Transportation Research Record

Intellectual Traffic Flow Optimization Using Variable Speed Limits

--Manuscript Draft--

Full Title:	Intellectual Traffic Flow Optimization Using Variable Speed Limits
Abstract:	Variable speed limit (VSL) strategies are proven effective traffic flow optimization policies. Previous research indicates when the VSL system was activated, it successfully lowered the average speed, hence retained the traffic inflow entering the congestion section and delaying the activation of a downstream bottleneck. In this paper, we proposed a deep reinforcement learning model that obtains input from the traffic camera system and determines the optimal policy of the variable speed limit. The model consists of two components: an intellectual agent based on Hybrid Deep Q Network and an online VSL controller to interact with traffic environment. The agent is trained to achieve the objective of improving vehicle mobility and overall traffic performance. The effectiveness of proposed model is evaluated in simulation and compared with different VSL strategies. Despite simplicity, the results show a promising performance of the proposed model in traffic flow optimization.
Manuscript Classifications:	Data and Information Technology; Information Systems and Technology ABJ50; Advanced Traffic Management Systems; Operations and Traffic Management; Intelligent Transportation Systems AHB15; Advanced Technology; Freeway Operations AHB20; Artificial Intelligence and Advanced Computing Applications ABJ70; Intelligent Agents
Manuscript Number:	20-04394
Article Type:	Presentation and Publication
Order of Authors:	Juanwu Lu
	Yu Tang

Intellectual Traffic Flow Optimization Using Variable Speed Limits

Juanwu Lu (Corresponding author)
College of Transportation Engineering
Tongji University, Shanghai, China 201804
Email: lujuanwu@tongji.edu.cn
Phone: +86-13557011135

Yu Tang
Key Laboratory of Road and Traffic Engineering of the Ministry of Education
Tongji University, Shanghai, China 201804
Email: tangyu@tongji.edu.cn
Phone: +86-21-69584687

Word Count: 4,539 words + 2 tables = 5,039 words

Submitted [August 1, 2018]

ABSTRACT

Variable speed limit (VSL) strategies are proven effective traffic flow optimization policies. Previous research indicates when the VSL system was activated, it successfully lowered the average speed, hence retained the traffic inflow entering the congestion section and delaying the activation of a downstream bottleneck. In this paper, we proposed a deep reinforcement learning model that obtains input from the traffic camera system and determines the optimal policy of the variable speed limit. The model consists of two components: an intellectual agent based on Hybrid Deep Q Network and an online VSL controller to interact with traffic environment. The agent is trained to achieve the objective of improving vehicle mobility and overall traffic performance. The effectiveness of proposed model is evaluated in simulation and compared with different VSL strategies. Despite simplicity, the results show a promising performance of the proposed model in traffic flow optimization.

Keywords: Variable speed limit (VSL), traffic flow optimization, reinforcement learning, deep Q-learning

INTRODUCTION

Variable speed limit (VSL) system is an important part of the intelligent transportation system (ITS). It optimizes traffic flows by adjusting speed limits on a regulated road. Drivers are informed the changes of speed limits via overhead or roadside variable message signs (VMS) upstream of the regulated road. Previous studies have demonstrated that the VSL system would lower average speeds, retain traffic flows entering a congested road segment and delay the activation of a downstream bottleneck (*1*). The implementation of VSL could improve safety and throughput, resolve traffic breakdown and reduce energy consumption and emission (*2; 3*).

VSL strategies are usually driven by measurements such as traffic volumes and occupancy, which are provided by loop detectors installed at discrete locations. The measurements are spot-based and only contain partial information of traffic operations. The nature of the measurements would potentially limit the efficiency of the VSL system. In contrast, traffic cameras cover larger spatial area and provide much richer information regarding traffic states. With advanced video processing techniques, the locations and speeds of individual vehicles could be extracted in real time. The wide deployment of traffic surveillance system in countries such as China motivates us to develop an intelligent VSL strategy based on real-time traffic video data.

Nevertheless, traffic video is high-dimensional inputs, which brings the difficulty of developing analytical algorithms. Fortunately, deep reinforcement learning techniques have shown great power in analyzing video inputs and searching for optimal strategies to play chess and play video games (*4; 5*). The success of deep reinforcement learning in various video-based tasks suggests that it is an appropriate tool to handle the video-based VSL problem.

This study customizes deep reinforcement learning algorithm for the development of a video-based VSL strategy. We design the state representations, reward function and the network structure to reflect the nature of the VSL problem. The effectiveness of the developed strategy is demonstrated numerically in comparison to an alternative approach.

This paper is organized as follows. We first present a brief review of related literature, an introduction of the deep reinforcement learning algorithm, and then briefly state the problem studied in this paper. The third section is a description of the proposed model and methods, followed by a demonstration in a real-world case study. Finally, the paper is summarized in the final section, with a discussion of possible direction for future research.

LITERATURE REVIEW

Existing VSL strategies can be classified into two categories: reactive rule-based approaches and proactive approaches. In rule-based strategies, the speed limits are determined based on the latest measurements of single or multiple parameters, such as traffic volumes, occupancies and speeds. For example, the strategy proposed by Smulders is based on traffic flows and speeds (*6*). The strategy lowers the speed limit when the observed parameters exceed preset thresholds. The strategy developed by Van den Hoogen and Smulders activates the VSL system only when traffic volume approaches capacity *(7)*. Elefteriadou et al. developed a VSL decision tree based on traffic flows, occupancies and speeds *(8)*. Many studies demosntrated that

the VSL is effective in reducing speed variance and improving traffic safety. Nevertheless, the abilitity of the reactive approaches in reducing the likelihood of traffic breakdown is limited *(1)*.

