. We propose a deep reinforcement learning framework for differential variable speed limits. The devised deep deterministic policy gradient algorithm is capable of producing dynamic and distinct speed limits of multiple lanes.
- The reward engineering and shaping issue for traffic control is considered in this paper, which has not been fully studied before in the transportation applications of reinforcement learning based control. We identify that there exist certain contradictions between different control aims.
- Experiments on both simulation with and without traffic incidents validate the effectiveness of our solution compared with state-of-the-art methods.
- The generalization of the DRL based DVSL agent is evaluated by testing it in the environments with different driving behavior attributes. To the best of our knowledge, this is the first systematic studies on the generalization of the DRL-based traffic controllers

The rest of this paper is organized as follows. Firstly, a literature review on related work is presented in Section 2. Then the problem statement is introduced in Section 3. The methodology is described in Section 4. The experimental results are shown in Section 5. Finally, some conclusions and remarks of this study is summarized in Section 7.

## 2. Related works

In this section, we firstly introduce conventional methods for VSL control, and then follow with an introduction to deep reinforcement learning and its bearing on related applications in intelligent transportation systems.

### 2.1. Conventional variable speed limits control

As mentioned in Section 1, VSL is essentially conducted in either rule-based or proactive ways. Early VSL studies have been mainly formulated as rule-based logic due to their relatively simpler problem settings. Dynamic speed limits are set according to predefined thresholds of traffic flow, occupancy, or mean speed. For example, Abdel-Aty et al. (2008) used speed differences between different sections as the threshold for changing speed limits, and indicated that VSLs can reduce crash probability. In Papageorgiou et al. (2008), a VSL system based on a flow–speed threshold was investigated. It was suggested that VSLs could be used in the interest of traffic safety rather than efficiency. The rule-based VSL systems have also achieved success in improving throughput and reducing travel time (Lin et al., 2004; Lyles et al., 2004). The rule-based method largely depends on the current traffic conditions, rather than using predictive information.

Proactive approaches additionally consider predictive information compared with rule based approaches. Hegyi et al. (2005a) demonstrated that model predictive controls for coordination control VSLs and ramp metering led to a 15% reduction in travel time. Carlson et al. (2010) showed that traffic flow can be substantially improved via proactive VSL control. The aforementioned studies and their models (Kattan et al., 2015; Hadiuzzaman and Qiu, 2013) all used prediction models to predict future traffic conditions. Hence, the success of proactive approaches is based on the robustness and reliability of a short-term traffic prediction model in representing future traffic states. However, accurate and reliable short-term traffic prediction (Wu et al., 2018; Li et al., 2018) is not an easy task because the evolution of a traffic state is related to many factors.

### 2.2. Deep reinforcement learning

The essence of RL is knowledge acquisition through interaction. In a typical RL framework, an autonomous agent, controlled by a machine learning algorithm, observes a traffic state in its environment. The agent interacts with the environment by taking an action. Then the environment transmutes to a new state as determined by the prior state and the chosen action. The objective of the agent is to maximize the accumulated rewards returned by the environment. Historically, RL had some successes in some areas (Singh et al., 2002; Ng et al., 2006). However, scalability and complexity issues have limited its application.

The advent of deep learning has dramatically improved RL. Typically “deep reinforcement learning” (DRL) is defined as the utilization of deep learning algorithms within RL. DRL has shown impressive successes in a wide range of tasks including playing video games (Mnih et al., 2015), defeating a human world champion in Go (Silver et al., 2016), controlling robots (Levine et al., 2018), and indoor navigation (Zhu et al., 2017). There are numerous DRL approaches including deep Q networks (DQN) (Mnih et al., 2015), and various policy gradient methods, such as TRPO (Schulman et al., 2015), A3C (Mnih et al., 2016), DDPG (Lillicrap et al., 2015), and PPO (Schulman et al., 2017). Those algorithms hold great promise in their ability to learn to solve challenging control problems.

Advances in DRL and big traffic data have led to potential applications of DRL techniques in tackling challenging control problems in intelligent transportation systems. DRL has given promising results in ramp metering (Belletti et al., 2018), traffic light control (Van der Pol and Oliehoek, 2016; Wei et al., 2018), fleet management (Lin et al., 2018) and hybrid electric vehicles energy management (Wu et al., 2019; Lian et al., 2020). Those works have many similarities to VSL systems in terms of problem settings.

## 3. Problem statement

In this section, we first present the differential VSL (DVSL) control example that considered in this paper. Then we briefly discuss

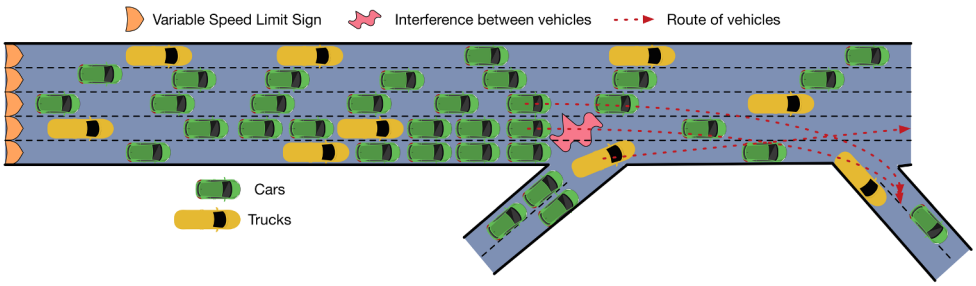



**Fig. 2.** The VSL control example considered in this paper.

how this could be viewed as a Markov Decision Process (MDP).

### 3.1. Differential VSL among lanes

Variable Speed Limit (VSL) control enables the dynamic changing of posted speed limits in response to prevailing traffic. The objective of VSL control is to maximize transportation network performance, including efficiency, safety, and environmental friendliness. From the perspective of traffic flow theory, VSLs can slow down traffic streams and change inflows to bottlenecks. Therefore, VSLs can maintain bottleneck traffic operating near its capacity state (Khondaker and Kattan, 2015).

The VSL control example considered in this paper is illustrated in Fig. 2. The freeway section in Fig. 2 is composed of five lanes and presents an on-ramp and an off-ramp. As may be seen in the figure, the interference between vehicles is occurring in the merging area between the on-ramp and the mainstream. The vehicles, which are exiting the freeway, move into the right lanes in the upstream of the on-ramp. In this case, the interference between mainstream and on-ramp vehicles causes further speed reductions in the merging area, contributing to the creation of a generalized bottleneck. Whereas the left lanes of the freeway is in the free-flow conditions when the bottleneck is formed. Traditionally, a homogeneous speed limit across all lanes would be posted to control the outflow of the controlled section and to prevent the capacity drop at active bottlenecks. In this paper we study how to post different speed limits across lanes. It is natural and beneficial to do that. For example, there might be a short period when the merging area between the on-ramp and mainstream is congested, whereas the left lanes are in free-flow conditions. In such cases, it is not necessary to reduce the outflow of all 5 lanes. On the contrary, a lower speed limit for the left-side lanes would degrade the efficiency of the freeway.

It should be noted that the DVSL control among lanes has not been well researched until now. One reason may be that drivers are not familiar with the VSL system. In a case study of Germany, it is shown that the DVSL control motivates drivers to change into the fastest lanes, therefore causes traffic breakdown and increase in crash probability (Schick, 2003). The aim of this paper is to explore a modern VSL solution under CAV environment. In CAV environment, it is not difficult to implement the DVSL control action by sending speed limit orders to the vehicles in the corresponding lane. Even the existing driver assistance systems e.g. fixed speed cruise control can be used to enforce the vehicle to drive with speed less than the received speed limit.

### 3.2. Formulation as a markovian decision process

To relate the DVSL control to the RL setting, the control problem should be formalized as a Markov decision process (MDP), which involves trial-and-error interaction in an environment. An MDP consists of $(S, A, P, \mathcal{R}, \gamma)$, where $S$ denotes the state space, $P$: $S \times A \times S$ is the transition probability, $R$: $S \times A \times S \to M(R)$ is the reward distribution, and $\gamma \in [0, 1)$ is the discount factor. At each discrete time step $t = 1, 2, 3, \cdots$ the agent selects an action according to some policies $\pi$ and the environment responds by transitioning into a new state $s_{t+1}$ sampled from $p(:|s_t, a_t)$, and the agent receives a scalar reward $r_{t+1}$. The agent's goal is to maximize the discounted cumulative sum of rewards, from the current state $s_t$, for some discount factor $\gamma < 1$. The formulations of VSL control as an MDP can be found in Li et al. (2017), Zhu and Ukkusuri (2014). In this paper, we extend the formalism to DVSL control. The definitions of agent, state, action, transition, and reward function are given as follows:

**Agent**: We consider a VSL controller as an agent. An agent can set different speed limits for each lane of its defined section. For a transportation network or a freeway corridor, there will be a large number of available VSL controllers. In such case, the VSL control problem can be formulated as a multi-agent RL problem. In this paper, we only consider the DVSL control with a single agent. The goal of the agent is to improve the efficiency, safety, and emissions of its own section in the presence of a recurrent downstream bottleneck.

**State** $s_t \in S$: State is a measure of the real time traffic environment, or the evolution of traffic flow. Due to the complexity of the dynamics of traffic flow, it is quite difficult to obtain a state representation that describes precisely how traffic may change from one state to another. The state variables can be any traffic parameters related to the controlled section that is reported by any sensors, such as by loop detectors or by probe vehicles. As it is suggested in Li et al. (2017), the traffic state at the immediate downstream of the merging area, the upstream mainline section, and the on-ramp should be considered by the VSL controller. In this paper, the state of the VSL controller agent is defined as $s_t \in R^{m_l+u_l+o_l}$, where $m_l$, $u_l$ and $o_l$ are the number of lanes of the merging area, the upstream mainline, and the on-ramp correspondingly. The occupancy rates reported by the loop detectors are used as state variables.

