Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-5 (2020) 176–181

Deep Behavioral Cloning for Traffic Control with Virtual Expert Demonstration Under a Parallel Learning Framework ⋆

Xiaoshuang Li *,** Fenghua Zhu * Fei-Yue Wang *

* *State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (Email:{lixiaoshuang2017,fenghua.zhu@ia.ac.cn,feiyue.wang}@ia.ac.cn).*
** *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.*

Abstract: Intelligent traffic signal control is necessary for improving traffic efficiency. These fast changing and challenging traffic scenarios and demands are generally handled by professional traffic engineers. However, it may take years of time and thousands of practices to train such an engineer. This paper proposes a deep behavioral cloning method to learn how to control the traffic signal effectively and efficiently from virtual expert demonstration. The method imitates promising working behavior of optimized offline solutions, and applies it to solve online traffic signal control problems of the similar scenario. Different traffic demand patterns are generated through a combination of different kinds of components. Then the virtual demonstration is constructed by getting an exclusive and optimized solution for each generated virtual traffic demand pattern through a heuristic random search method. After that, a deep neural network-based behavioral cloning method is employed to learn from the virtual demonstration and finish on-line traffic signal control task. The experimental results show that compared with other methods, the proposed method significantly reduces the waiting time and time loss in different situations. And the average traffic speed of the road network at different saturation levels can be improved by 1.58% to 11.54%.

Keywords: intelligent transportation system, behavioral cloning, virtual demonstration, parallel learning

1. INTRODUCTION

In traffic control engineering practice, there are two main traffic control method: pre-defined fixed-time plan method and adaptive method. Due to the reliable performance, pre-defined fixed-time plan method is widely used in a great number of intersections over the worldwide. This method and its variants operate according to a predetermined time plan, which is based on the historical traffic demand. However, traffic demand is fast changing nowadays because of rapid growth in travel demand and uncertainties. The city traffic scenario is becoming more and more complicated, which has led to a big challenge for effective and efficient traffic signal control [Mirchandani and Wang (2005)]. Adaptive traffic signal control methods are more flexible because different control strategies will be adopted when the pattern of traffic demand changes. Genetic algorithm [Gokulan and Srinivasan (2010)], neural networks [Srinivasan et al. (2006)] and so on are all have been used in adaptive traffic signal control methods. Although these methods behave better than pre-defined fixed-time methods in most cases, these method cannot cope with the rapid change of traffic flow pattern well. Professional traffic signal engineers are still needed to handle the complicated situation.

Although pre-defined fixed-time plan method and adaptive method are able to alleviate the existing traffic congestion problems in urban areas, professional traffic signal engineers are still needed. They assist the employed control system to cope with exceptional situation [Jin et al. (2020)]. Imitating the professional traffic signal engineers is a feasible approach to solve complicated traffic signal control problems. Traffic engineers can provide demonstration data for the imitation learning methods, such as behavioral cloning. However, there are not enough demonstration data to train the deep neural network. The impact of demonstration on control performance is also unclear.

Parallel learning framework [Wang et al. (2016)] helps to solve the issue that the demonstration of experts are hard to be obtained. The parallel learning framework is designed for the complex system and in this framework, parallel system [Wang (2010)], including real-world complex systems and artificial system, the results of the computational experiments in the artificial system can guide the real world [Lv et al. (2019)]. To tackle the limitation

⋆ Research supported in part by the National Key R&Development Program of China under Grant 2018YFB1700202, NSFC U1811463, U1909204, 61773381, 61533019.(*Corresponding author:Fenghua Zhu*)

10.1016/j.ifacol.2021.04.096

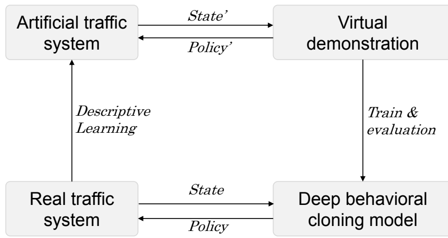

Fig. 1. Parallel learning framework of the deep beahvioral cloning method.

of demonstration in the real-world, generating some big virtual demonstration data in the artificial system provide a solution. As shown in Fig. 1, with the help of virtual demonstration data, the deep behavioral cloning model can be trained to respond to different traffic states properly.