Proactive approaches make decisions based on predicted traffic states, which improve the ability to delay the activation of a downstream bottleneck. Most of the prediction models used in proactive approaches were based on three equations developed by Payne *(9)*, which describe traffic flow conservation, fundamental traffic flows and temporal and spatial speed evolution, respectively. For instance, Alessandri *et al.* used Payne's model to predict traffic density and determined the speed limits based on the evaluation of speeds over time *(10)*. Hegyi *et al.* modified the prediction model proposed by Papageorgiou *et al. (11)* when developing a VSL strategy *(12)*. They showed that the VSL strategy is effective in suppressing and even eliminating the effects of shockwaves. Conceivably, the effectiveness of the proactive VSL strategy depends on the accuracy and validness of the prediction model.

VSL strategies based on reinforcement learning (RL) techniques have been proposed recently. Walraven *et al.* developed a RL based VSL strategy based on vehicle speeds and density in observed area *(13)*. They showed that the strategy could reduce traffic congestion under high traffic demand. Li *et al.* proposed a RL-based VSL strategy for a large traffic system and presented positive results *(14)*. A deep RL algorithm introduces deep neural network into the RL framework to approximate some of the components in RL algorithm. It could extract high-dimensional features directly from video or image inputs. The deep RL algorithm has been applied in studies of traffic signal controls. For example, Liang *et al.* proposed a deep RL model to handle complex vehicular information matrix in traffic signal controls *(15)*. To the best of our knowledge, few studies have explored the validness of the deep RL algorithm on VSL applications.

PROBLEM STATEMENT

We proposed a deep reinforcement learning approach to optimally adjust speed limits in the area upstream of an on-ramp through, which is demonstrated in Figure 1. The objective of the system is to improve vehicle moblity and overall traffic performance.

Specifically, the left side of Figure 1 shows the structure of the RL model. It first use a trajectory extraction technique proposed by NGSIM *(16)* to take out vehicle positions and vehicle speeds from the road traffic camera data. The VSL agent continuously recieves and processes the data to obtain traffic state and corresponding reward. Then the agent would choose an appropriate speed limit based on the state and reward using the deep neural network in the right side. The change of speed limits is demonstrated by overhead or roadside variable message signs.

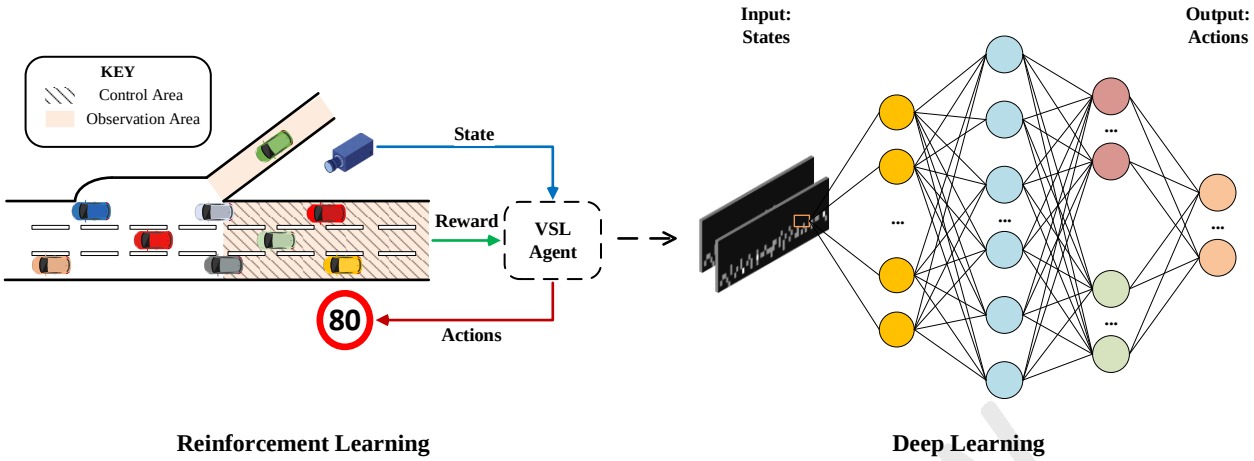

Figure 1 The variable speed limit control system

METHODOLOGY

Reinforcement Learning (RL) algorithm learns the optimal action policy to maximize the cumulative reward in the long run through trials and errors. Typically, an RL problem is formulated as a Markov Decision Process (MDP) in which the relationship of observations about the environment satisfies the Markov property *(17)*. In our case, optimization of the traffic flow using VSL control is to determine the optimal speed limits based on current state. Each time an action is taken the current state changes. Therefore, the VSL control problem in our case is formulated as a MDP problem and processed by RL technique *(18)*.

