Deep Imitation Learning for Traffic Signal Control and Operations Based on Graph Convolutional Neural Networks

Xiaoshuang Li[1,2], Zhongzheng Guo[1,3], Xingyuan Dai[1,2], Yilun Lin[1], Junchen Jin[1,4], Fenghua Zhu[1,*], Fei-Yue Wang[1]

Abstract— **Traffic signal control plays an essential role in the Intelligent Transportation Systems (ITS). Due to the intrinsic uncertainty and the significant increase in travel demand, in many cases, a traffic system still has to rely on human engineers to cope with the complicated and challenging traffic control and operation problem, which cannot be handled well by the traditional methods alone. Thus, imitating the good working experience of engineers to solve traffic signal control problems remains a practical, smart, and cost effective approach. In this paper, we construct a modelling framework to imitate how engineers cope with complex scenarios through learning from the historical record of manipulations by traffic operators. To extract spatial-temporal traffic demand features of the entire road network, a specially designed mask and a graph convolutional neural network (GCNN) are employed in this framework. The simulation experiments results showed that, compared with the original deployed control scheme, our method reduced the average waiting time, average time loss of vehicles, and vehicle throughput by 6.6%, 7.2%, and 6.85%, respectively.**

I. INTRODUCTION

Over the past few decades, more and more vehicles have come into our lives. As a consequence, traffic congestion and traffic accidents are worsening [1]. Solving the problem of traffic congestion has become an urgent task. Obviously, an efficient and intelligent traffic signal control system is a necessary and critical part of any solution to current traffic congestion problems [2], [3].

In most cases, a traffic signal control scheme can be divided into two parts: signal phase and signal timing. Since a signal phase is stable, most developed traffic signal control methods aim at optimizing signal timing. In engineering practice, most traffic signal lights are still controlled by predefined fixed-time plans [4]. This kind of methods performs well when the traffic demand is steady. But they cannot handle the increasingly complex and changing traffic demands. To tackle the problem, modelling-based methods are proposed to reduce traffic congestion [5], [6]. But the modelling-based methods rely heavily on the model. Whether the model can accurately describe the environment seriously affects the performance of the method. Nowadays, the traffic scene is becoming more and more complicated, accurate modelling of traffic scene costs much and still remains a big challenge [6]. Aiming at difficult modelling problems, parallel intelligence provides us with some new methods to deal with modelling complex system and has been used to solve traffic control problems [7], [8].

The simulation-based traffic control method is another widely used approach and it has received more and more attention. Since data in the simulation environment is cheap and the traffic demand can be changed according to our requirements, it has been used to verify different methods. Deep reinforcement learning (DRL) methods develop rapidly [9], [10] and with the development of traffic simulation software, deep reinforcement learning methods have been applied to traffic signal control problems [11], [12]. These methods take the traffic signal light as an agent and the agent interacts with the traffic scenarios to get a higher reward. However, at present, these methods also assume a relatively stable traffic flow and a small road network scale, which is far from the actual traffic scene [13].

The adaptive traffic signal control method adjusts the control scheme by detecting the changes of traffic state [14]. It is more intelligent than the fixed-time schemes and has spawned many new methods. The behaviour of vehicles in the road network has a great influence on the traffic conditions and the vehicle-actuated method [14] could make use of real-time traffic state information. But this method needs hand-craft rules for different traffic states, and the rules need human engineers to update when the pattern of traffic demand changes. Thus, it is hard for the traffic control system to cope with the changing situation. The coming of Internet of Things (IoT) and big data era has done a great help to many different problems [15], [16]. With the help of different kinds of data, social transportation [17] can generate more flexible and robust traffic control strategy. Nevertheless, this approach also faces many challenges, such as dependence on data, failure in a complex scenario and so on.

In some metropolis, human engineers have been monitoring traffic conditions for saturated urban traffic networks. When traffic congestion occurs, human engineers will take the place of the original control strategy and manually

Research supported in part by the National Key R&Development Program of China under Grant 2018YFB1700202, NSFC U1811463, U1909204, 61773381, 61533019, Basic and Applied Basic Research Fund of Guangdong Province 2019B1515120030.