**Action** $a_t \in A$: An action interacts with the speed limit of all lanes at time $t$. Therefore $a_t \in R^{c_l}$, where $c_l$ is the number of lanes at the controlled section. Considering real world implementation challenges and driver compliance issues, the elements of $a_t$ are set as discrete values $a_t^i \in [0, 1, 2, ..., M-1]$, and the speed limits $V_t \in R^{c_l}$ is equal to $V_0 + Ia_t$, where $V_0 \in R^{c_l}$ is the minimum value of the speed limit, $I$ is the integer multiples, the maximum value of speed limits is $V_0 + I(M-1)$. For a section with multiple lanes, the dimension of the action space would become very high, thereby increasing the difficulty of learning. For example, for a 5-lane freeway section, suppose that we can choose 10 kinds of speed limit for each lane, the number of discrete speed limits option could be $10^5$. The learning algorithm with discrete action space (e.g.. Deep Q learning) have to estimate the value function of all possible choice, which makes them not suitable for the DVSL control problem. The learning strategy of such an action space will be given in the next section.

**State transition probability** $p(s_{t+1}|s_t, a_t)$: The training of an agent is conducted on a simulation platform SUMO[1]. SUMO provides flexible APIs for network design, traffic sensors, and traffic control solutions. The transition from $p(s_{t+1}|s_t, a_t)$ is implicitly defined by SUMO, and the cars in the simulation.

**Reward** $r_t \in \mathcal{R}$: The goal-directed or hedonistic behaviour is the foundation of reinforcement learning (RL), which is learning to choose actions that maximize the given reward signal. The key issue in this approach is to ensure that the agents receive rewards that promote good system level behavior. Defining a reward function $r_t$ for the VSL control problem is not obvious. The optimized objective of the VSL control can be total travel time, low crash probability, minimal vehicular emissions, etc. In order to improve the efficiency of the freeway section, the reward function cannot be straightforwardly defined as average travel time, because the travel time of the vehicles cannot be computed until they have completed their routes, which leads to the problem of extremely delayed rewards. Fortunately, it is known that there is a direct relationship between the total travel time and the inflow and outflow of a traffic network (Papageorgiou and Kotsialos, 2002). The relationship is given as follows:

$$TTS = KTn_0 + K\left(K-1\right)\frac{T^2}{2}\left(F^{in} - F^{out}\right), \tag{1}$$

where $K$ is the total time steps, $T$ is the time interval, $F^{in}$ is the total inflow of the transportation network, and $F^{out}$ is the total outflow of the network. Obviously, $F^{in} = \sum_{t=0}^{\infty} f_t^{in}$ and $F^{out} = \sum_{t=0}^{\infty} f_t^{out}$. Therefore, the reward $r_t^0$ associated with total travel time can be defined as $r_t^1 = f_t^{out} - f_t^{in}$. The total outflow $f_t^{out}$ and inflow $f_t^{in}$ at time point $t$ can be easily collected from the loop detectors located on the upstream mainline/on-ramp and the downstream mainline/off-ramp. Li et al. (2017) suggested another metric related to freeway efficiency. They used the traffic condition of a bottleneck as a reward function. The second reward function was considered to be the average velocity reported by detectors at the bottleneck $r_t^2 = avg(vel_t^{bottleneck})$.

Another objective of VSL control is to reduce crash probability. With SUMO APIs, we can obtain the acceleration of vehicles. The reward $r_t^3$, which is related to the safety of the DVSL controlled section, is defined as $r_t^3 = -\theta_t$ where $\theta_t$ is the number of emergency braking vehicles (deceleration is above 4.5 m/s$^2$) in the last step. Meanwhile, we are also interested in measuring the emission reductions resulting from implementing the DVSL control. SUMO provides flexible APIs to calculate the CO, HC, NOx and PMx emissions. We can use the emission standards to obtain an emission reward $r_t^4$,

$$r_t^4 = -\left(\frac{e_t^{CO}}{1,\ 5} + \frac{e_t^{HC}}{0.13} + \frac{e_t^{NOx}}{0.04} + \frac{e_t^{PMx}}{0.01}\right) \tag{2}$$

where $e_t^{CO}$, $e_t^{HC}$, $e_t^{NOx}$ and $e_t^{PMx}$ are the total CO, HC, NOx and PMx emissions of the freeway section, the units of the emission are $kg$. 1.5, 0.13, 0.04 and 0.01 respectively. Those units are set according to the Euro VI[2] standards for CO, HC, NOx and PMx emissions. The usages of four reward functions $r_t^1$, $r_t^2$, $r_t^3$, and $r_t^4$ will be studied in the experiments of Section 5.

The RL solutions for aforementioned MDP process of DVSL are challenged by three key facts. First, different speed limits among different lanes form a large discrete action sets, which making it difficult to learn using discrete DRL methods like DQN. Second, DVSL control have multiple possibly conflicting objectives. Because it is typically not clear how to evaluate available trade-offs between different objectives, there is no single optimal policy. Third, the solutions should be able to achieve an optimal control scheme under uncertainty traffic demand and driving behavior.

## 4. Deep reinforcement learning for differential variable speed limit

In this section, we present the actor-critic architecture for DVSL solutions. The basic framework of the architecture is given in Fig. 3. This architecture avoids the heavy cost of evaluating all the sets of different speed limits among the lanes. This policy builds upon the actor-critic framework. The actor is used to generate an action for the VSL control, and the critic is utilized to evaluate the actor's policy. The estimated Q value of the critic is related to the efficiency, safety, and emission reduction ability of the transportation network. Multi-layer neural networks are used as function approximators for both the actor and critic functions.

### 4.1. Action generation

The architecture reasons over actions within a continuous space $R^{c_l}$, and then simply uses the integer conversion to generate a

[1] http://sumo.dlr.de

[2] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32012R0459.

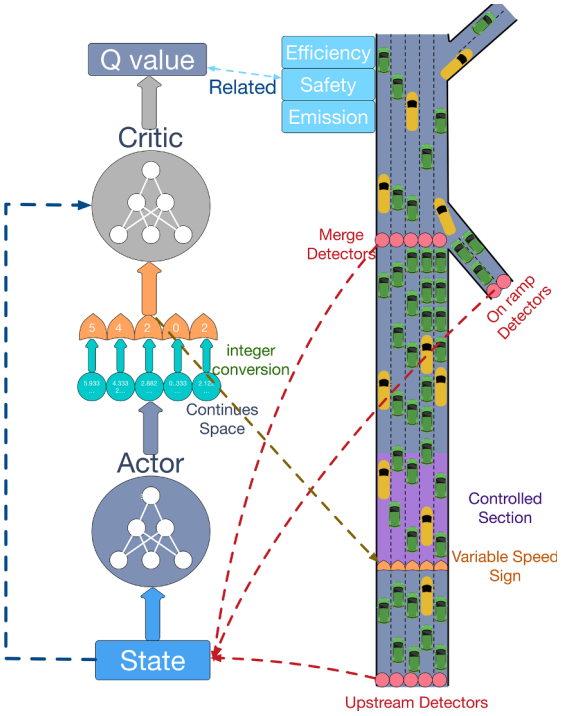



**Fig. 3.** The actor-critic architecture for DVSL.

discrete action $a_t$. The speed limits can be obtained by $V_0 + Ia_t$. The input state $s_t \in R^{m_l+u_l+o_l}$ of the actor is the occupancy rates collected from the upstream, merging area, and on-ramp detectors. The actor can be defined as the following:

$$f_{\theta^\pi}: R^{m_l+u_l+o_l} \rightarrow R^{c_l},$$
$$f_{\theta^\pi}(s_t) = \widehat{a_t}. \tag{3}$$

$f_{\theta^\pi}$ is a function parameterized by $\theta^\pi$, mapping from state space $R^{m_l+u_l+o_l}$ to action space $R^{c_l}$. As stated before, a continuous speed limit is not feasible to post in a variable speed sign. As a result, the speed limits calculated from $\widehat{a_t}$ would not be a valid one. Suppose that there are $M$ kinds of speed limits for each lane; in this case, we need to map $\widehat{a_t}$ to an element in the valid action set $A$. The proposed mapping strategy is given as following:

$$g: R^{c_l} \rightarrow A,$$
$$g(\widehat{a_t}) = int(clip(\widehat{a_t}, 0, M)). \tag{4}$$

$g$ is a mapping from a continuous space to a discrete set. It first clips the values of $\widehat{a_t}$ into $(0, M)$, then the discrete action $a_t$ can be easily obtained by the integer parts of clipped $\widehat{a_t}$. It guarantees the values of $a_t \in (0, 1, ..., M-1)$. The differing speed limits among lanes can be then calculated by $V_t = V_0 + a_t I$.

The critic function is used to evaluate the action representation, and estimate the value function $Q_{\theta^Q}$ of choice of actor:

$$\pi_\theta(s) = \text{argmax}_{a \in g(f_{\theta^\pi}(s))} Q_{\theta^Q}\left(s, a\right), \tag{5}$$

where $Q_{\theta^Q}$ is the estimated value function parameterized by $\theta^Q$, $\theta^Q$ represents the parameter of the critic, and $\theta^\pi$ represents the parameter of the actor. The goal of the actor for VSL control is to maximize the value function $Q$.

$$Q\left(s_t, a_t\right) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k}\left(s_t, a_t\right)\right] = r_t + \gamma^k Q\left(s_{t+1}, a_t\right), \tag{6}$$

where $Q$ is the actual value function, and $r_t$ is the reward function, which is depending on the reward selection for the VSL control system. The goal of the critic is to produce a perfect approximation of the value function. For a complex MDP, the true value function cannot be obtained until a large number of policy have been tried. But it can be learned by bootstrapping from the current estimate of the value function. The parameters $\theta^Q$ of the critic can be updated by the temporal difference (TD) error signal:

$$\theta^Q = argmin_{\theta^Q} Q_{\theta^Q}(s_t, a_t) - (r(s_t, a_t) + Q_{\dot{\theta}^Q}(s_{t+1}, \pi_\theta(s_{t+1}))), \tag{7}$$

where $Q_{\theta^Q}(s_t, a_t) - (r(s_t, a_t) + Q_{\dot{\theta}^Q}(s_{t+1}, \pi_\theta(s_{t+1})))$ is the estimated TD error signal.