In this paper, we proposed to generate virtual traffic signal control demonstration to train a deep behavioral cloning model. Under the parallel learning framework, different traffic demand patterns are generated and an exclusive control strategy will also be generated for each generated traffic demand pattern. A deep behavioral cloning (BC) method is employed to learn from the virtual demonstration and finish an on-line traffic signal control task. It is a new data-driven method for urban traffic control problem.

2. DEEP BEHAVIORAL CLONING METHOD WITH VIRTUAL DEMONSTRATION

In this section, the details of the presented deep behavioral cloning method via the virtual expert demonstration are introduced. Firstly, a special approach is designed to solve the problem of the diversity of traffic demand patterns. Then, we describe how to obtain virtual demonstration with different patterns of traffic demand. After constructing our virtual demonstration dataset, a deep behavioral cloning model is constructed to complete the traffic control task. Fig. 2 shows the whole work process.

2.1 Using different components to synthesize traffic demand

The professional and experienced traffic engineers have learned from various traffic demand patterns in daliy work, so our virtual demonstration also needs different traffic demand patterns. With the help of the simulation platform, the traffic demand can be designed and the cost of collecting data is close to zero, which is impossible in real traffic scenarios. High quality of demonstration data makes it possible for behavioral cloning method to learn an excellent control strategy. We utilize different traffic demand components to synthesize traffic demand here, look like $d = kf(t)+b+\varepsilon$. d is the parameter which informs the simulation platform how to generate traffic demand. $k, f(t), b$ and ε are componets and represent the change speed of the demand, the shape of demand, the demand value at begining and noise, respectively. The combination of different components value constitutes a variety of traffic demands. The part named different traffic demand patterns in Fig. 2 shows some instances of different traffic demand patterns.

2.2 Virtual demonstration of traffic signal

Demonstration can determine the performance of the control method. Therefore, a novel method is designed to collect excellent virtual demonstration by optimizing each generated traffic demand pattern. Virtual traffic demand that under control makes it convenient to estimate and improve the performance of the exclusive scheme of a certain traffic demand pattern. For a specific traffic demand pattern, the off-line optimization method can give a nearly perfect solution. In this paper, a heuristic random search (HRS) method is designed to gain a better scheme for every different traffic demand patterns we have synthesized.

$$d_{j+1} = d_j - \frac{\alpha\beta}{N\sigma}\sum_{k=1}^{b}[w_{j,k,+} - w_{j,k,-}]\delta_j^k \tag{1}$$

The core of HRS looks like Eq. (1), where d_j and d_{j+1} are the traffic signal schemes, j is the iteration step, α is the step size, $\beta = \frac{1}{\frac{1}{\beta_1}e^{j/\beta_2}+1}$ is the reduction ratio of step size, N is the total number of exploration schemes, σ is the variance of the $2b$ objective function value used in the update step, δ_j^k is the exploration noise vector, which has the same shape with d_j and w is the objective function value. For each iteration step j, N exploration noise vectors are sampled from the normal distribution and N new exploration schemes are generated by adding these vectors to the current scheme d_j and $w_{j,k,+}$ are the objective function value of the $k-th$ exploration scheme. The other N new exploration schemes are obtained by subtracting the b noise vectors from d_j and $w_{j,k,-}$ are the objective function value, too. Pick the top b exploration schemes with the best objective function value to decide the update direction.