[1]X. Li, Z. Guo, X. Dai, Y. Lin, J. Jin, F. Zhu and F.-Y. Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (Email:{lixiaoshuang2017, guozhongzheng2017, daixingyuan2015, linyilun2014, fenghua.zhu, feiyue.wang}@ia.ac.cn).

[2]X. Li and X. Dai are also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China.

[3]Z. Guo is also with Harbin University Of Science And Technology, Harbin, 150080, China.

[4]J. Jin is also with Enjoyor Co., Ltd. Hangzhou 310030, China. Email: junchen@kth.se.

*Correspoinding author, Email: fenghua.zhu@ia.ac.cn

control the traffic signal lights [3]. According to a long-term operation experience, human engineers are required to handle extremely complex traffic situation, which is hard to be handled by the original methods. Fortunately, obtaining data has become much easier nowadays [18] and how human engineers deal with traffic jams can also be documented. Learning some useful information from the historical data and imitating the experience of human engineers is a feasible method to cope with complex and mutable traffic demands. We hope such hybrid-augmented human-like methods learn and think like human engineers [19] and could replace human engineers to some extent.

Therefore, we propose a deep imitation learning method to model complex traffic phenomena using the Graph Convolution Neural Networks (GCNNs) technique [20], [21], and complete the traffic signal control operation task. We focus on a road network that consists of more than ten connected intersections. Since the change of traffic state in a single intersection can influence its neighbourhoods even the entire region, we try to alleviate traffic jams in the road network and improve traffic efficiency by utilizing the topology property of the road network. A deep neural network model, which contains GCNNs, has been adopted to extract spatial-temporal traffic demand features and imitate traffic engineers to control the traffic signal lights in the road network. Due to the data comes from actual traffic scenarios, there is no guarantee that all detectors are functioning at every moment. In fact, the worst case we have encountered is that only 3 of the 16 detectors are working correctly. Such an issue is also addressed in the processed approach by a specially designed mask and it allows incomplete data to be used by the model.

The main contributions of this paper are summarized as follows.

- We propose a deep imitation learning model to control the traffic signal lights. This model includes a graph convolution neural network, and utilizes the topology property of the road network. Our deep imitation learning model can learn from the experience of human traffic engineers.
- The data obtained from the actual traffic scenarios is not perfect and we solve this problem by using a specially designed mask and preprocess network. We believe that such a method is not only suitable for traffic data, but also other tasks.
- We compare the proposed method with the original deployed methods in actual traffic scene, the simulation experiments results show that the proposed method has superior performance.

The rest of this paper is organized as follows: Section II introduces the details of our method used in this paper. Section III presents the experimental configuration, results and analysis. Finally, we conclude the paper in section IV.

II. Deep Imitation Learning Method

To imitate the experience of traffic engineers, we established a deep neural network model, which contains a graph convolutional neural network and preprocessed the original data. In this section, we first have a brief introduction to GCNNs. It is one of the most essential parts of our deep imitation learning model. Then, we describe the detail about how to deal with missing data. After that, the model structure is deccribed, and finally, we clarify how our imitation learning model learns from human engineers.

A. *Graph Convolutional Neural Networks (GCNNs)*

GCNNs are proven to be suitable for dealing with the variable input data (e.g., signal or feature values) given on general graphs [22], which is different from the structural graph, for example, a picture. A graph is a pair $G = (\mathcal{N}, E)$, where $\mathcal{N}$ is a set whose elements are called vertices or nodes, and E is a set of links between any two nodes, called edges. Graphs can be used to represent knowledge and many other discrete structures. The road network can be regarded as a graph, every intersection can be considered as a node and the traffic volume and other information in the intersection comprise the feature value for the node. If there is a road between two intersections, it can be regarded as an edge between nodes. We established a connected undirected graph for all intersections within the study, shown as Fig. 1.