We adopt the Q-Learning (QL), a commonly used deep RL algorithm, to solve the problem. In QL, an agent is the action executor who interacts with the environment over time. The interaction can be denoted by <*s*, *a*, *s'*, *r*>, in which the agent takes an action *a* after obtaining a state *s* as input to reach next state *s'* and get a reward *r*. A value function and a policy are involved in RL. The value function predicts the expected, accumulative, discounted, future reward, which measures the quality of each state or state-action pair. The policy is a corresponding relationship between action *a* and state *s*. The value fuction under a specific policy $\pi(a|s; \theta)$ could be express by $Q^{\pi}(s, a)$:

$$Q^{\pi}(s, a) = \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k} \mid s_t = s, a_t = a, \pi\right]. \quad (1)$$

where γ is the discount factor in the range of [0, 1). Eq.(1) indicates that the nearest rewards are worthier than the rewards in the future. If the agent knows the optimal *Q* values of the succeeding state, the optimal *Q(s, a)* is calculated based on the Bellman optimality equation as follows:

$$Q^{\pi^*}(s, a) = \mathrm{E}_{s'}\left[r_t + \gamma \max_{a'} Q^{\pi^*}(s', a') \mid s, a\right]. \quad (2)$$

During training, a recursive algorithm is used to find the optimal policy π* based on Eq. (2). The QL algorithm will finally converge to the optimal *Q* value (i.e. the maximum *Q* value) if

explore for enough times and the learning rate is decreased at an appropriate rate *(19)*. The main components of our deep QL model, including the state space, action space, reward function and the deep neural network structure in our case are specifically designed and introduced as follows.

State space

A state *s* in our problem is defined as a matrix consisting of the position and speed information of vehicles within the observation area. Figure 2 illustrates the process of obtaining the state from observation data. After taking a snapshot of the observation area, vehicle trajectories can be extracted using trajectory extraction technique. We divide the whole observation area into small squared grids with the same size. The size is set such that the grid is sufficient to contain at least one passenger car. Each grid is associated with a two-dimensional vector <*Vehicle number, Average speed*> describing the status of the inside vehicles. *Vehicle number* is an integer, indicating the number of vehicles within the grid. A vehicle is considered in a grid if most parts of the vehicle are within the grid. *Average speed* denotes the mean speed of the vehicles within the grid.

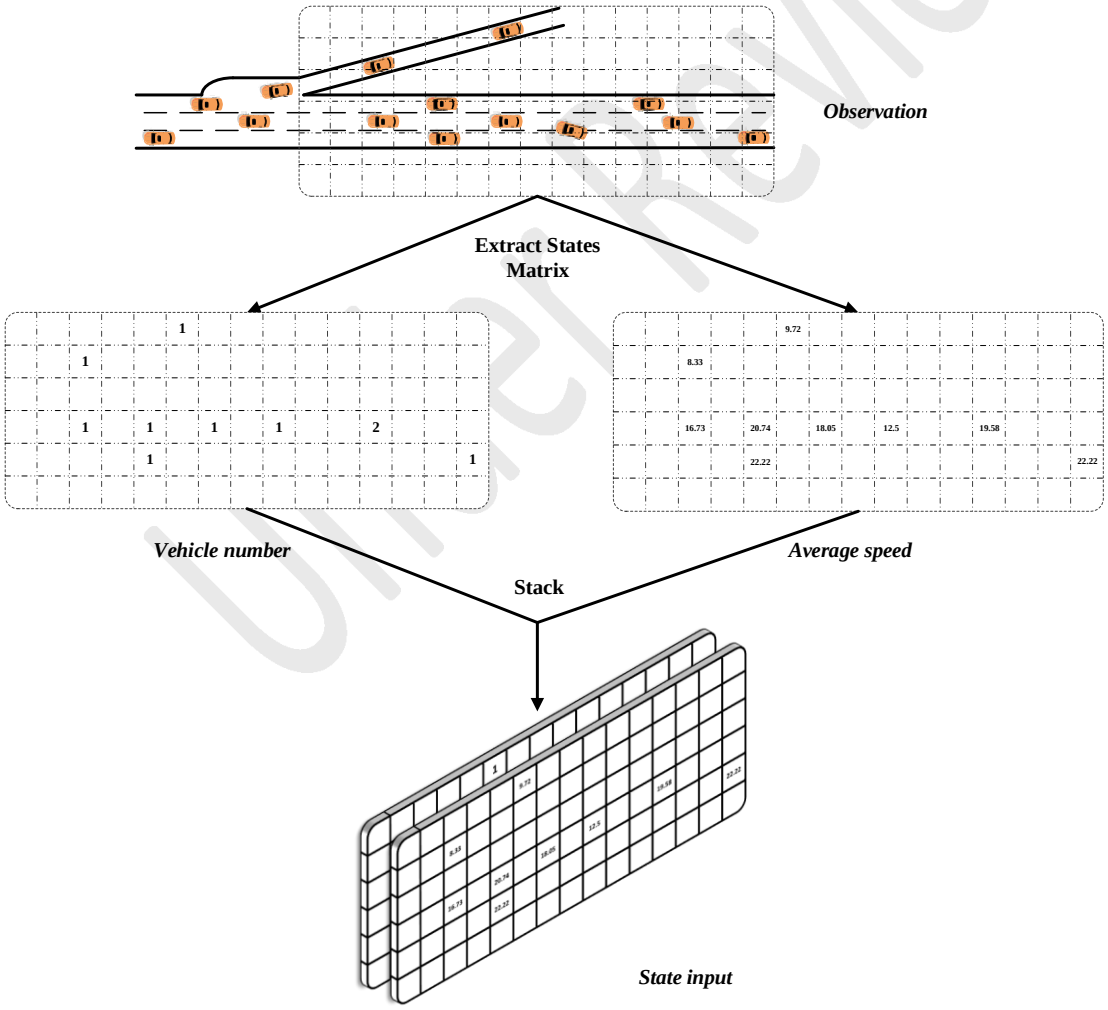

Figure 2 The process to generate states input

Action space

The action space A contains possible speed limits in the control area, which are defined in accordance with traffic regulations and rules. In the case study, our agent selects an action $a_t \in A$ every 3 minutes and the action space ranges from 40 to 80 km/h with fixed interval of 5 km/h.