### 4.2. Training with deterministic policy gradient

The training algorithm's goal is to find a parameterized policy $\pi_\theta$ which maximizes its expected reward return over the DVSL controlled time period. $\pi_\theta$ is characterized by the actor $g(f_{\theta^\pi})$. We perform the training using the Deep Deterministic Policy Gradient (DDPG) algorithm (Lillicrap et al., 2015). The core of the DDPG is to use a stochastic behavior policy for exploration, but to estimate a deterministic target policywhich means that the final estimated $\pi_\theta(s_t)$ will produce a deterministic action $a_t$ rather than a stochastic one. The DDPG optimizes the parameters of actor-critic in a bilevel optimization manner, and the loss function $L(Q, \theta^Q)$ for the critic is:

$$\begin{aligned} y_i &= r(s_i, a_i) + Q_{\dot{\theta}^Q}(s_{i+1}, \pi_{\dot{\theta}^\pi}(s_{i+1})), \\ L\left(Q, \theta^Q\right) &= \frac{1}{N}\sum_i (y_i - Q_{\theta^Q}(s_i, a_i))^2. \end{aligned} \tag{8}$$

Here $N$ is the number of samples, with the $i$ index referring to the $i$th sample. $y_i$ is the $i$th label, computed from the sum of the immediate reward and the outputs of the target actor and critic networks, having weights $\dot{\theta}^\pi$ and $\dot{\theta}^Q$ respectively. Then the critic loss $L(Q, \theta^Q)$ can be computed. The weights $\theta^Q$ of the critic network can be updated with the gradients obtained from the loss function in Eq. (8). The goal of the actor is to optimize the loss function:

$$L\left(\pi, \theta^\pi\right) = -\frac{1}{N}\sum_i Q\left(s_i, \pi_{\theta^\pi}(s_i)\right). \tag{9}$$

In order to update the weights $\theta^\pi$, we can use the gradient:

$$\nabla_{\theta^\pi} = \frac{1}{N}\sum_i \nabla_a Q_{\theta^Q}\left(s_i, a\right)\Big|_{a=\pi_{\theta^\pi}(s_i)} \nabla_{\theta^\pi}\pi_{\theta^\pi}(s_i). \tag{10}$$

The detail of the deterministic policy gradient in Eq. (10) can be found in Silver et al. (2014), it has been proven that the deterministic policy gradient in Eq. (10) is equivalent to the stochastic policy gradient. If we take $g$ as parts of the actor, the architecture of the actor will not be fully differentiable. However, $g$ can be considered as a function of the VSL signs. The VSL signs can take the action generated from $f_{\theta^\pi}$, and use $g$ to produce feasible speed limits.

Traditional RL agents incrementally sample the experience including state $s_i$, $s_{i+1}$, action $a_i$, and reward $r_i$, updating their parameters and then discarding these experiences immediately. This approach causes strong temporal correlations between samples and rapid forgetting of possibly useful experiences. Experience replay (Lin, 1992; Mnih et al., 2015) addresses both of these problems by storing experience into a replay memory. The experience are constantly sampled from the replay memory to update the agents, this process stabilizes the training of the neural networks for DRL. In this paper, we apply priority experience replay as proposed in Schaul et al. (2015), to sample experience to update the actor weights $\theta^\pi$ and critic weights $\theta^Q$, in which the probability $p_i$ of transition $(s_i, a_i, r_i, s_{i+1})$ being sampled from replay memory is:

$$p_i = \frac{1}{rank(i)}, \tag{11}$$

where $rank(i)$ is the rank of transition $i$ when the replay memory is sorted according to the absolute value of TD error signal $\|\delta_i\|$. The central idea is that an RL agent can learn more effectively from certain transitions. The transitions with high absolute TD errors are more valuable and surprising than other transitions.

A core challenge in RL is how to balance exploration-actively seeking out actions that might yield high rewards and lead to long-term gains; and exploitation–maximizing short-term rewards using the agent's current knowledge. Without adequate exploration, the agent might fail to discover effective DVSL control strategies. One advantage of DDPG, as an off-policy RL framework, is that its exploration can be independent from the learning algorithm. The exploration is done by adding noise $x$ sampled from a noise process to $f_{\theta^\pi}(s_t)$. In the experiments, the noise $x$ is modeled as a Laplacian process $L\left(x\middle|b\right) \sim \frac{1}{2b_t} exp\left(-\frac{x}{b_t}\right)$. The parameter $b_t$ is decayed with respect to the learning time. The algorithm for the proposed framework is summarized in algorithm.1.

**Algorithm 1.** DVSL control agent training with DDPG.

1: Set a reward function that is chosen from $r^1$, $r^2$, $r^3$, $r^4$ and the integer multiples $I$ for DVSL.
2: Randomly initialize critic network $Q_{\theta^Q}$ and actor network $\pi_{\theta^\pi}$ with parameters $\theta^Q$ and $\theta^\pi$;
3: Initialize target weights: $\dot{\theta}^\pi \rightarrow \theta^\pi$, $\dot{\theta}^Q \rightarrow \theta^Q$.
4: Initialize replay memory.
5: **for** *episode* = 1 to $m$
6: Start the traffic simulation with SUMO.
7: Initialize a random process $L\left(x\middle|b\right) \sim \frac{1}{2b_t} exp\left(-\frac{x}{b_t}\right)$ for action exploration.
8: Recieve initial observe state $s_1$ from the loop detectors in SUMO.
9: **for** $t$ = 1 to time length of the traffic simulation $T$

Select action $a_t = g(f_{\hat{\theta}^\pi}(s_t) + x_t)$ according to the current policy and exploration method.
Decay the noise parameter $b_t$.
Execute DVSL with speeds $V_0 + I * a_t$ and observe reward $r_t$, new state $s_{t+1}$ from the SUMO simulation.
Store $(\{t, s_t, a_t, r_t, s_{t+1}\})$ in the replay memory.
Sample a random minibatch of $k$ transitions $(\{i, s_i, a_i, r_i, s_{i+1}\})$ from the replay memory using probility $\frac{1}{rank(i)}$;
Update the critic by minimizing the loss $L(Q, \theta^Q)$ in 8.
Update the actor $f_{\theta^\pi}$ by sampled gradient given in 10.
Update transition priority $p_i$ according to the TD error $\delta_i$ computed by the critic
Update the target network:

$$\hat{\theta}^\pi \leftarrow \tau\theta^\pi + (1-\tau)\hat{\theta}^\pi,$$
$$\hat{\theta}^Q \leftarrow \tau\theta^Q + (1-\tau)\hat{\theta}^Q$$

**end for**
**end for**

## 5. Simulation framework

### 5.1. Traffic network in SUMO

The goal of this study is to evaluate the usage of deep reinforcement learning in DVSL. The open source software SUMO is selected for the experiments. This software is highly flexible, well documented, and supports setting the speed limits for each lane using its API–the Traffic Control Interface (TraCI) package. Most previous applications of RL to traffic control use SUMO as their simulation environment (Van der Pol and Oliehoek, 2016; Mousavi et al., 2017; Wei et al., 2018; Chu et al., 2019). Following this fashion, we also use SUMO as the simulation environment for this study. In order to seamlessly operate in the real world the DRL agents need to transfer the learning it does in simulation. Currently, this is a non-trivial task as traffic simulation environments are often simplistic and lack the richness or diversity of the real world. In other words, there is no simulation platform that can perfectly replicate the real world traffic dynamics. SUMO, as a microscopic simulation platform, is more diverse and meticulous than macroscopic simulation models that are extensively used in previous studies (Li et al., 2017) on VSL. With SUMO, we can define various kinds of car following and lane changing models for the vehicles. In order to evaluate the generalization of our model, we will train our model with some fixed car following and lane changing parameters, and then evaluate it on environments with different parameters. In addition, we have try our best to make the simulation environment more complex, and evaluate the DRL agents in a stochastic environment.

An 875.51 m, on- and off-ramp inclusive, northbound freeway section of I405 in California, USA was selected. It should be noted that the traffic network can be any freeway section with on- and off- ramps. The reason we choose this particular freeway section was because of its simple structure. Moreover, we could use data from an open dataset–California PeMS[3] to generate demand data for the simulation. A map of the selected study area was first exported from OpenStreetMap.org. Next, a traffic network for simulation was built based on the map. We used the netconvert package to convert the map into SUMO.net file. Then we deleted the irrelevant roads from the.net file. The original speed limit for the mainlane of the section was $65 mile/h$, and for both the on- and off- ramps were 50 mile/h.

Next, we needed to generate travel demand data for the simulation. For the section in Fig. 4, we needed to consider only three route choices: (1) From mainlane to mainlane (M2M), (2) From mainlane to off-ramp(M2Off), and 3) From on-ramp to mainline (On2M). The stations with IDs 717727, 771898, and 714847 of PeMS are in a freeway section. The travel demand is set based on observations of traffic flow recorded by the detectors of these stations. Each simulation round lasts for 5 h from 5:00 am to 10:00 am. The number of vehicles per hour, within three routes, is modeled as a Poisson process. The average value of the Poisson distribution for each of the three routes is given in Table 1. It should be noted that the drivers would not make dynamic route choice, which means that the number of vehicles with those three routes are fixed in each round simulation. The departure lane of the vehicles is randomly set according to uniform distribution. Passenger cars with a length of 3.5 meters and trucks/buses with a length of 8 meters are selected as vehicle types in the simulated traffic stream. Specifically, the traffic consisted of 85% passenger vehicles and 15% trucks and buses. The demand is randomly generated for each round simulation. In order to make the simulation more complex, we randomly set half of the vehicle with the “Krauss” car following model, and another half of the vehicle with the “IDM” car following model. The default “LC2013” model of SUMO is used as the lane change model for all vehicles.