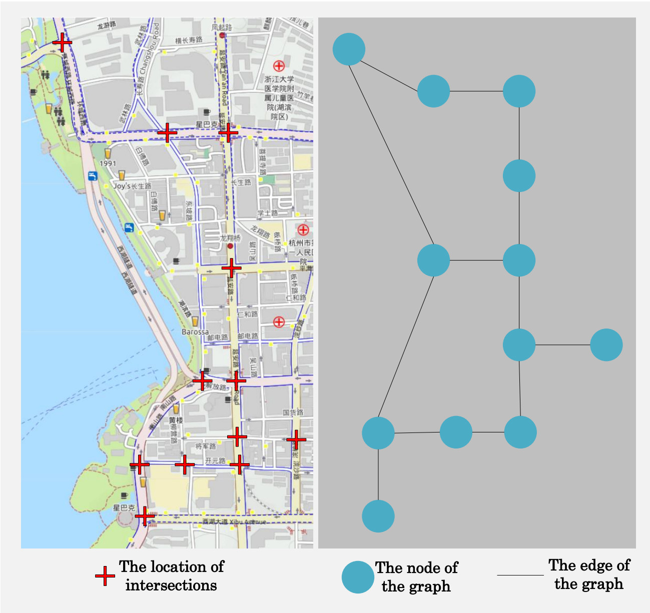

Fig. 1: The road network and the graph

GCNNs help the model to capture some critical information behind the historical data of the whole road networks. Vehicles travel from the upstream intersections to the downstream intersections, making the state of the road network change. This process is very similar to the dissemination of information in a graph. Eq.(1) shows how information is transmitted between different layers of GCNNs.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{1}$$

The output of $l+1$-th layer, $H^{(l+1)}$, is calculated with the output of the l-th layer $H^{(l)}$, where σ is the non-linearity function, W is the weight matrix for this layer. D represents the degree matrix of the graph, A is the adjacency matrix and $\tilde{D} = D+I$ and $\tilde{A} = A+I$. Adding a self-connection to each node of the graph and building the corresponding degree and adjacency matrix are renormalization tricks. $H^{(0)}$ is $N \times D$, where N is the number of nodes and D is the number of input features. In this paper, the number of nodes equals the number of intersections and the number of features is one of the hyperparameters.

B. Mask for missing data

Poor data quality is a problem we need to address in engineering practice. But this problem is not easy to be solved. Due to sensor failures, data transmission errors, and other issues, some of the information we need was lost. Data quality changes dynamically. Common data cleaning methods choose to drop dirty samples. But the number of samples is vital for the generalization of the deep learning-based method. Although we can train the model with a dataset consisting of perfect samples, the missing data always exists in the actual situation. The model has to adapt to real traffic scenarios.

A mask is used to reduce the impact of missing data. In actual traffic scenarios, which sensors fail are not fixed and the dimension of the data changes often. The model needs to know where it has changed and take the changes into consideration. Moreover, the number of lanes and detectors at an intersection keep stable for a long time. Therefore, we assume that all the sensors are working properly and fill the return values of all the non-functioning sensors with zero. At the same time, a flag bit number is generated for each sensor to indicate whether the sensor is working correctly. The flag bits numbers consist of a mask vector for current sensor data, and its value can influence the output of the model. Thus, the data dimensions are twice the number of sensors and every intersection has a certain feature number.

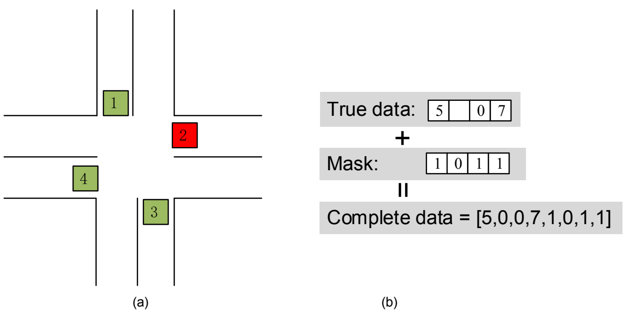

Fig. 2: Mask for incomplete data sample. (a) An intersection with four detectors and the second detector in red does not work; (b) The true data is obtained from the detectors, the mask data is the corresponding set of flag bits and zero in second bit represents the data is missing.