The HRS is a gradient-free optimization method. It utilizes the difference between different exploration schemes and gains some heuristic information, for example, the variance σ indicates how much the objective function value changes at the current step. When the parameters are updating, large variance means more careful search is necessary and small means bigger step size are needed to search more. We make use of it to generate the demonstration control strategy for each certain traffic demand pattern. At last, we obtain an exclusive and optimized solution for each synthesized traffic demand and running the simulation helps us to collect the demonstration dataset $D = \{(s_t, a_t)\}$, where s_t is the traffic state at t and a_t is the corresponding time schedule.

2.3 Deep behavioral cloning model for traffic control

Deep behavioral cloning method [Kebria et al. (2020)] is a kind of imitation learning approaches. The imitation learning method learns and imitates how experts, who are good at dealing with complicated problems, to handle a difficult situation. Given the state s, the goal of imitation lerning is to learn a policy π which can produce expected action or trajectory. One of the possible methods to achieve the goal is to obtain a mapping from state s to action a.

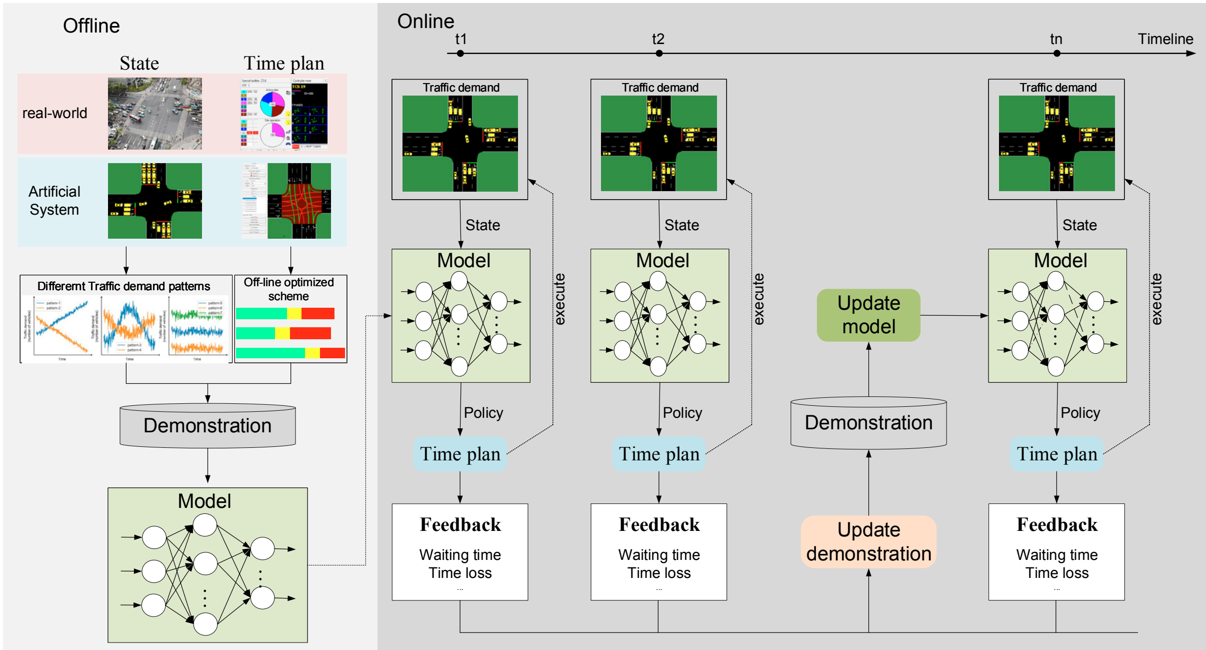

Fig. 2. Deep behavioral cloning traffic signal control model framework. The offline part generates virtual demonstration and trains the deep behavioral cloning neural network. The online part takes command of the traffic signal.