Fig. 5 gives the average speed of the merging area during three simulation rounds without VSL control. It is found that the merging area speed are very low in the simulation period. The average speed is 25 mile/h, which is far lower than the speed limit.

The simulation is used to train and evaluate deep reinforcement learning agents for VSL control. For VSL control, the longitudinal and lateral driving behaviors have great impacts on the VSL performance. There are various applications that can be used to define vehicle behaviors for SUMO. In this paper, we consider two important properties of the drivers. The first one is the desired longitudinal driving speed with respect to speed limits. It can be defined by the “speedFactor” variable, which is the vehicles expected multiplicator for lane speed limits. The attribute “speedFactor” can also be defined as a normal distribution. The speedFactor for

[3] http://pems.dot.ca.gov.


(a) The I405 northbound section in openstreetmap.

(b) The I405 northbound section, as well as its VSL controlled and merge areas in SUMO.

**Fig. 4.** The freeway sections in OpenStreetMap and SUMO.

**Table 1**
The average values of the Poisson distributions for 3 routes.

| | M2M | M2Off | On2M |
|---|---|---|---|
| 5;00–6:00 am | 3000 | 999 | 480 |
| 6;00–7:00 am | 5427 | 1809 | 1153 |
| 7;00–8:00 am | 4821 | 1608 | 1129 |
| 8;00–9:00 am | 5026 | 1676 | 1176 |
| 9;00–10:00 am | 4804 | 1602 | 1095 |

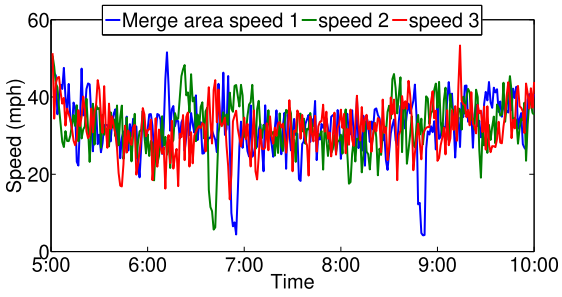



**Fig. 5.** The merge area speed in 3 round simulation.

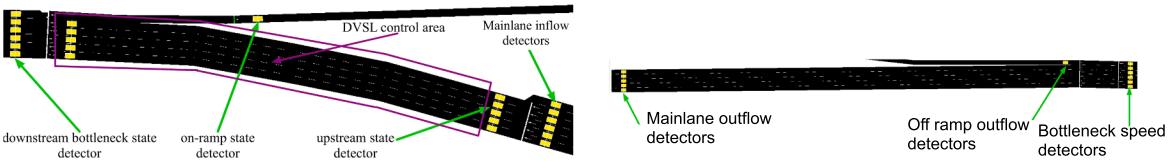
training DVSL agents is defined as "“normc(1,0.1,0.2,2)”, which means that the mean and standard deviation of the speedFactor is 1 and 0.1, its lowest and highest values are 0.2 and 2. The second one is the desire of the vehicle to perform lane change for obtain higher speed, which would encourage more lane changes. We defined a *speedgen* variable, which is corresponding to the “lcSpeedGain” attribute of the vehicle. The attribute “lcSpeedGain” is set as 1 for training VSL agents. A larger “lcSpeedGain” value means the vehicle is more likely to change lane for gaining high driving speed.

### 5.2. Measures of state, action and reward variables

To collect the state, action and reward variables we locate a large number of loop detectors in all links of the freeway shown in Fig. 6, and monitor the vehicles' speed and emission in the road networks. The TraCI methods of each measure are detailed in Table 2.

The state, action and reward calculation methods are detailed as following:

- The state variables $s_t$ are collected from the upstream state detectors, on-ramp detector and downstream bottleneck state detectors given in Fig. 6(a).
- $r_t^1$ can be measured from inflow collected from mainlane inflow detectors, on-ramp detector in Fig. 6(a) and outflow collected from mainlane outflow detectors, off-ramp detectors given in Fig. 6(b).
- $r_t^2$ can be measured by averaging the speed collected from bottleneck speed detectors given in Fig. 6(b).
- $r_t^3$ can be measured by summing the number of vehicle whose acceleration is below $-4.5$ m/s.
- $r_t^4$ can be measured by the emission collected from TraCI.




(a) The detectors and DVSL control area in upstream.

(b) The detectors in downstream.

**Fig. 6.** The loop detectors and DVSL control area.

**Table 2**
SUMO TraCI methods for evaluated measures and related calculation method.

| Measures | SUMO TraCI method | Aggregation method |
|---|---|---|
| $s_t$ | traci.inductionloop.getLastStepOccupancy() | Mean over time steps |
| $a_t$ | traci.lane.setMaxSpeed() | / |
| $r_t^1$ | traci.inductionloop.getLastSteoVehicleNumber() | Sum over time steps |
| $r_t^2$ | traci.inductionloop.getLastStepMeanSpeed() | Mean over time steps |
| $r_t^3$ | traci.vehicle.getSpeed() | Sum over time steps |
| $r_t^4$ | traci.edge.getCO(HC,NOx,PMx) Emission() | Sum over time steps |

Our simulation environment is open source and available online.[4]

### 5.3. Action space and agent parameters

The proposed DVSL control strategy adjusts speed limits from 40 to 75 mph with an increment of 5 mph. The action set for each lane is given by [40,45,50,55,60,65,70,75] mph. There are eight options for each lane, therefore the number $M$ of $g$ in Eq. (4) is set as 8. The state dimension for the agent is 12. There are six lanes upstream, five lanes in the downstream merging area, and one lane on the on-ramp. The actor and the critic of the agent are composed of neural networks. Specifically, the actor $f_{\theta^\pi}$ is expressed by:

$$\begin{aligned} h_{t_1}^\pi &= relu(W_1^\pi s_t + b_1^\pi), \\ h_{t_2}^\pi &= relu(W_2^\pi h_{t_1}^\pi + b_2^\pi), \\ h_{t_3}^\pi &= relu(W_3^\pi h_{t_2}^\pi + b_3^\pi), \\ a_t &= M \times sigmoid(W_4^\pi h_{t_3}^\pi + b_4^\pi). \end{aligned} \tag{12}$$

where *relu* and *sigmoid* are nonlinear activations. $W_1^\pi \in R^{200\times 12}$,$b_1^\pi \in R^{200}$, $W_2^\pi \in R^{100\times 200}$, $b_2^\pi \in R^{100}$, $W_3^\pi \in R^{50\times 100}$, $b_3^\pi \in R^{50}$, $W_4^\pi \in R^{5\times 50}$ and $b_4^\pi \in R^5$ are the parameters $\theta^\pi$ for the actor. The critic $Q_{\theta^Q}$ is expressed by:

$$\begin{aligned} h_{t_1}^Q &= relu(W_s^Q s_t + W_a^Q a_t + b_1^Q), \\ h_{t_2}^Q &= relu(W_2^Q h_{t_1}^Q + b_2^Q), \\ h_{t_3}^Q &= relu(W_3^Q h_{t_2}^Q + b_3^Q), \\ Q_t &= (W_4^Q h_{t_3}^Q + b_4^Q). \end{aligned} \tag{13}$$

where $W_s^Q \in R^{200\times 12}$, $W_a^Q \in R^{200\times 5}$, $b_1^Q \in R^{200}$, $W_2^Q \in R^{100\times 200}$, $b_2^Q \in R^{100}$, $W_3^Q \in R^{50\times 100}$, $b_3^Q \in R^{50}$, $W_4^Q \in R^{1\times 50}$ and $b_4^Q \in R^1$ are the parameters $\theta^Q$ for the critic. During training, the noise parameter $b_t$ for exploration is set to 2.5, and decays 99.999% in each step. The action dynamically changes per 1 min, which means that the speed limits of the controlled area change every minute.

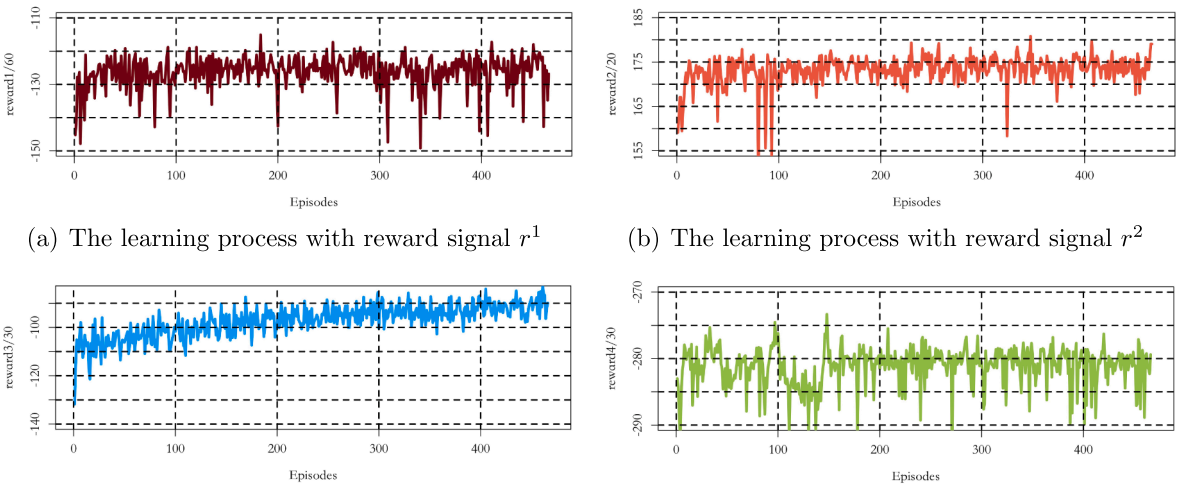
The time costs of the environment and agents are very important for training DRL agents. SUMO is a microscope simulation software, in which each vehicle's dynamic needs to be calculated and recorded. The neural networks use a light-weight structure. Therefore the time costs are concentrated to SUMO simulation. A 5 h episode of SUMO simulation without DRL agents takes average 104.7375 s, and the one with DVSL control only takes 107.1425 s. The training process involves both weights update by back-propagation and neural networks execution takes 109.5522 s.