Fig. 2 shows an example of how to handle missing data in this method. If a sensor works well, it will return the correct information (such as the speed of a vehicle, and the value is allowed to be zero). The corresponding flag bit will be set to one; if it does not work, it will return nothing. We replace the empty value with zero to keep the shape of the data. Simultaneously, the corresponding flag bit will be set to zero. All these flag bit data constitutes a mask vector, and it will be part of the final complete data. By concatenating the original data and the mask vector, we get the final complete data and the deep imitation learning model uses it as the input data. Such a vector can describe the traffic state and sensor working state at the same time.

C. The model structure

Fig. 3 shows the structure of the deep imitation learning model. In order to deal with the difference in the number of lanes at different intersections, multiple deep neural networks are constructed at the beginning and end of the model to process the data and generate a specific signal control scheme for each intersection. The outputs of the beginning part have the same shape, which could be accepted by the GCNNs. The final outputs of the model are restored to a different shape to meet the requirements of a single intersection.

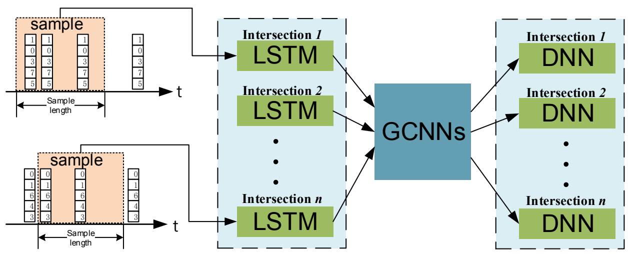

Fig. 3: Deep imitation learning model structure, n is the number of intersections.

Short-term actual traffic conditions have a greater impact on congestion. Therefore, we use the data of the past 30 minutes to control the traffic light. Another troublesome fact is that the actual historical data is not collected according to a fixed time step length, so the length of the sample sequence is not fixed. Fortunately, there are some methods to overcome difficulty. In the header of the model, we use the long-short term memory (LSTM) [23] network, which can handle variable-length sequences to extract temporal feature from historical data. GCNNs use the output of LSTM networks as input feature data. The GCNNs can link the intersections and analyze the data of the entire network as a whole. At the end of the model, the different intersection has different deep neural networks to generate a unique traffic signal control scheme.

D. Deep imitation learning method for traffic signal control

Imitation learning provides a way to find a better strategy by learning the knowledge of human experts in a certain field [24]–[27]. In the field of traffic signal control, human engineers can directly control the traffic lights when the performance of the deployed method is not good enough. Engineers determine when and how to adjust control strategies by observing current traffic state and analyzing historical data, even making some prediction for the traffic situation. The operation records provide a large number of learnable samples for imitation learning model.

With the help of a variety of sensors, we are able to obtain various data on traffic status. It is natural to divide the data into two parts: state data and strategy data. State data s_i contains the number of vehicles in each lane, speed, delay time and other available information. It describes the traffic scene. The strategy data a_i decides how to control the traffic signal lights and it includes the phase split and cycle length. There is a sequence $< s_1, a_1, s_2, a_2, ..., s_i, a_i >$, which record the dynamic change and response, i is the time step. If we construct a set like $D = \{(s_1, a_1), (s_2, a_2), (s_3, a_3), ..., (s_i, a_i)\}$, we can call the state data s_i feature and call the strategy data a_i label. In this paper, s_i and a_i are both vectors but the length of the vectors varies from each other because the number of lanes at different intersections is different. Thus, our traffic signal control problem becomes a classical supervised learning problem here. And the target of our deep imitation learning model is to finish a regression problem [24]. After the model has been well trained, the state data is the only requirement to use the model to control the traffic lights.

The state data used in this paper contains traffic flow data, which was collected from the SCATS (Sydney Coordinated Adaptive Traffic System [4]) system. And the strategy data is a historical control scheme record that includes the split and cycle length (seconds) for every intersection in the road network. Our training loss function is the mean square error (MSE), as shown in Eq.(2). Split and cycle time have different value range, so we divide them into two parts in the loss function.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(p_i - \hat{p_i})^2 + \alpha\frac{1}{N}\sum_{i=1}^{N}(q_i - \hat{q_i})^2 \qquad (2)$$

where p_i, q_i represent the true historical phase split data and cycle time, $\hat{p_i}$, $\hat{q_i}$ represent the split and cycle time given by the imitation learning model. Because the value range of phase split and cycle time varies from each other, the penalty factor, α, is necessary to balance the difference.