Reward function

Setting up an appropriate reward function is critical since it plays an important role in motivating the agent to fulfill our expectations. In our case, the reward r_t at time-step t corresponding to action a_t is defined by:

$$r_t = \omega_1 \overline{V}_t + \omega_2 L_t. \quad (3)$$

where $\overline{V}_t$ represents the average speed in the merge area and L_t represents the total queue length in the merge area, the mainline upstream and on-ramp at time-step t. Positive ω_1, as well as negative ω_2, denotes the weights for different factors in rewards function. Proper values are assigned to two weights to balance different factors.

Hybrid Deep Q Network

We develop a Hybrid Deep Q Network (HDQN) as our deep neural network structure, which is illustrated in Figure 3. The HDQN structure is a combination of several neural network structures, including a primary DQN, a target DQN, and a prioritized experience replay. The primary DQN consists of two fundamental structures: (1) a Convolutional Neural Network (CNN) for feature extraction *(20)*, and (2) a dueling DQN for online Q value estimation *(21)*.

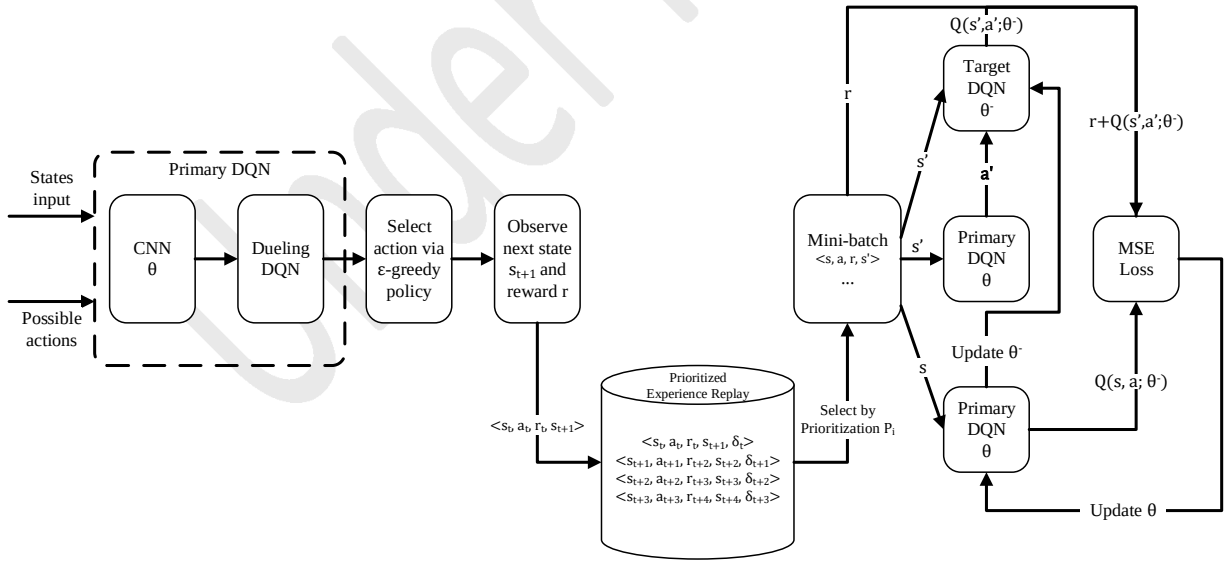

Figure 3 The structure of the deep QL model with Hybrid Deep Q Network.

The CNN consists of three convolutional layers. Each layer has a similar structure consisting of convolution and activation parts. The convolution structure includes multiple filters with different shapes. Each filter contains weights to aggregates patches in the input layer.

Different filters have different weights to extract different features as output to the next layer. The convolution structure operates as a feature extractor to illustrate areas of interests in states input. The activation function decides how a unit is activated. In this paper we employ the ReLU *(22)* as the activation function

The dueling DQN is formed by three linear fully-connected layers. After receiving the flattened output data from the convolutional layer, it splits the data into two different flows. One flow uses a linear fully-connected layer to calculate the value of the current state V and the other calculates an advantage function A. The outputs of the two flows merge in the third layer to estimate the Q value, which is a sum of the value V and the function A defined by:

$$Q(s, a; \theta) = V(s; \theta) + (A(s, a; \theta) - \frac{1}{|A|}\sum_{a'} A(s, a'; \theta)). \quad (4)$$

$A(s, a; \theta)$ shows the absolute importance of a single action to the value function among all actions. The parameters in primary DQN are updated each time sampling from the experience replay by calculating the training error using the Mean Square Error (MSE) define by:

$$J(\theta) = \mathrm{E}_{s,a}[(Q_{target}(s, a) - Q(s, a; \theta))^2]. \quad (5)$$

The target DQN is a separate neural network to increase training stability. A double DQN algorithm *(23)* is used to generate the target Q value estimation in the target network, which is expressed in the following equation:

$$Q_{target}(s, a) = r + \gamma Q(s', \operatorname{argmax}_{a'}(Q(s', a'; \theta), \theta^-)). \quad (6)$$

It uses a duplicated online primary DQN to estimate the target Q value and could effectively resolve the overestimation issue. The parameters in target network are updated by copying those in primary DQN with certain frequency.