## 6. Experimental results

### 6.1. Comparison between different rewards

To compare the benefits of different reward signals, the DVSL agents are firstly trained on SUMO with rewards $r^1$, $r^2$, $r^3$ and $r^4$. The travel demands for the training simulation are fixed, which means that agents with different rewards are trained by one simulation with the same configuration. Each agent is trained by 470 episodes of simulation, and then we conduct a comparison between them. The learning processes of DVSL-$r^1$, DVSL-$r^2$, DVSL-$r^3$, and DVSL-$r^4$ are given in Fig. 7. It is observed that the learning processes of different reward signals exhibit significantly different patterns. Obviously, the safety reward signal $r^3$ is the easiest one to train, as the curve in Fig. 7(c) grows steadily during the training process. In Fig. 7(b), the reward signal $r^2$ also reaches a stable value, whereas the reward signals $r^1$ and $r^4$ are more difficult to converge. Several oscillations can be observed in Fig. Fig. 7(a) and (d). The reason is related to the MDP formalization of the DVSL problem. The reward signal $r^3$ is computed by the overall number of emerging braking vehicles in the controlled section. The reward signal $r^2$ is the average velocity in the downstream bottleneck, which is also highly related to the speed limits of its compatriot upstream section. Conversely, the reward signals $r^1$ and $r^4$ are related to many other factors such as the inflow of on-ramp and outflow of off-ramp. The MDP process relies on an assumption of full observability, where a learning agent can observe environment states fully and the reward under a given state that can be fully controlled by the

[4] https://github.com/Kaimaoge/SUMO-DVSL.




(a) The learning process with reward signal $r^1$

(b) The learning process with reward signal $r^2$

(c) The learning process with reward signal $r^3$

(d) The learning process with reward signal $r^4$

**Fig. 7.** The learning process of the DVSL agents with different reward signals.

action of the learning agent. This assumption does not hold for an environment with reward signals $r^1$ and $r^4$, in which the agent can only partially control the rewards and observes only partial information related to the reward signals.

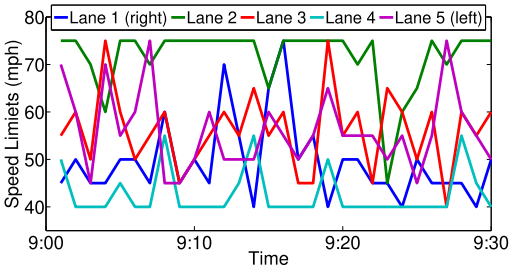
Table 3 compares the average performances of the DVSL agents, as trained by different rewards through different training episode indexes. It should be noted that we choose the networks with the best records for the corresponding rewards to conduct comparison. In our experiments, DVSL-$r^1$ achieves the best $r^1$ in the 181th step. DVSL-$r^2$, DVSL-$r^3$ and DVSL-$r^4$ give the best performance in the 338th, 462nd and 155th steps. The results show that most of the indexes of DVSL-$r^2$ are better than the ones of other agents. DVSL-$r^2$ is better than DVSL-$r^1$ in terms of $r^1$ and total traffic volume. The emission indexes of DVSL-$r^2$ are comparable to DVSL-$r^4$. The reason is that reward signal $r^2$ is easier to converge compared with $r^1$ and $r^4$. The DVSL-$r^3$ can reduce the number of emergency braking, but it decreases the efficiency of the freeway section. Its travel time and traffic volume indexes are even lower than the ones without a VSL controller. DVSL-$r^1$, DVSL-$r^2$ and DVSL-$r^4$ all achieve better performance compared with the one without a VSL, which indicates that a DVSL can improve the efficiency, safety, and emissions of the transportation systems.

Fig. 8 plots the speed limits of 5 lanes between 9:30 and 10:00 am, as given by different DVSL agents. All agents have learned to give differential speed limits among lanes. Although it is difficult to explain the reason why the agents driven by different reward signals give those differential speed limits, we can still observe several certain patterns from the speed limits. DVSL-$r^2$ and DVSL-$r^4$ achieve more desirable efficiency and emission indexes; and we can find that they set higher speed limits for right side lane1 and left side lane4 and lane5, and lower speed limits for middle lane2 and lane3. DVSL-$r^3$ always gives low speed limits for all 5 lanes; its highest speed limit in Fig. 8(c) is 60 mph. This indicates that DVSL-$r^3$ has learned that lower speed limits can bring safety benefits for the controlled section.

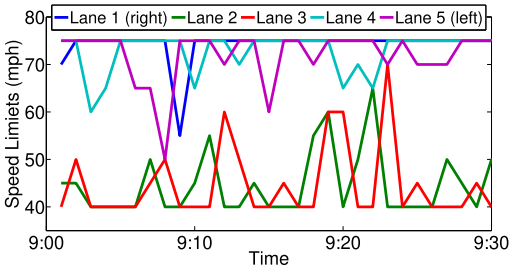
In order to better describe the behaviors of DVSL agents with different reward functions, we simulate an episode, in which there is no VSL control during 5:00 and 7:16 AM. Different DVSL agents start to control the freeway after 7:16 AM. The control lasts for 25 min. Then we compare the speed limits and performance criteria indexes in Fig. 9 and 10. All the agents share a same initial state given in Fig. 9(a), and lead to different speed limits and outcomes. It can be found that most of the lane's occupancy rate in merge bottleneck area is relatively low (lower than 0.15), which indicate that the merge area is not in congested state. However, the first 3 upstream lane's and on-ramp's occupancy rates are relatively higher (above 0.3). These inflow might bring future congestion. Four DVSL agents adopt significantly different speed limits to control the inflow. Their actions are given in Fig. 9. We can find that DVSL-$r^1$ always gives the leftest lane 5 with high speed limits, whereas the middle lane 2, 3 and 4 are set with lower speed limits. It mainly adjusts the inflow by the right lane 1. DVSL-$r^3$, which is designed to reduce emerging braking, gives lowest speed limits compared with other agents. The speed limits of DVSL-$r^2$ and DVSL-$r^4$ are more complex. To better understand the differential speed limits produced by different agents, we can pay more attention to the criteria performance given in Fig. 10.

**Table 3**
Average performance of different models on one episode of simulation. The best controller for each index are shown in boldface.

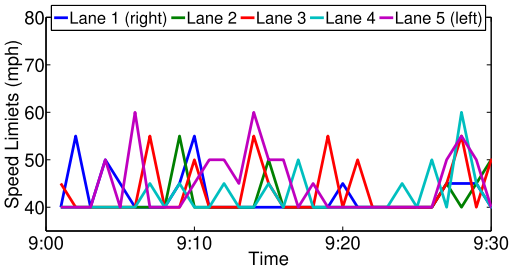
| method | $r^1(10^3)$ | $r^2(10^3)$ | $r^3$ $(10^3)$ | $r^4(10^7)$ | Co(kg) | HC(kg) | Nox(kg) | Pmx(kg) | ATT(s) | Volume(Veh) |
|---|---|---|---|---|---|---|---|---|---|---|
| NoVSL | -7.807 | 3.318 | −3.130 | −8.547 | 216.2 | 1.229 | 4.215 | 0.214 | 82.58 | 30479 |
| DVSL-$r^1$ | −7.689 | 3.363 | −2.937 | −8.546 | 214.6 | 1.223 | 4.256 | 0.215 | 81.69 | 30719 |
| DVSL-$r^2$ | **−7.568** | **3.490** | −3.021 | −8.416 | 210.9 | **1.203** | **4.194** | 0.213 | **79.20** | **30727** |
| DVSL-$r^3$ | −8.623 | 3.155 | **−2.201** | −8.665 | 221.0 | 1.250 | 4.240 | 0.213 | 87.70 | 29874 |
| DVSL-$r^4$ | −7.655 | 3.463 | −3.090 | **−8.412** | **209.1** | 1.204 | 4.197 | **0.213** | 80.15 | 30707 |



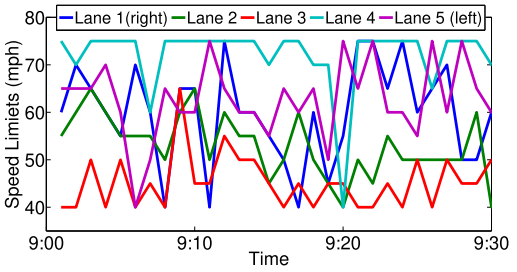

(a) The speed limits variations generated from DVSL-$r^1$




(b) The speed limits variations generated from DVSL-$r^2$

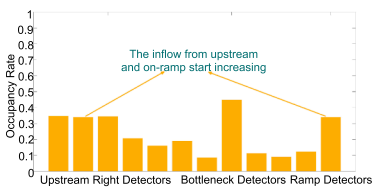



(c) The speed limits variations generated from DVSL-$r^3$

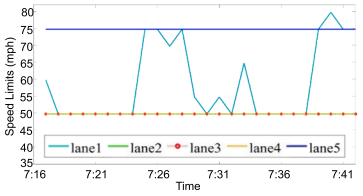



(d) The speed limits variations generated from DVSL-$r^4$

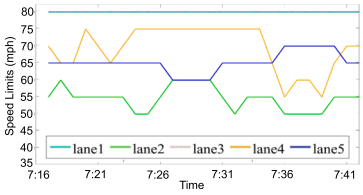
**Fig. 8.** Speed limit variations of different DVSL agents from 9:30 to 10:00 AM.