III. Experiments and analysis

In this section, we conduct simulation experiments using real-world traffic data collected in Hangzhou, China and show great improvement in some criteria compared with the original employed scheme, which was based on the fixed-time method. We construct our whole experiments on a simulation platform SUMO 1.1 (Simulation of Urban Mobility) [28]. The python API of SUMO helped us to simulate the operation of traffic control and microscopic vehicles in the road network.

A. Dataset and evaluation criteria

Traffic demand data is of great significance for traffic control. To make our approach more practical, we use data which is collected from real traffic scenes to generate the simulation traffic demand. By adjusting the relevant parameters, which determine how to generate vehicles in the simulation environment, we make the traffic demand generated in SUMO and the real traffic flow as consistent as possible.

The demonstration data set comes from traffic engineers. In this paper, we mimic the historical traffic signal control data of weekdays during 2018.7.1-2018.7.31 in Hangzhou, Zhejiang province. In real data, the time interval is uncertain, ranging from 1 minute to 3 minutes. All historical traffic state data and historical control data are processed using the same 0-1 normalization method. Since there will be a large amount of missing data in the dataset, we use the method proposed in section II to generate a mask that can describe the missing data, and finally get a data set that contains 7686 effective samples for training the deep imitation learning network. In the test stage, we also used the same method to preprocess the data obtained from the simulation environment.

The performance of our method is evaluated under three different criteria. The first one is the average waiting time as Eq.3. It is an important criterion that 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.

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

The second criterion is the average time loss. This criterion, shown as Eq.(4), describes the gap between the ideal time and actual time the vehicle spends to arrive at its destination. The time it takes for a vehicle to travel from start to end at a free-flow speed is the ideal time. 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. Using more than one criterion can be more comprehensive.

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

Another criterion is the vehicle number in the road network. It describes the occupation of road resources by vehicles. More vehicles in the road network, vehicles have fewer road resources on average and higher risk of traffic jam.

B. Hyperparameters set

Hyperparameters play an essential role in the experiments and they have a direct impact on the final experimental results. A suitable set of hyperparameters can shorten training time and improve model performance. All hyperparameters used in this paper are listed in Table I.

There are 12 interconnected intersections within the scope of this work, so we use the same number of nodes in the GCNNs, and other parameters are specified by grid search.

TABLE I: hyperparameters setting

Hyperparameters	Experiment value
Activation function	Relu
Number of hidden layers in LSTM	2
Number of units in LSTM layers	10
Number of GCNN layers	2
Number of feature numbers for GCNN	64
Learning rate	0.001
Penalty factor	10

C. Results and analysis

We compare the performance of the imitating learning model with the method actually used in the field. Fig. 4 shows the average waiting time and average time loss of two different schemes with the different simulation end time. In the cases of four different simulation end times, the imitating learning scheme has all achieved better performance than the original scheme in these two indexes. In the case of 6 hours simulation, the number of vehicles in the road network was relatively small and the change was relatively stable. The average waiting time and average time loss decreased by 42.8% and 30.6%, respectively. In the case of full-day simulation, the morning and evening traffic peaks and other periods are simulated. The average waiting time and average time loss decreased by 3.42 seconds and 6.25 seconds, respectively, and the performance increased by 6.6% and 7.2%, respectively. The other two cases showed the same comparison results.

Decreased waiting time means less time in a congested state, and less time loss means vehicles travel faster in the road network. Since the traffic demand in the simulation environment is highly consistent with the actual traffic scenario, different simulation end time represents different flow patterns. The 6-hour simulation represents low traffic demands, while the 24-hour is close to real-world scenarios. The experiment results show that our deep imitation learning model performs better in both situations. Our method can greatly reduce waiting time and time loss when the situation is relatively simple; when the situation becomes complicated, it can also effectively improve traffic efficiency.