While training, we adopt a prioritized experience replay *(24)* to sample mini-batches from an experience replay and use them to update primary DQN parameters in order to improve the independence among samples. We optimize our neural networks by implementing ADAptive Moment estimation algorithm (ADAM) *(25)*.

The overall algorithm for training the variable speed limit control system is presented in **Table 1**. Each training episode can be set as one experiment in microscopic traffic simulator or one day in reality. Its goal is to train a mature agent who can adjust upstream speed limit value in response to different traffic scenarios. The agent initially chooses actions randomly till the time-step reaches the pre-train steps and the replay buffer is filled with samples for training. Initially, each sample has the same priority. Hence, samples are randomly selected into mini-batches for training. The samples' priorities change during the training process and then they are selected by different probabilities. We use ADAM backpropagation algorithm to update parameters in the primary DQN. After training, the agent chooses actions based on the ε-greedy policy and finally learns to looking forward to a high reward while reacting on different traffic scenarios.

TABLE 1 Hybrid Deep Q-Learning Algorithm for Variable Speed Limit

Input: Replay memory size M, pre-train steps T_p, Mini-batch size B, greedy ε, Discount factor γ, Target network update interval τ, Learning rate η, Action interval T_d

Notations:
θ: the parameters in primary DQN.
θ^-: the parameters in target DQN.
f: the simulation time-step in an episode.
Initialize parameters θ, θ^- with random values.
Initialize empty prioritized experience replay buffer m with a size of M.
Initialize training time-step t to be zero.
Initialize states input s by obtaining the starting scenario of the simulation.
While there exists a state input s **do**
  If mod (f, T_d) = 0 **then**
    Choose an action a based on ε-greedy policy.
    Take the action a and observe new state s' and reward r.
    If the size of replay buffer m = M **then**
      Remove the oldest experiences in the replay buffer.
    End if
    Add the four-element tuple <s, a, r, s'> into m.
    Assign s' to s: $s \leftarrow s'$.
    If the size of replay buffer m = M and $t = T_p$ **then**
      Uniformly sample mini-batches with a size of B from buffer m based on the sampling priorities.
      Calculate the MSE loss J:
      $J(\theta) = E_{s,a}[(Q_{target}(s, a) - Q(s, a; \theta))^2]$.
      Update θ with ∇J using ADAM backpropagation algorithm with learning rate η.
      If mod (t, τ) = 0 **then**
        Update the target network by setting $\theta^- = \theta$.
      End if
      Update every experience's sampling priority based on TD-Error δ.
      Update the value of ε.
    End if
  End if
  $t \leftarrow t + 1$.
End while

CASE STUDY

We conduct experiments to evaluate our model using the microscopic simulation platform SUMO *(26)*. The the simulation environment is built based on a real-world expressway, which is shown in **Figure 4**. The expressway connects Qingdao and Huangdao through the tunnel in Shandong province, China. The one-lane on-ramp and the upstream three-lane mainline merge in a four-lane merge area and gradually reduce to three lanes in the downstream segment.