Algorithm 1 Deep behavioral cloning algorithm

obtain demonstration $D = (s_t, a_t)$;
construct deep neural network to present the policy π with θ, denoted as π_θ;
select an object or loss function $\mathcal{L}$;
adjust policy parameters θ to optimize $\mathcal{L}$ based on D;
Return: optimized policy parameters θ and get policy π_θ

The general algorithm of behavioral clonging is shown as Algorithm. 1. Here, $D = (s_t, a_t), t = 1, 2, 3, ..., N$, (s_i, a_i) is a pair of state and action at time step i. The loss function $\mathcal{L}$ evaluates the difference between a_i and $\pi_\theta(s_i)$. In other words, the loss function measures the similarity between the expert policy contained in the demonstration and the student policy π_θ. Common loss functions include $l1$-loss, $l2$-loss, KL divergence and other functions.

In this paper, the deep behavioral cloning model is utilized to learn traffic signal control policy. Demonstration $D = \{(s_t^{(i)}, a_t^{(i)})\}$, consisting of sample pairs of traffic state and corresponding time schedule, is used as the train dataset for the neural network. The control policy $\pi_\theta(*)$, parametrized by θ, learns a function which can reconstruct the time schedule from the traffic state for each (state, action) pair. Traffic flow collected during the last 30 minutes has been used as the input data of the model. The input $s_t = \{f_t^l\}, t < N, l < M$, includes all traffic flow information of total M lanes during last N time steps, $s_t \in \mathbb{R}^{M \times N}$. The output $a_t = \{r_1, r_2, \ldots, r_q\}$, $a_t \in \mathbb{R}^q$, r_i is the duration of phase i, q is the number of signal phases for the signalized intersection. The number and order of the phases for a intersection do not change in most cases and in this paper, $q = 4$ and it does not change.

The deep behavioral cloning is a deep learning based method but it is designed to deal with the decision-making

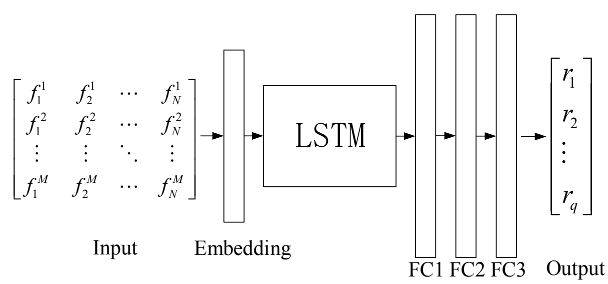

Fig. 3. The deep behavioral cloning neural network architecture.

problems, which is different from the parttern recognition peoblems. The neural network of deep behavioral cloning model, shown as Fig. 3, chooses long short-term memory network (LSTM) as the most important part. LSTM cell [Hochreiter and Schmidhuber (1997)], a typical variant of RNNs, has overcome the limitations of RNNs in vanishing gradients. After some necessary embedding process, LSTM cell helps us to extract temporal features of the traffic state. In order to make use of the outcome of the LSTM and generate correct traffic control schemes, there are some fully connected (FC) layers at the end of the model.

$l1$ and $l2$ loss functions are often used in deep behavioral cloning algorithm to fit the demonstration data. In our experiments, the action in virtual demonstration is the duration of all phases, therefore, the $l2$ loss function (Eq. 2) is used in this paper. By minimizing the loss function, the model will be able to generate a scheme which is as consistent as possible with the strategy of demonstration,

$$\mathcal{L}_{l2} = \|\pi_\theta(s_t) - a_t\|_2^2 \tag{2}$$

3. CASE STUDY

3.1 Experiment setting

Simulation of Urban MObility (SUMO) and its python API are used to construct our micro traffic model and finish our traffic simulation tasks. The simulation environment for the case study is a classical four-way intersection. The length of all arms in this intersection is set to 1000 meters and each road has three incoming lanes and two outgoing lanes. At a distance of 100 meters from the intersection, the number of lanes changed from two to three, to achieve flexible drainage. The traffic light here contains four different phases (Green-SouthNorth-through, Green-SouthNorth-left-turn, Green-WestEast-through, Green-WestEast-left-turn) and a 3-seconds yellow light phase follows the green phase before it turns to red light.