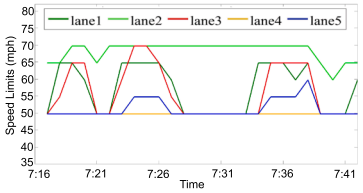

(a) The initial state at the time point that DVSL agents start control



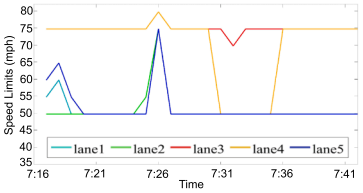

(b) The speed limits of DVSL-$r^1$ during control periods




(c) The speed limits of DVSL-$r^2$ during control periods



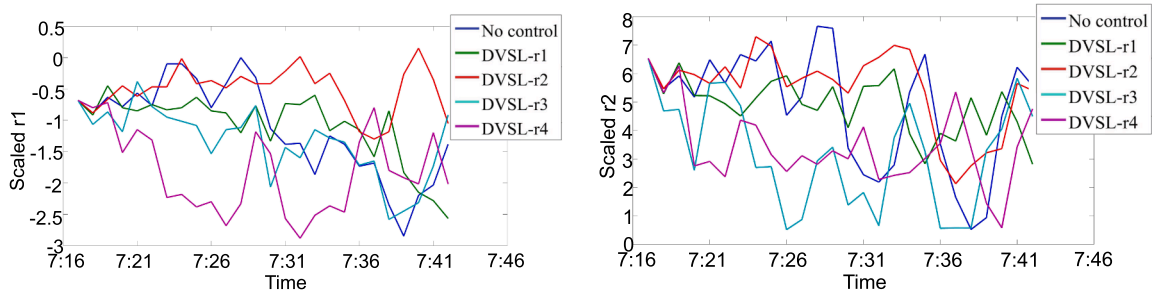

(d) The speed limits of DVSL-$r^3$ during control periods




(e) The speed limits of DVSL-$r^4$ during control periods

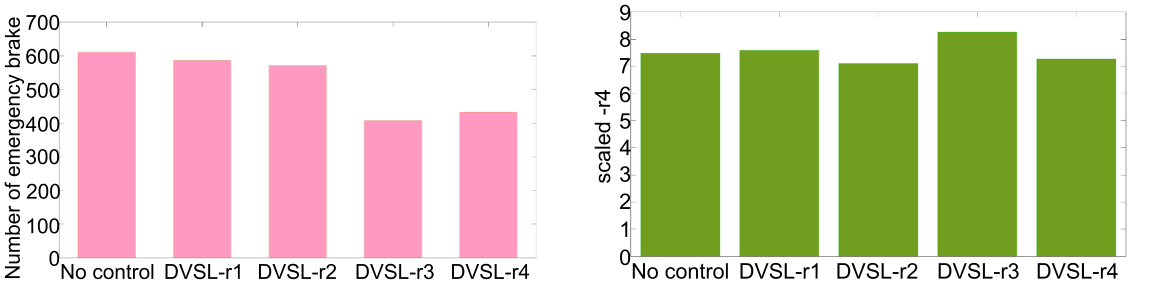
**Fig. 9.** Speed limit variations of different DVSL agents during control periods.

From Fig. 10(a) it is observed that the scaled $r^1$ values of DVSL-$r^2$ is higher than other methods. As stated before $r^1$ value is an index to measure throughput of the freeway section. In this case, only DVSL-$r^2$ avoids the dramatic throughput decrease after 7:30. The throughput index of DVSL-$r^1$ after 7:30 is better than the one without VSL control, but it is not comparable to DVSL-$r^2$. DVSL-$r^4$ fails to improve the throughput of the freeway in this case. DVSL-$r^3$'s scaled $r^1$ values are even worse than the one without VSL control. Similar phenomenon can be observed from Fig. 10(b), which can be used to measure the patency of the merge area. In this case, DVSL-$r^2$ is able to avoid future traffic congestion. The reason might be that the agent is easy to train with $r^2$ reward compared with $r^1$ reward. In this case the state-action trajectories are different with the one they trained with, $r^2$ reward might also enhance the agent's generalization ability. In Fig. 10(c), it is found that all 4 agents reduce the number of emergency braking. However, the improvements of DVSL-$r^3$ and DVSL-$r^4$ are larger than the ones of DVSL-$r^1$ and DVSL-$r^2$. It should be noted that the freeway's patency given by DVSL-$r^3$ and DVSL-$r^4$ are lower. It indicates that there may exist contradictions or conflicts between safety ($r^3$) and efficiency ($r^1$ and $r^2$) in a certain extent. The emission indexes (scaled $-r^4$) are given in Fig. 10(d), it can be found that DVSL-$r^2$'s emission is even better than DVSL-$r^3$, and DVSL-$r^1$ and DVSL-$r^3$ fail to reduce emission in this case. Contradiction between rewards can be also




(a) The scaled r1 values of different agents during control periods

(b) The scaled r2 values of different agents during control periods




(c) The overall number of emergency braking during control periods

(d) The overall scale emission indexes during control periods

**Fig. 10.** Comparison of different indexes during control periods.

observed from emission indexes.

### 6.2. Comparison with other models

In this section, the actor-critic based DVSL agent is bench-marked against the baseline scenario in which no control occurs at all, and baselines with actor-critic and DQN based VSL controllers:

- **NoVSL, baseline**: The baseline without any VSL. The vehicles are running with speed limits 65 mph in the mainline.
- Q learning, same speed limit for each lane: A similar model proposed by Li et al. (2017). The Q table size of the agent is $125 \times 6$. The average occupancy in upstream section, merge area and on-ramp over lanes are used to determine the states for the agent.
- **Deep Q networks (DQN), same speed limit for each lane (VSL-DQN)**: A deep version of Q learning. The state of the agent is as the same as the one in the DDPG models. The neural network structure of the DQN model is:

$$\begin{aligned} h_{t_1}^{QN} &= relu(W_s^{QN} s_t + b_1^{QN}), \\ h_{t_2}^{QN} &= relu(W_2^{QN} h_{t_1}^{Q} + b_2^{QN}), \\ h_{t_3}^{QN} &= relu(W_3^{QN} h_{t_2}^{Q} + b_3^{QN}), \\ Q_t &= (W_4^{QN} h_{t_3}^{Q} + b_4^{QN}). \end{aligned} \tag{14}$$

  where $W_s^{QN} \in R^{200\times 12}$, $b_1^{QN} \in R^{200}$, $W_2^{QN} \in R^{100\times 200}$, $b_2^{QN} \in R^{100}$, $W_3^{QN} \in R^{50\times 100}$, $b_3^{QN} \in R^{50}$, $W_4^{QN} \in R^{1\times 50}$ and $b_4^{QN} \in R^1$. Those parameters guarantee that the DQN model has a similar learning capability to the DDPG models, and the same learning capability as Q learning based models. Further details on learning DQN models can be found in Mnih et al. (2015).
- **Actor-critic (AC), same speed limit for each lane (VSL-AC)**: The difference between VSL-AC and DVSL is that the actor of VSL-AC only outputs one single action, and the action is transmitted into the speed limit for each lane of the controlled section.

All models are trained on 2000 episodes of simulation. The travel demands of each episode are generated from the random processes given in 5.1 (Each episode's demand is different from each other). There is a 50% chance that the freeway section will suffer from a traffic incident. The incident is simulated by randomly setting a vehicle to stop for 5–30 min. All agents are trained by a weighted sum of reward signals $r^1$, $r^2$, $r^3$ and $r^4$:

$$r^{overall} = w^1 \frac{r^1}{\eta^1} + w^2 \frac{r^2}{\eta^2} + w^3 \frac{r^3}{\eta^3} + w^4 \frac{r^4}{\eta^4}. \tag{15}$$

**Table 4**
Average performance of different models in episodes without incidents. The best controller for each index are shown in boldface.

| method | $r^1(10^3)$ | $r^2(10^3)$ | $r^3$ $(10^3)$ | $r^4(10^7)$ | Co(kg) | HC(kg) | Nox(kg) | Pmx(kg) | ATT(s) | Volume(Veh) |
|---|---|---|---|---|---|---|---|---|---|---|
| NoVSL | −8.718 | 3.257 | −2.837 | −8.684 | 222.9 | 1.259 | 4.221 | 0.215 | 85.82 | 30042 |
| Q learning | −7,633 | 3.422 | −2.833 | −8.493 | 231.2 | 1.243 | 4.083 | 0.214 | 83.69 | 30393 |
| VSL-DQN | −7.458 | 3.300 | **−2.832** | −8.582 | 216.7 | 1.232 | 4.251 | 0.215 | 83.26 | 30886 |
| VSL-AC | −7.870 | 3.305 | −2.873 | −8.585 | 215.2 | 1.224 | 4.253 | 0.215 | 83.52 | 30395 |
| DVSL | **−7.133** | **3.511** | −2.848 | **−8.568** | **214.6** | **1.224** | **4.209** | 0.215 | **80.82** | **30953** |

where $w^1 = 0.15$, $w^2 = 0.7$, $w^3 = 0$ and $w^4 = 0.15$. $\eta$s are used to scale the 4 reward signals to the same level. $w$s are the weight parameters for 4 reward signals. We give highest weights to $r^2$ because $r^2$ is easy to train and agents trained by $r^2$ can improve all aspects of the freeway section from the experimental results in 6.1. The weights for $r^3$ are set to 0 because $r^3$ is much easier to train when compared with other signals, and it would degrade other performance of the freeway section except the number of emergency braking in 6.1. $\eta^1$, $\eta^2$, $\eta^3$ and $\eta^4$ are used to scale different rewards final into similar ranges. Their values are calculated by the maximum values of their corresponding reward signals in the experiments. $\eta^1 = 80$, $\eta^2 = 30$, $\eta^3 = 40$ and $\eta^4 = 0.1$.Table 4 and 5 compare the average performance of the different methods across the 100 test episodes with and without incidents.