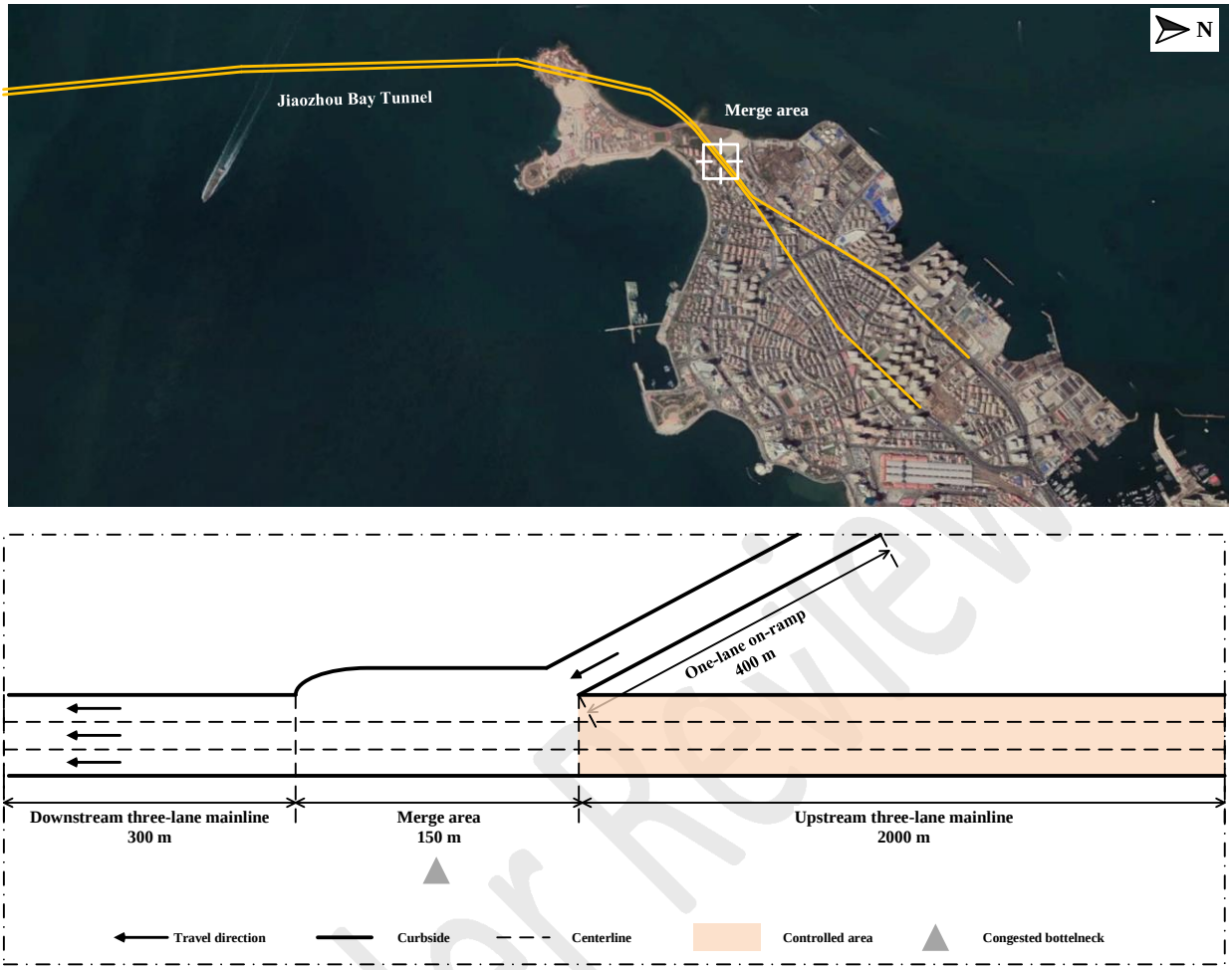

Figure 4 Illustration of study expressway site

The simulation considers a road network covering the on-ramp from Tai Xi Third Rd, the upstream and downstream mainlines and the merge area of the tunnel. Current speed limits in the mainline and on-ramp are 80 *km/h* and 40 *km/h* respectively. Parameters regarding drivers' behaviors, such as drivers' compliance of the speed limit and the accepted headway, have been well calibrated in this simulation. **Figure 5** illustrates the fluctuation of the traffic volume recorded every 15 min from 8:00 a.m. to 10:30 a.m. on the mainline and on the on-ramp in the study site.

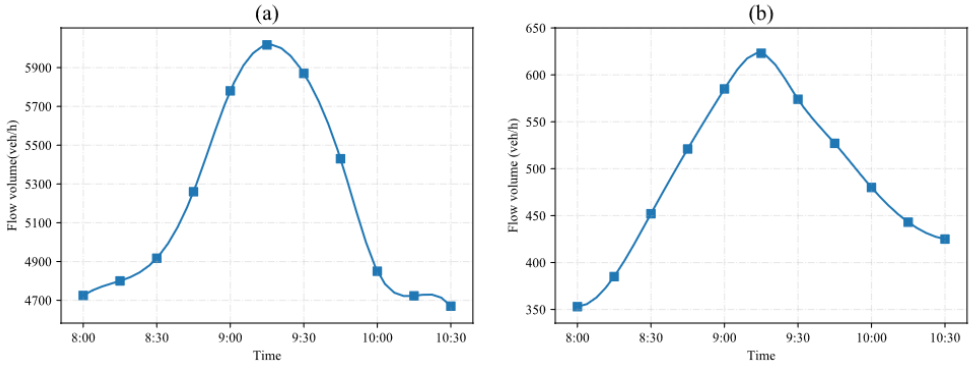

Figure 5 Traffic volumes: a) on the mainline; b) on the on-ramp.

The states input for our deep RL agent covers the entire simulated expressway network. The size of a single state input is a 2×10×200 matrix after down-sampling and stack operation. In the three convolutional layers mentioned above, each layer has different filters and stride setting. The first and the second convolutional layers have 32 4×4 filters with a stride of 2×2 and 64 2×2 filters with a stride of 1×1, respectively. The third convolutional layer for calibration has 64 1×1 filters with a stride of 1×1. The linear fully-connected layers for calculating value and advantage of actions have the same structure of 256 units. The output layers for calculating the value and for calculating the advantage are fully-connected layers with one unit and nine units, respectively. Other hyper-parameters of the HDQN in our control system are listed in **Table 2**. The weights ω_1 and ω_2 for the reward function is calibrated through pre-experiments.

TABLE 2 Hyper-parameters in HDQN for Variable Speed Limit

Parameters	Value
Replay memory size M	200000
Mini-batch size B	64
Starting $\varepsilon_{initial}$	1.00
Ending ε_{final}	0.01
Maximum time-step for ε	400000
Pre-train steps T_p	200000
Discount factor γ	0.99
Target network update interval τ	1000
Learning rate η	0.0001
Action interval T_d	180
Reward weight ω_1	0.1
Reward weight ω_2	-0.04

Results

Before conducting experiments for evaluation, we use random seeds to generate various traffic scenarios and have trained our agent for 10,000 episodes.