Besides our deep behavioral cloning traffic light control method, pre-defined fixed-time method, vehicle-actuated method and DQN-based on-line control methods are employed here to compare with our methods. (1) For fixed-time controller, the duration of all phase is 30 seconds, which is a universal setting in the real world. (2) Vehicle-actuated control method changes the split dynamically. There are detectors under the ground at the intersection and if the time-gap between two vehicles over three seconds when the duration of the green phase is greater than the minimum duration or the length of current phase equals to maximal duration, the traffic light will change to the next phase automatically. The duration of phase ranges from 25 to 35 seconds for the actuated controller. (3) DQN-based method takes the same state data with the proposed method and the average speed of vehicles passing through the intersection is used as the reward. The deep Q-network decides how to adjust the duration of every phase. The duration can increase one second, decrease one second or hold for every step. In order to stabilize the training process, the minimum green light time is set to 15 seconds. The neural network of the DQN controller also utilizes the LSTM cell as the core part to extract the temporal feature to find better control strategy.

There are two kinds of traffic demands. (1) Different saturation levels (the number of generated vehicles per hour) is set to simulate a different traffic condition. (2) Weekdays and weekends are also taken into consideration to verify whether the deep behavioral cloning method has the flexibility to adapt to different situations.

3.2 Evaluation criteria and datasets

The performance of our method is evaluated under four different criteria. The first one is the average waiting time. It can evaluate the time each vehicle has spent in halting speed on the road on average. The longer the waiting time, the more serious the traffic congestion is. In this paper, the average waiting time is defined as Eq. 3.

$$\overline{T}_{wait} = \hat{\mathbb{E}}\left[\sum_{i=0}^{N} T_{wait}^{(\mathrm{veh}_i)}\right] \tag{3}$$

Another criterion is the average time loss. This criterion as Eq. 4 describes the gap between the ideal time and actual time it spends to arrive at its destination. If the speed of vehicles is higher than the halting speed but lower than the free-flow speed, the time loss will increase but waiting time holds.

$$\overline{T}_{loss} = \hat{\mathbb{E}}\left[\sum_{i=0}^{N} \left(T_{real}^{(\mathrm{veh}_i)} - T_{ideal}^{(\mathrm{veh}_i)}\right)\right] \tag{4}$$

Macroscopic fundamental diagrams (MFD) [Daganzo and Geroliminis (2008)] and average speed are another two significant criteria. MFD can describe the number of vehicles on the road net at every moment. Under the same traffic demand, more vehicles in the net, higher risk of traffic jam. It evaluates the probability of the traffic congestion of every moment during the day. Average speed is the most intuitive description of traffic conditions. All these four different criteria are used to enable a comprehensive assessment of performance of the proposed deep behavioral cloning method.

All differernt traffic demand components used in this paper are listed in Table. 1 and we generate 1628 different traffic demand patterns to make the demonstration diverse. For example, $d = 2sin(\pi t/120) + 3 + \epsilon$, where $t = [1, 2, \ldots, 60], \epsilon \in \mathbb{R}^{60}$, defines a vector d, which could be used to adjust the repetition rate in SUMO. Different d means different vehicles generate partterns and different traffic demands in the simulation environment.

Table 1. traffic demand components setting

Components	Range
k : change speed	0.5, 1, 2, 4, 6, 8
$f(t)$: shape of demand	$0, 1, 2,\ t,\ -t+p,\ sin(\pi t/p),$ $-sin(\pi t/p)+1,\ cos(\pi t/2p),\ sin(\pi t/2p)$ $-sin(\pi t/2p)+1,\ -cos(\pi t/2p)+1$
b : value at start	0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4
ε : noise	$\varepsilon \sim N(0,1)$ with different random seeds

* p is the length of simulation time (off-line) here.