We highlight some of the key findings and present a summary of the experimental results here, concentrating on different indexes. With the actor-critic architecture, the DVSL agents are often successful in improving the efficiency and emission reductions of the freeway section. The improvements in episodes with incidents are higher than the improvements in episodes without incidents. For example, the DVSL improves 8.1% in average travel time in episodes with incidents and 5.8% in ones without incidents, whereas the VSL-DQN only improves 1.7% and 2.9%. The improvements of DVSL in episodes with incidents are higher than those in episodes without incidents. However, the VSL-DQN agent reduces more numbers of emergency brakings compared with the DVSL agent. This indicates that having a same speed limit is safer than having a differential speed limit among lanes, under the simulation environment. The VSL-DQN and Q learning are better than the VSL-AC. The reason is that the VSL-AC uses integer conversion to transmit continuous speed limits into discrete ones. The discretization process makes the policy of the VSL-AC harder to optimize and less efficient compared with the VSL-DQN. VSL-DQN is better than Q learning. This is because the neural networks give stronger learning capacity to the RL agents.

To understand what the DVSL agent has learned from the dynamic traffic condition, we show the agent's differential speed limits from 5:30 to 6:30 in one test episode in Fig. 11. During the time period, the traffic of the freeway section became congested. With the changing of traffic, an ideal VSL control method would be able to adjust its speed limits according to the traffic flow. In Fig. 11, the speed limits of all the lanes except that of lane 2 are in the maximum value before 6:00 am. After 6:00 am, the speed limits of lane 1 and lane 3 start to decrease, and the speed limit of lane 3 decreases to the minimum value. The DVSL agent has learned to always set a maximum speed limit for the left lanes, in other words it automatically set the left lanes as overtaking lanes. The agent mainly adjusts inflow to the bottleneck by adjusting the speed limits of the right lanes, because the right lanes probably contain more vehicles that are planning to leave the mainlanes, which would interfere with entering vehicles on the on-ramp.

### *6.3. Assessing generalization of DRL agents*

It is well known that traffic flow is a uncertain process, the uncertainty could lead to the failure of VSL control (Wang et al., 2012). Moreover, DRL has achieved breakthrough results on many tasks, but has been shown to be sensitive to system changes at test time (Packer et al., 2018). In this study, the DRL agents are trained purely in a simulation environment. The question then becomes: how can a DVSL agent utilize simulation to enable it to perform useful tasks in the real world? The challenge with traffic simulation is that even the best available simulators do not perfectly capture real traffic flow uncertainty and dynamics. The simulation environment used in this paper is covered with highly developed CAV techniques enabling vehicles driving with different speed limits among different lanes, whose testbed might be not available. Therefore we are not able to directly assess the significance for practical deployment. However we can assess the generalization of the DVSL agents using test environments with different attributes to the one the agent is trained with. The generalization ability gives us evidences on how good the agents are when they are on a different environment. The real-world implementation can be viewed as an environment with different attributes to the simulation one.

As stated before, the DVSL agents are trained with an environment with "speedFactor" value "normc(1,0.1,0.2,2)" and

**Table 5**
Average performance of different models in episodes with incidents. The best controller for each index are shown in boldface.

| method | $r^1(10^3)$ | $r^2(10^3)$ | $r^3$ $(10^3)$ | $r^4(10^7)$ | Co(kg) | HC(kg) | Nox(kg) | Pmx(kg) | ATT(s) | Volume(Veh) |
|---|---|---|---|---|---|---|---|---|---|---|
| NoVSL | −9.225 | 2.908 | −3.175 | −8.902 | 231.1 | 1.299 | 4.268 | 0.217 | 89.95 | 29751 |
| Q learning | −8,333 | 3.322 | −2.893 | −8.933 | 222.5 | 1.279 | 4.353 | 0.216 | 87.69 | 30467 |
| VSL-DQN | −8.261 | 3.300 | **−2.818** | −8.809 | 224.0 | 1.268 | 4.322 | 0.217 | 88.40 | 30401 |
| VSL-AC | −8.892 | 3.003 | −2.852 | −8.800 | 224.1 | 1.269 | 4.310 | 0.220 | 89.41 | 29856 |
| DVSL | **−8.059** | **3.452** | −3.099 | **−8.566** | **217.0** | **1.231** | **4.203** | **0.214** | **82.69** | **30509** |

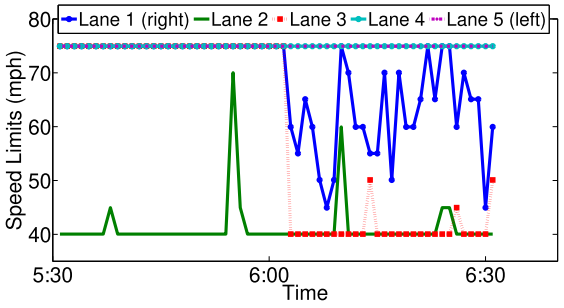
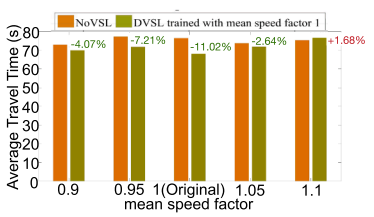



**Fig. 11.** The differential speed limits generated from DVSL agent.

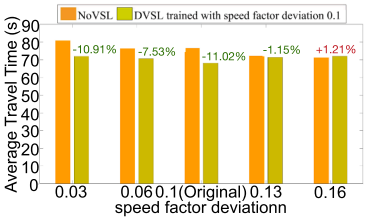
"lcSpeedGain" (*speedgen*) value 1. In order to assess the generalization of our algorithm, we evaluate the agents on environments with different mean and deviation values of "speedFactor" and different values of "lcSpeedGain". We evaluate the performance of all models on 30 episodes, in which the demand is fixed but the 3 driving behavior attributes are allowed to vary. In each round comparison, we only change 1 attribute and the others are fixed. The average travel time, CO2 emissions and $-r^1$ of the agents on these environments are given in Fig. 12.

We highlight some of the key findings and present a summary of the experimental results here. It is clear that deep RL agents are often successful in environments having same driving behaviors with the one they trained with. It reduced 11.07% and 6.46% travel time and CO2 emission, its overall $r^1$ is improved by 6.45% compared with the cases without VSL control. Its performance with respect to the varying attributes will be referred to in the following discussion.

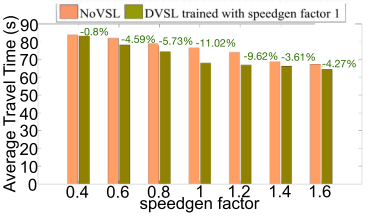
**Mean value of "speedFactor"**: It is evident that the emission and throughput of the freeway will degrade with higher mean value



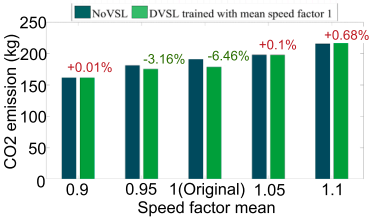

(a) The average travel time on environments with different mean value of speedFactor attributes



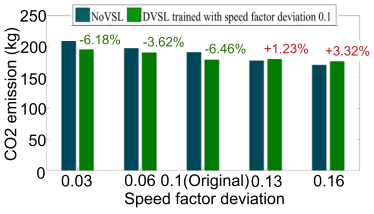

(b) The average travel time on environments with different deviation value of speedFactor attributes



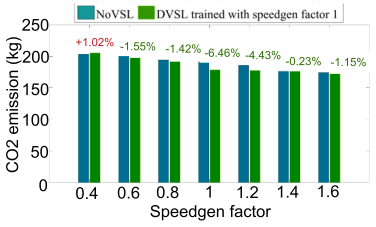

(c) The average travel time on environments with different *speedgen* values



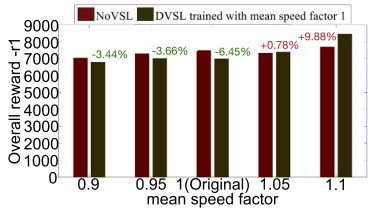

(d) The CO2 emission on environments with different mean value of speedFactor attributes



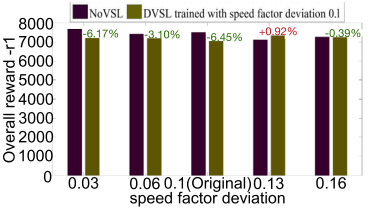

(e) The CO2 emission on environments with different deviation value of speedFactor attributes



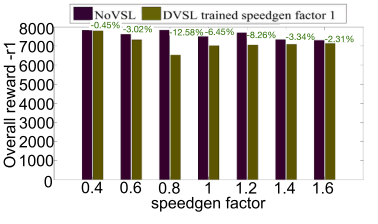

(f) The CO2 emission on environments with different *speedgen* values




(g) The overall $-r^1$ on environments with different mean value of speedFactor attributes




(h) The overall $-r^1$ on environments with different deviation value of speedFactor attributes




(i) The overall $-r^1$ on environments with different *speedgen* values

**Fig. 12.** The performance of the DVSL agents on environment with different driving behaviors to the training environments.

of "speedFactor" both with and without VSL control. We can observe that the CO2 emission and $-r^1$ increase with the mean value. It means that if the driver intend to drive faster than the speed limits, the freeway section will suffer from more serious congestion. The DVSL agent still improve the emission and efficiency when the mean value is equal to 0.95, and its average travel time and $-r^1$ are lower compared with the one without VSL control when mean value is equal to 0.95. However, it produces unsatisfactory performance when mean value equal to 1.1. The overall $-r^1$ value is even 9.88% higher than the one without VSL control under this case. The results indicate that the DVSL agent is less sensitive to the environment with lower desired longitudinal driving speed. However, it may fail when the real world driver has higher desired longitudinal driving speed than simulation drivers. In a highly-developed CAV environment, most vehicles will drive with speed below the required speed limits. Thus those findings indicate that the proposed method can be well adapted to an environment with high penetration rate of CAVs.