For comparisons, we consider three scenarios in our evaluation, a scenario without VSL control, a scenario with rule-based VSL control strategy proposed by Elefteriadou, L. *et al.*, and

a scenario with the proposed HDQN-based VSL control system. Parameters in two control strategies have been tested and calibrated for optimal control performance.

A. Average Reward

We first compare the average reward under different control strategies. By making all strategies to have the same reward functions as our work, the average reward of an episode in our HDQN model is -1400 at the beginning and gradually increases and reaches 5000, while it's -2200 and around 2000 in the no-control scenario and the rule-based control scenario, respectively. The results suggest that our HDQN model has outperformed the rule-based strategy after a long period of training, revealing that our HDQN model is able to handle the high-dimensional state inputs and manage to effectively control the upstream flows.

B. Vehicle Mobility

We further compare multiple parameters under different control strategies to analyze vehicle mobility with different VSL strategies. The mainline travel time with three control strategy are compared in Figure 6(a). The average travel time in 2.5 hours with the proposed strategy, with rule-based strategy and without control are 3.32 minutes, 3.43 minutes and 3.61 minutes, respectively. The result shows a 3.24% and a 7.76% reduction in travel time with our model, when compared with the other two strategies. The results indicate that our agent is able to optimize the traffic flows more efficiently.

The comparison of the on-ramp queue length shown in Figure 6(b) also supports the conclusion. The maximum queue length on ramp with proposed strategy is 0 m, compare to 8.3 m with rule-based strategy and 34.4 m without control. The results reveal that our proposed model could effectively balance traffic flows on the mainline traffic and at the on-ramp.

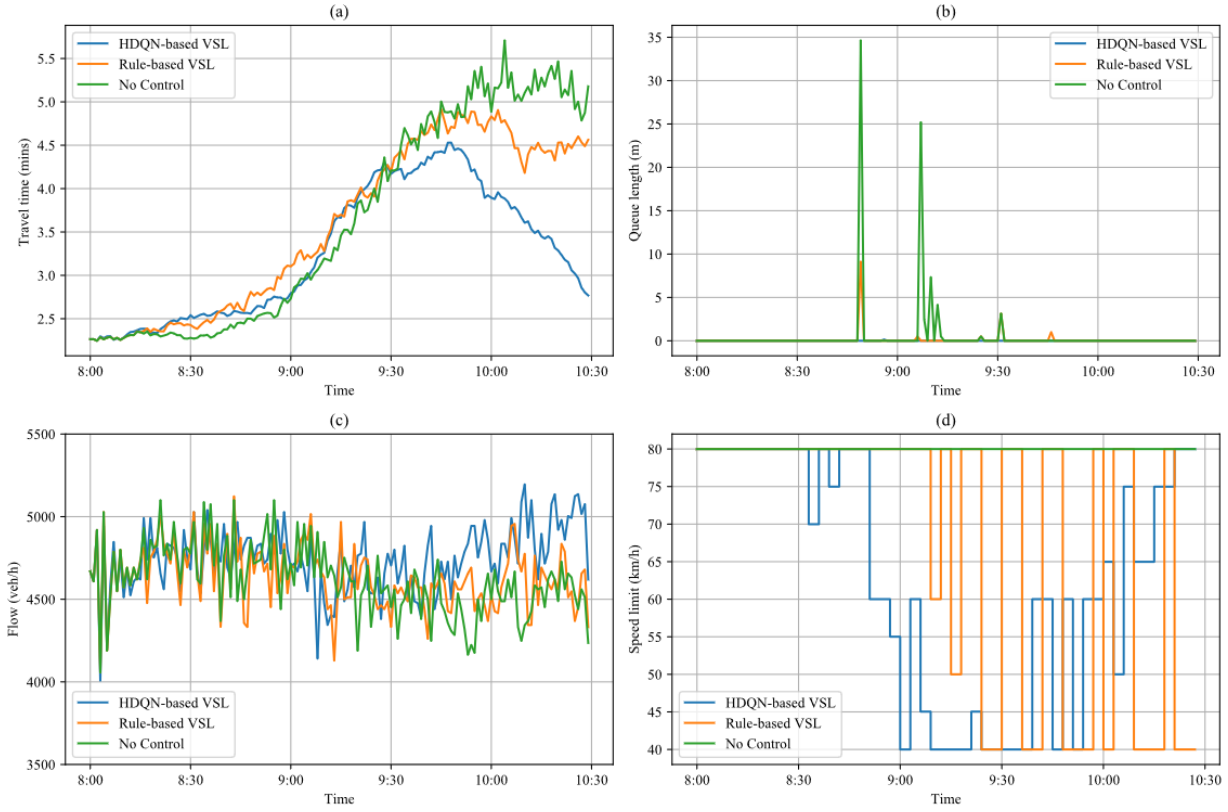

Figure 6 Results: a) speed limit; b) mainline travel time; c) queue length on ramp in meters; d) downstream flow.

Downstream traffic flows with different control strategies are also compared in Figure 6(c). It can be found that the downstream flow is more stable with the proposed strategy, when compared with the no-control strategy. The flow is higher than that resulting from the rule-based strategy. Specifically, with HDQN-based strategy, the average downstream flow in 2.5 hours is 4,534 veh/h, which are 4,437 veh/h and 4,420 veh/h, respectively, with the rule-based strategy and the no-control strategy.