Three different virtual demonstration datasets are generated according to the improvement in waiting time after the same HRS optimization process, which contains 20 iteration steps and same parameters. Since a certain demand pattern does not last too long in real traffic scenarios, we choose 3600 seconds as the length of simulation when we synthesize traffic demand and generate the virtual demonstration. We collect a sample pair (s_t, a_t) every minute when the simulation step between 600 and 3300 seconds because the simulation environment needs time to load vehicles at the start time and the simulation has to end at 3600 seconds whether there is an exceptional situation. We give up the data for the first 10 minutes and last five minutes to ensure the quality of the data. For every generated traffic demand pattern, the exclusive time plan will be optimized to improve the performance. Since we need the demonstration datasets teach our model to reduce the waiting time, the sample that failed to reduce the waiting time by more than 10% will be dropped. And if it improved more than 10%, the sample will be assigned to three different datasets with which train the deep imitation learning model: (1) Dataset-10: improved 10%-15%, model name: BC-10; (2) Dataset-15: improved 15%-20%, model name: BC-15;(3) Dataset-20: improved more than 20%, model name: BC-20.

In order to obtain a dimensionless expression to reduce the impact of the value range of data on the neural networks, Min-max data normalization was used to rescale the raw state such that all feature values were within the range of 0 and 1.

3.3 Hyperparameters set

Hyperparameters can seriously affect the performance of neural networks. We use TPE method [Yoo et al. (2017)] to find better hyperparameters set. The search space of hyperparameters and value used to construct the BC model are listed in Table. 2.

Table 2. Hyperparameters setting

Hyper parameters	Value range	Experiment value
Embedding layer units	$[15, 64]$	36
LSTM hidden layer units	$[8, 64]$	64
LSTM hidden layers	$[1, 4]$	3
FC layers units	$[32, 128]$	74
Optimizer	Adam,Rmsprop,SGD	Adam
Activation function	Relu	Relu
Learning rate	$[0.00001, 0.1]$	0.0099

3.4 Results and analysis

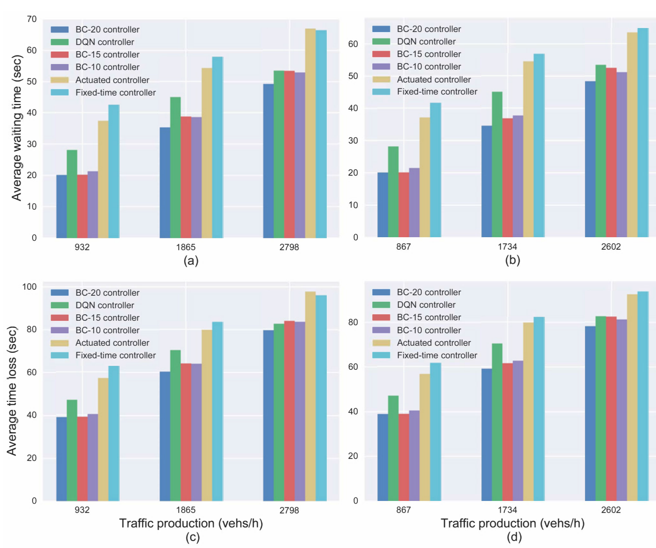

Fig. 4. Performance of six models at different saturation levels. (a) average waiting time on weekdays; (b) average waiting time on weekends; (c) average time loss on weekdays; (d) average time loss on weekends.

Fig. 4 shows the performance of six controllers at different saturation levels in weekdays and weekends. As the experimental result has shown, the waiting time and time loss of all controllers increase while more vehicles enter the road network, which leads to the more complicated traffic situation. BC-20 controller performs better than the fixed-time controller, vehicle-actuated controller and DQN-based controller in all cases. In the cases of low and medium saturation level, all deep imitation learning controllers are significantly better than DQN-based controller, which has the similar performance of deep imitation learning controllers in the cases of high saturation level. Since the time loss is more sensitive to the speed, and the reward of DQN model is the average speed of vehicles passing through the intersection but the virtual demonstration aims at the global waiting time, the advantage of deep behavioral cloning controller under the waiting time indicator is more prominent.