Fig. 5 shows the vehicle numbers in the road network. Under the original scheme, the total number of vehicles in net throughout the day is 2,728, which is 2,541 under the deep imitation learning scheme, a decrease of 6.85%. Fewer in-net vehicles mean fewer vehicles occupy the road, resulting in less crowding. This phenomenon also means higher travel efficiency. Among the data counted every two hours, the number of vehicles under the deep imitation learning method is significantly fewer than the original method, except the 18:00-20:00, which is almost equal to each other. Such a result suggests that the peak time is more difficult to handle. According to the truth that evening peak congestion often occurs in Hangzhou, one of the possible reason is the number of vehicles is too much for the system. The adaptive system has lost room for adjustment.

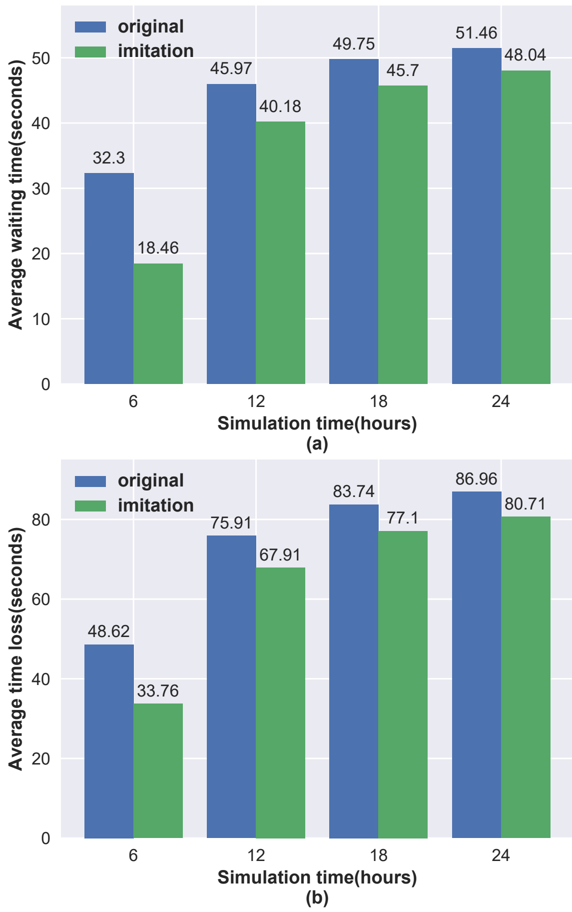

Fig. 4: Performance of original scheme and imitation learning scheme with different simulation end time. The original scheme is used in the actual traffic scenes and the deep imitation learning scheme comes from our method. (a) Average waiting time of different schemes. (b) Average time loss of different schemes.

IV. Conclusion and Prospect

In this paper, we proposed a deep imitation learning method for traffic signal control problems. Inspired by the excellent performance of the GCNNs on the data with the graph structure, we combined GCNNs and traffic scenarios to mine the information in the topological property of the road network. We also proposed a specially designed mask to handle the missing data in engineering practice. By learning the experience of human engineers, we can integrate human knowledge and experience into artificial intelligence methods, so that our method can adjust traffic control strategies in real-time according to traffic conditions. In this way, we are able to control complex traffic scenarios better while reducing the workload of human engineers. Experiments verified the effectiveness of our method and it showed superior performance than the original deployed sheme.

In this study, we found that the existing intelligent traffic signal control methods usually only focus on the traffic state

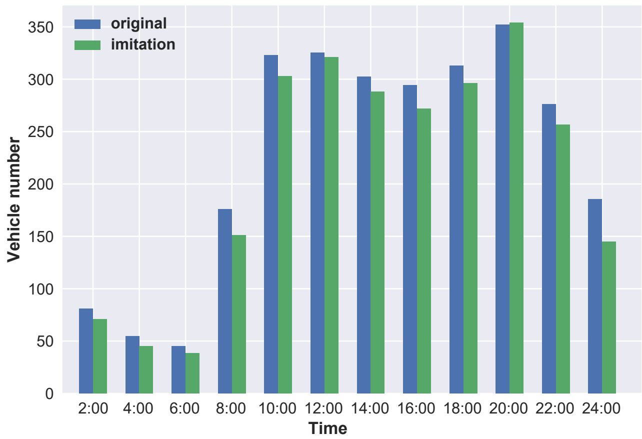

Fig. 5: Vehicle numbers in net of every two hours.

data, but ignore the human experience. Although human experts can handle some complicated traffic control problems well, there are few studies about how to mimic their experience. We believe that building a hybrid-augmented smart approach will bring about tremendous change. This article has made some attempts and shown good prospects. In the future, we plan to improve the performance of our method in peak time and combine our method with deep reinforcement learning algorithms to get better results.