Figure 6(d) reveals that the proposed controller starts to lower speed limits before the rule-based controller. The results suggest that the proposed controller is proactive. That is, it adjusts the speed limits before the formation of traffic congestion. In addition, it is found that the proposed controller is less aggressive than the rule-based controller.

CONCLUSIONS AND DISCUSSION

We propose a variable speed limit control system based on the HDQN neural network and reinforcement learning to regulate traffic flows of highway traffic. An important contribution of our study is introducing the combination of convolutional neural network to better process the high-dimensional traffic data for variable speed limit control. We further evaluate our model by comparing it with a rule-based control strategy and no-control case. The results reveal that our model has better performance in managing traffic flow.

In real-world scenarios, traffic conditions could be more complicated. The digital cameras would suffer from severe weather condition with low visibility. The state extraction process could be disturbed by video resolution, vehicle size, velocity, and other factors. Hence, although the result is encouraging, further research is required to exam the robustness of our model under various traffic situations. Furthermore, traffic accidents and other abnormal traffic events are not considered in our study. A more efficient and reliable training method remains to be lucubrated. Finally, we only cover the variable speed limit control at one on-ramp, and we leave aside the factor of drivers' compliances. The extension of the proposed approach for various types of weaving areas is worth investigation.

ACKNOWLEDGMENTS

This research is part of a project attending the NACTranS preliminary. Special thanks to professor Yuxiong, Ji for supporting and providing enlightening suggestions, and to all the fellows, including Lei Chen, Xiaonan Shi, who have given us their help in this project.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: Juanwu Lu, Yu Tang; data collection: Juanwu Lu; analysis and interpretation of results: Juanwu Lu, Yu Tang; draft manuscript preparation: Juanwu Lu, Yu Tang. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

1. Khondaker, B., and L. Kattan. Variable speed limit: an overview. *Transportation Letters*, 2015. 7:5: 264-278.

2. Hegyi, A., B. De Schutter, and J. Hellendoorn. Optimal Coordination of Variable Speed Limits to Suppress Shock Waves. *IEEE Transactions on Intelligent Transportation Systems*, 2005. 6(1):102-112.

3. Schutter, B. D.. A Model-Based Predictive Traffic Control Approach for the Reduction of Emissions. *Trail in Perspective Proceedings International Trail Congress*, 2008.

4. Silver, D., *et al*.. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016. 529(7587): 484–489.

5. Silver, D., et al.. Mastering the game of go without human knowledge. Nature, 2017. 550(7676): 354-359.

6. Smulders S.. Control by variable speed signs - the Dutch experiment. *Proceedings of the Sixth International Conference on Road Traffic Monitoring and Control*, 1992. 99-103.

7. Van den Hoogen, E., and Smulders, S. Control by variable speed signs: results of the Dutch experiment. *Proceedings in 7th International Conference on Road Traffic Monitoring and Control*, 1994. 391: 145-149

8. Elefteriadou, L., S. Washburn, Y. Yin, V. Modi, and C. Letter. Variable Speed Limit (VSL) - Best Management Practice. *Bottlenecks*, 2012.

9. Payne, H. J. Models for freeway traffic and control. In *Mathematical models of public systems Vol. 1* (G. A. Bekey, ed.), Vista, CA: SCS. pp. 51-61

10. Alessandri, A., A. Di Febbraro, A. Ferrara, and E. Punta. Nonlinear optimization for freeway control using variable-speed signaling. *IEEE Transactions on Vehicular Technology*, 1999. 48(6): 2042-2052.

11. Hegyi, A., B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 2005. 6(1): 102-112

12. Papageorgiou, M., J.-M. Blosseville, and H. Hadj-Salem. Modeling and real-time control of traffic flow on the southern part of Boulevard Peripherique in Paris: part I: modeling. *Transportation Research Part A*, 1990. 24A: 345-359.

13. Walraven, E., M. T. J. Spaan, and B. Bakker. Traffic flow optimization: a reinforcement learning approach. *Engineering Applications of Artificial Intelligence*, 2016. 52, 203-212.

14. Li, Z., P. Liu, C. Xu, H. Duan, and W. Wang. Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks. *IEEE Transactions on Intelligent Transportation Systems*, 2017. 1-14.

15. Liang, X., X. Du, G. Wang, and Z. Han. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, 2019. 68(2): 1243-1253.

16. *Next Generation SIMulation Fact Sheet: NGSIM Overview*. FHWA-HRT-06-135. https://www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm. Accessed March 23, 2019.

17. Sutton, R., and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

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

19. Watkins, J. C. H. Christopher, and P. Dayan. Q-learning. *Machine Learning*, 1992. 8(3–4): 279–292.

20. LeCun, Y., Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015. 521:436–444.

21. Wang, Z., T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas. Dueling network architectures for deep reinforcement learning. *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn.*, 2016. pp. 1995–2003.

22. Hinton, G. E.. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. *International Conference on International Conference on Machine Learning*, 2010.

23. Van Hasselt, H., A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. *Proc. 13th AAAI Conf. Artif. Intell.*, 2016. 2094-2100

24. Schaul, T., J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. *Proc. 4th Int. Conf. Learn. Representations*, 2016.

25. Kingma, D. P., and J. Ba. Adam. A method for stochastic optimization. *Computer Science*, 2010.

26. Krajzewicz, D., J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of sumo-simulation of urban mobility. *Int. J. Adv. Syst. Meas.*, 2012. 5(3): 128-138