The controller trained with the best virtual demonstration dataset, BC-20, has the best performance over all three deep behavioral cloning controllers, which approves that the demonstration is of great significance for deep behavioral cloning method. For instance, at different saturation levels, the waiting time of BC-20 controller in weekdays is lower than the other two deep behavioral cloning methods 5.96%, 9.84%, 8.43%, respectively.

Table. 3 shows the mean value of in net vehicles number and speed over 24 hours. In terms of in net vehicles number, the BC-20 gets the best result in the cases of low and medium saturation level while DQN-based controller get the best when saturation increases. The BC-20 controller gets the highest average speed overall different experiment settings and the average speed increases 1.58%-11.14%. In general, the proposed deep behavioral cloning method is able to improve the traffic efficiency.

To better understand the experimental results, we draw the macroscopic fundamental fiagrams and average speed for different traffic demand patterns when different controllers are employed. Fig. 5 illustrates that the number of in net vehicles of BC-20 controller is fewer and the average speed is higher than both the other three controllers in the case of low and medium saturation level. It is obvious that the average speed decreases while the rush hour is coming. The deep behavioral cloning controller can keep higher average speed to improve the traffic efficiency at low saturation level and it can handle the rush hour better at medium saturation level. In the case of a high saturation level, a large number of vehicles enter the road network and such a situation last longer time, which leads to a relatively low average speed.

Such experiment results prove that the proposed deep behavioral cloning method can evacuate the waiting time better than other methods. It also does help to increase the average travel speed for the whole road net at different saturation levels. Moreover, better demonstration dataset has obtained better control strategy, indicating that the deep behavioral cloning method has truly learned something from the demonstration and it is able to handle various traffic scenes properly.

4. CONCLUSION AND PROSPECT

In this paper, we design to generate a virtual demonstration for traffic signal control and propose the deep behavioral cloning method to solve traffic signal control problem under the parallel learning framework and get some encouraging results. The simulation experiments verify the effectiveness of our mentod. And it even gets better performance in some criteria compared to one of the most advanced deep reinforcement learning methods. The deep behavioral cloning method can adapt to different environments very well and gets competitive results compared to other methods. Simultaneously, as a supervised

Table 3. Average in-net vehicles number and average speed at different saturation levels

	Traffic production	Average in net vehicle number				Average speed (Km/h)			
	vehs/h	BC-20	DQN	Actu	Fixed-time	BC-20	DQN	Actu	Fixed-time
Weekdays	932	**48.55**	50.80	53.19	54.67	**39.33**	37.93	36.30	35.40
	1865	**108.25**	113.58	118.99	120.72	**36.96**	35.27	33.84	33.25
	2798	172.21	**170.38**	173.58	172.62	**33.26**	32.74	31.12	30.86
Weekends	867	**44.95**	47.07	49.19	50.48	**39.33**	38.08	36.54	35.92
	1734	**99.95**	104.44	109.65	111.34	**37.02**	35.59	34.05	33.42
	2602	159.10	**158.08**	159.15	158.40	**33.88**	33.33	31.94	31.58