References

[1] D. Zhao, Y. Dai, and Z. Zhang, "Computational intelligence in urban traffic signal control: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 485–494, 2012.

[2] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2020.

[3] J. Jin, H. Guo, J. Xu, X. Wang, and F. Wang, "An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[4] J. C. Aydos and A. O'Brien, "Scats ramp metering: strategies, arterial integration and results," in *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 2194–2201.

[5] B. Ye, W. Wu, K. Ruan, L. Li, T. Chen, H. Gao, and Y. Chen, "A survey of model predictive control methods for traffic signal control," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 623–640, May 2019.

[6] L. Li, D. Wen, and D. Yao, "A Survey of Traffic Control With Vehicular Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 425–432, Feb. 2014.

[7] L. Li, Y. Lin, D. Cao, N. Zheng, and F. Wang, "Parallel learning — a new framework for machine learning," *Acta Automatica Sinica*, vol. 43, no. 1, pp. 1–8, 2017.

[8] G. Xiong, X. Dong, H. Lu, and D. Shen, "Research progress of parallel control and management," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 355–367, 2019.

[9] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2019.

[10] T. Liu, B. Tian, Y. Ai, and F. Wang, "Parallel reinforcement learning-based energy efficiency improvement for a cyber-physical system," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 617–626, 2020.

[11] Y. Lin, X. Dai, L. Li, and F.-Y. Wang, "An Efficient Deep Reinforcement Learning Model for Urban Traffic Control," *arXiv:1808.01876 [cs, stat]*, Aug. 2018, arXiv: 1808.01876. [Online]. Available: http://arxiv.org/abs/1808.01876

[12] L. Li, Y. Lv, and F. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, July 2016.

[13] M. Guo, P. Wang, C. Chan, and S. Askary, "A reinforcement learning approach for intelligent traffic signal control at urban intersections," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 4242–4247.

[14] B. Pratama, J. Christanto, M. T. Hadyantama, and A. Muis, "Adaptive traffic lights through traffic density calculation on road pattern," in *2018 International Conference on Applied Science and Technology (iCAST)*, Oct 2018, pp. 82–86.

[15] L. Cheng, J. Liu, G. Xu, Z. Zhang, H. Wang, H.-N. Dai, Y. Wu, and W. Wang, "SCTSC: A Semicentralized Traffic Signal Control Mode With Attribute-Based Blockchain in IoVs," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1373–1385, Dec. 2019.

[16] R. Munjal, W. Liu, X. J. Li, and J. Gutierrez, "Big data offloading using smart public vehicles with software defined connectivity," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 3361–3366.

[17] F.-Y. Wang, P. Wang, J. Li, Y. Yuan, and X. Wang, "Social Transportation: Social Signal and Technology for Transportation Engineering," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 2–7, Feb. 2019.

[18] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan 2018.

[19] N. Zheng, Z. Liu, P. Ren, Y. Ma, S. Chen, S. Yu, J. Xue, B. Chen, and F. Wang, "Hybrid-augmented intelligence: collaboration and cognition," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 2, pp. 153–179, 2017.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[21] Z. Li, G. Xiong, Y. Chen, Y. Lv, B. Hu, F. Zhu, and F. Wang, "A hybrid deep learning approach with gcn and lstm for traffic flow prediction*," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 1929–1933.

[22] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in neural information processing systems*, 2015, pp. 2224–2232.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[24] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 661–668.

[25] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.

[26] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, p. 21, 2017.

[27] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 82–95, January 2020.

[28] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.