**Deviation of "speedFactor"**: The higher deviation will lead to higher throughput and lower emission for the freeway. The higher deviation value makes the driving speed more diverse, which would allow more gaps for merging vehicles, and improves the efficiency of merging area. The DVSL agents improves the average travel time, emission and throughput of the freeway when the deviation is 0.03 and 0.06. However, its performance is nearly equal to the case without VSL control when deviation varied to 0.13 and 0.16. In these cases, the congestion is relieved thus there might be limited space for VSL agents to improve. The average travel time is down from 76.5200 s to 71.1400 s when deviation increased from 0.1 to 0.16. The increased deviation slightly improves the average travel time of DVSL agents, which changes from 68.0900s to 72.0000 s. The results show that the efficiency of traffic will be improved by a more diverse driving style.

**"lcSpeedGain"**: The higher "lcSpeedGain" value is beneficial for the freeway. It means that the tendency of lane change for gaining speed is beneficial for relieving the congestion. It is observed that the generalization of DVSL agent is very well with respect to varying "lcSpeedGain". The DVSL agent's performance are all above the one without VSL control in nearly all the cases. However its improvements is relatively lower when the "lcSpeedGain" value is as low as 0.4. Under this case, the driver is not willing to perform lane changes, therefore the affects of differential speed limits among different lane will become small.

In summary, we observe that the generalization of DVSL agents is good in most cases. However, it is sensitive to part of attribute changes. The results expose some of the open questions and motivating future directions of research on implementing DRL agents to real world traffic control problems.

## 7. Conclusion

We develop a deep reinforcement learning framework for DVSL control. Our framework is built upon actor-critic architecture. Employing the actor-critc architecture of the proposed model, a large amount of discrete DVSL solutions can be efficiently learned in a continuous space. The experiments have shown that DDPG-based DVSL control models exhibit advantages in congestion alleviation, as well as accident and emission reductions in a simulation study. The reward engineering issue and generalization capability of the proposed method are also studied in this paper. Those are promising research directions toward bridging the reality gap for traffic control behaviors learned in simulation.

Future directions that could extend the scope of this study include:

- Combination of multi-agent DRL framework and extension to DVSL control for a larger transportation network and/or a long freeway with multiple control sections.
- The exploration of more powerful states and reward formulations.
- The future research on enhancing the generalization of the DRL agents.

A potential solution to enhance generalization is the combination of meta learning with deep reinforcement learning. Meta learning is an emerging machine learning area that allows agent to adapt to new environments with only a few trials. It is expected to make the agent to work in diverse environments, including ones that have never been trained before.

## CRediT authorship contribution statement

**Yuankai Wu:** Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Data curation, Resources, Investigation. **Huachun Tan:** Resources, Supervision, Project administration, Funding acquisition. **Lingqiao Qin:** Writing - review & editing, Validation. **Bin Ran:** Conceptualization, Resources, Supervision.

## Acknowledgement

The work was supported by IVADO and National Natural Science Foundation of China (61620106002 and 5170520).

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.trc.2020.102649.

## References


Abdel-Aty, M., Cunningham, R., Gayah, V., Hsia, L., 2008. Dynamic variable speed limit strategies for real-time crash risk reduction on freeways. Transp. Res. Rec.: J. Transp. Res. Board 2078, 108–116.

Belletti, F., Haziza, D., Gomes, G., Bayen, A.M., 2018. 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., Messmer, A., 2010. Optimal mainstream traffic flow control of large-scale motorway networks. Transp. Res. Part C: Emerg. Technol. 18 (2), 193–212.

Chu, T., Wang, J., Codecà, L., Li, Z., 2019. Multi-agent deep reinforcement learning for large-scale traffic signal control. IEEE Trans. Intell. Transp. Syst.

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

Hegyi, A., De Schutter, B., Hellendoorn, H., 2005a. Model predictive control for optimal coordination of ramp metering and variable speed limits. Transp. Res. Part C: Emerg. Technol. 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.

Hellinga, B., Mandelzys, M., 2011. Impact of driver compliance on the safety and operational impacts of freeway variable speed limit systems. J. Transp. Eng. 137 (4), 260–268.

Kattan, L., Khondaker, B., Derushkina, O., Poosarla, E., 2015. A probe-based variable speed limit system. J. Intell. Transp. Syst. 19 (4), 339–354.

Khondaker, B., Kattan, L., 2015. Variable speed limit: an overview. Transp. Lett. 7 (5), 264–278.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D., 2018. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int. J. Robot. Res. 37 (4–5), 421–436.

Li, L., Qu, X., Zhang, J., Li, H., Ran, B., 2018. Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network. Appl. Soft Comput.

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.

Lian, R., Peng, J., Wu, Y., Tan, H., Zhang, H., 2020. Rule-interposing deep reinforcement learning based energy management strategy for power-split hybrid electric vehicle. Energy 117297.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Lin, K., Zhao, R., Xu, Z., Zhou, J., 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. arXiv preprint arXiv:1802.06444.

Lin, L.-J., 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. Mach. Learn. 8 (3–4), 293–321.

Lin, P.-W., Kang, K.-P., Chang, G.-L., 2004. Exploring the effectiveness of variable speed limit controls on highway work-zone operations. Intelligent Transportation Systems, vol. 8. Taylor & Francis, pp. 155–168.

Lu, X.-Y., Shladover, S., 2014. Review of variable speed limits and advisories: Theory, algorithms, and practice. Transp. Res. Rec.: J. Transp. Res. Board 2423, 15–23.

Lyles, R.W., Taylor, W.C., Lavansiri, D., Grossklaus, J., 2004. A field test and evaluation of variable speed limits in work zones. In: Transportation Research Board Annual Meeting (CD-ROM), Washington, DC.

MacDonald, M., 2008. Atm monitoring and evaluation, 4-lane variable mandatory speed limits 12 month report (primary and secondary indicators). Published by Department of Transport.

Middelham, F., 2006. Dynamic traffic management. Ministry of Transport, Public Works and Water Management, Directorate-General of Public Works and Water Management, AVV Transport Research Centre, Rotterdam, The Netherlands, Presentation to PCM Scan Team.

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning. Int. Conf. Mach. Learn. 1928–1937.

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

Mousavi, S.S., Schukat, M., Howley, E., 2017. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. IET Intel. Transp. Syst. 11 (7), 417–423.

Ng, A.Y., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E., Liang, E., 2006. Autonomous inverted helicopter flight via reinforcement learning. Experimental Robotics IX. Springer, pp. 363–372.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., Song, D., 2018. Assessing generalization in deep reinforcement learning. arXiv preprint arXiv:1810.12282.

Papageorgiou, M., Kosmatopoulos, E., Papamichail, I., 2008. Effects of variable speed limits on motorway traffic flow. Transp. Res. Rec.: J. Transp. Res. Board 2047, 37–48.

Papageorgiou, M., Kotsialos, A., 2002. Freeway ramp metering: An overview. IEEE Trans. Intell. Transp. Syst. 3 (4), 271–281.

Piao, J., McDonald, M., 2008. Safety impacts of variable speed limits-a simulation study. In: 11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008. IEEE, pp. 833–837.

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. Part C: Emerg. Technol. 57, 260–275.

Schaul, T., Quan, J., Antonoglou, I., Silver, D., 2015. Prioritized experience replay. arXiv preprint arXiv:1511.05952.

Schick, P., 2003. Effects of corridor control systems upon capacity of freeways and stability of traffic flow. Ph.D. thesis, Institute for Road and Transportation Science, Faculty of Civil and Environmental Engineering, University of Stuttgart.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P., 2015. Trust region policy optimization. Int. Conf. Mach. Learn. 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., 2016. Mastering the game of go with deep neural networks and tree search. Nature 529 (7587), 484.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M., 2014. Deterministic policy gradient algorithms. ICML.

Singh, S., Litman, D., Kearns, M., Walker, M., 2002. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. J. Artif. Intell. Res. 16, 105–133.

Soriguera, F., Torné, J.M., Rosas, D., 2013. Assessment of dynamic speed limit management on metropolitan freeways. J. Intell. Transp. Syst. 17 (1), 78–90.

Van der Pol, E., Oliehoek, F.A., 2016. Coordinated deep reinforcement learners for traffic light control. In: Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016).

Wang, Y., Zhang, Y., Hu, J., Li, L., 2012. Using variable speed limits to eliminate wide moving jams: a study based on three-phase traffic theory. Int. J. Mod. Phys. C 23 (09), 1250060.

Wei, H., Zheng, G., Yao, H., Li, Z., 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp. 2496–2505.

Weikl, S., Bogenberger, K., Bertini, R., 2013. Traffic management effects of variable speed limit system on a german autobahn: Empirical assessment before and after system implementation. Transp. Res. Rec.: J. Transp. Res. Board 2380, 48–60.

Wu, Y., Tan, H., Peng, J., Zhang, H., He, H., 2019. Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus. Appl. Energy 247, 454–466.

Wu, Y., Tan, H., Qin, L., Ran, B., Jiang, Z., 2018. A hybrid deep learning based traffic flow prediction method and its understanding. Transp. Res. Part C: Emerg. Technol. 90, 166–180.

Zhang, K., Liu, Z., Zheng, L., 2019a. Short-term prediction of passenger demand in multi-zone level: Temporal convolutional neural network with multi-task learning. IEEE Trans. Intell. Transp. Syst.

Zhang, Kunpeng, Jia, Ning, Zheng, Liang, Liu, Zijian, 2019b. A novel generative adversarial network for estimation of trip travel time distribution with trajectory data. Transport. Res. Part C: Emerg. Technol. 108, 223–244.

Zhu, F., Ukkusuri, S.V., 2014. Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. Transp. Res. Part C: Emerg. Technol. 41, 30–47.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A., 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 3357–3364.