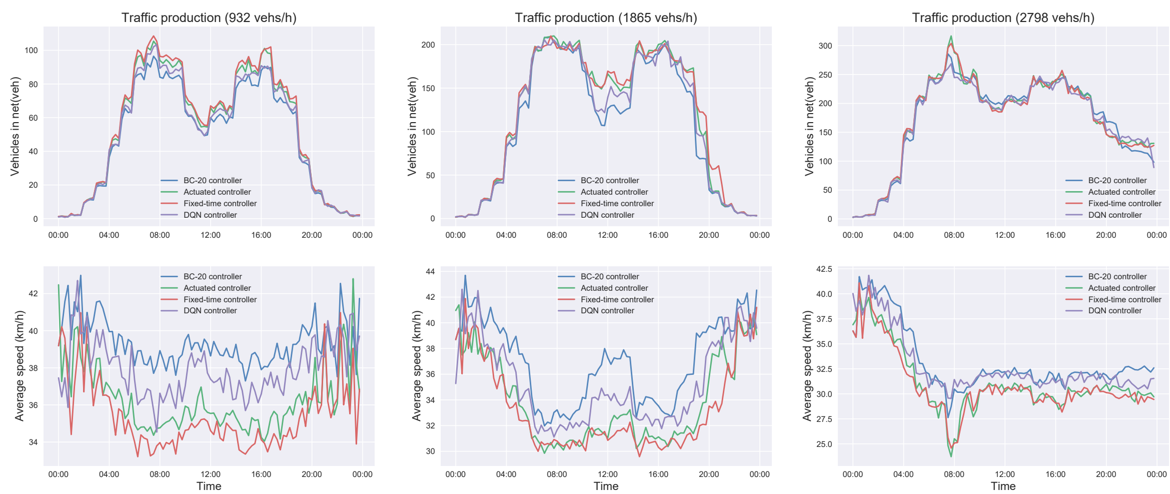

Fig. 5. Average in net vehicle numbers and average speed at different saturation levels in a weekday. The figures in the first row show the number of in net vehicles and the average speed under different controllers in second row.

learning method, this approach has certain advantages in engineering implementation and training stability.

It should be pointed out that there is something deserves more research. Although the BC controller gets the highest speed in the case of a high saturation level, the DQN-based controller gets the fewest in net vehicles number. Fewer in net vehicles means lower congestion probability. We hope that this paper provides a new idea to solve traffic jam problems and combines the deep behavioral cloning method with deep reinforcement learning methods to obtain some achievements in the near future.

REFERENCES

Daganzo, C.F. and Geroliminis, N. (2008). An analytical approximation for the macroscopic fundamental diagram of urban traffic. *Transportation Research Part B: Methodological*, 42(9), 771–781.

Gokulan, B.P. and Srinivasan, D. (2010). Distributed geometric fuzzy multiagent urban traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 11(3), 714–727.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Jin, J., Guo, H., Xu, J., Wang, X., and Wang, F. (2020). An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. doi:10.1109/TITS.2020.2973736.

Kebria, P.M., Khosravi, A., Salaken, S.M., and Nahavandi, S. (2020). Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7(1), 82–95. doi:10.1109/JAS.2019.1911825.

Lv, Y., Chen, Y., Jin, J., Li, Z., Ye, P., and Zhu, F. (2019). Parallel transportation:virtual-real interaction for intelligent traffic management and control. *CHINESE JOURNAL OF INTELLIGENT SCIENCE AND TECHNOLOGIES*, 1(1), 21–33.

Mirchandani, P. and Wang, F.Y. (2005). RHODES to intelligent transportation systems. *IEEE Intelligent Systems*, 20(1), 10–15. doi:10.1109/MIS.2005.15.

Srinivasan, D., Choy, M.C., and Cheu, R.L. (2006). Neural networks for real-time traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 7(3), 261–272.

Wang, F. (2010). Parallel Control and Management for Intelligent Transportation Systems: Concepts, Architectures, and Applications. *IEEE Transactions on Intelligent Transportation Systems*, 11(3), 630–638. doi:10.1109/TITS.2010.2060218.

Wang, F.Y., Wang, X., Li, L., and Li, L. (2016). Steps toward parallel intelligence. *IEEE/CAA Journal of Automatica Sinica*, 3(4), 345–348.

Yoo, S., Haider, M.A., and Khalvati, F. (2017). Estimating optimal depth of vgg net with tree-structured parzen estimators. *Journal of Computational Vision and Imaging Systems*, 3(1).