This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Data Augmented Deep Behavioral Cloning for Urban Traffic Control Operations Under a Parallel Learning Framework

Xiaoshuang Li, Peijun Ye, Junchen Jin, *Member, IEEE*, Fenghua Zhu, *Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*— It is indispensable for professional traffic signal engineers to perform manual operations of traffic signal control (TSC) to mitigate traffic congestion, especially with complicated scenarios. However, such a task is time-consuming, and the level of congestion mitigation heavily relies on individual expertise in engineering practice. Therefore, it is cost-effective to learn traffic engineers' knowledge to enhance the problem-solving skills for a large-scale urban traffic network. In this paper, a data augmented deep behavioral cloning (DADBC) method is proposed to imitate the problem-solving skills of traffic engineers. The method is under a conceptual framework, parallel learning (PL) framework, that incorporates machine learning techniques for solving decision-making problems in complex systems. The DADBC method enhances a hybrid demonstration by exploiting a generative adversarial network (GAN) and then uses the deep behavioral cloning (DBC) model to learn traffic engineers' control schemes. According to the validation results using the real manipulation data from Hangzhou, China, our method can imitate complex human behaviors in intervening traffic signal control operations to improve traffic efficiency in urban areas.**

***Index Terms*— Intelligent traffic signal operations, parallel learning, deep behavioral cloning, generative adversarial networks.**

Manuscript received August 4, 2020; revised November 2, 2020; accepted December 17, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101502; in part by the National Natural Science Foundation of China under Grant U1811463, Grant 61533019, and Grant 62076237; and in part by the Youth Innovation Promotion Association, Chinese Academy of Sciences. The Associate Editor for this article was F. Qu. *(Corresponding author: Junchen Jin.)*

Xiaoshuang Li is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: lixiaoshuang2017@ia.ac.cn).

Peijun Ye and Fenghua Zhu are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Parallel Intelligence Research Center, Qingdao Academy of Intelligent Industries, Qingdao 266109, China (e-mail: peijun.ye@ia.ac.cn; fenghua.zhu@ia.ac.cn).

Junchen Jin is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with Enjoyor Co., Ltd., Hangzhou 310030, China (e-mail: junchen@kth.se).

Fei-Yue Wang is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, also with the Center of China Economic and Social Security, University of Chinese Academy of Sciences, Beijing 100149, China, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: feiyue.wang@ia.ac.cn).

Digital Object Identifier 10.1109/TITS.2020.3048151

I. Introduction

ALTHOUGH intelligent decision approaches in the field of urban traffic control and management have made great strides [1]–[3], operations for large-scale urban traffic signal control (TSC), especially for saturated networks in metropolis are remain big challenge [4]. Professional traffic signal engineers are still by far the widely applied solvers of complex traffic congestion problems in intervening traffic signal control operations. They are required to cope with the uncertainty and complexity of the city traffic system when the control system, for instance, SCATS [5], can not handle it well. Professional traffic signal engineers can quickly and flexibly adjust the control scheme according to the actual situation and historical experience to ensure that the scheme deployed is the most suitable for the current situation. The knowledge and experience they have gained during their careers have equipped them with the ability to handle complex congested situations.

However, since manual operations are time-consuming and labor-intensive, not all of the congested intersection can be always well handled due to the limited human capabilities, particularly for a large-scale urban network in rush hours. To address the issue above, an essential and urgent task is to develop a software-aided tool that can learn from human experts and provide human-like schemes within a short time to accelerate the operational process of traffic control and management.

The stochastic and uncertainty properties in human knowledge for traffic control operations are difficult to be captured [6] with analytical modeling approaches. A data-driven approach [7], [8] has shown great power and been used to solve various modeling problems [9], [10]. It also has been proved as a promising candidate for imitating human behaviors [11]. Specifically, imitation learning (IL) [12]–[14] is one of the most popular methods to learn the knowledge of human experts from the demonstrations. Previous studies, such as the game of Go [15], self-driving [11], [16], robot [17], and computer vision tasks [18], have illustrated that human knowledge and experience involved is beneficial to the complex problem-solving process, and meanwhile, IL can capture human knowledge and utilize the learned knowledge to avoid inefficient exploration and speed up the complex operational process. IL-based methods deserve more attention

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

in solving complex TSC operations, which are commonly carried out by humans.

In an IL application, collecting a large number of demonstration samples is laborious, but sufficient data is essential for training a neural networks-based high-performing model. In this paper, a DADBC method is proposed to solve the problem of insufficient demonstration and imitate the professional traffic engineers' behaviors for TSC scenarios. The proposed DADBC approach enables the control system to provide timely and high-performance control schemes in this study. This approach is formulated under the concept of parallel learning framework [19], [20], and integrate a deep behavioral cloning method [21] with generative adversarial networks [22].

The remainder of this paper is organized as follows: In Section II, we review the related work in TSC systems and briefly introduce the development of the generative adversarial model and deep behavioral cloning method. Details of the method proposed in this paper are presented in Section III. In Section IV, a case study from actual transportation is investigated to verify the effectiveness of the method. The article concludes in Section V, with some future research directions in this area.

II. Related Works

In the literature, human expertise is modeled by a spectrum of IL approaches [12]. Behavioral cloning is one popular form of IL in which knowledge and experience are rapidly acquired through demonstrations [11], [13], [23]. Xiong *et al.* and Huo *et al.* applied a behavioral cloning approach for learning the self-organizing traffic light control method and a rule-based traffic signal control method, respectively [24], [25]. Both of these two previous studies have achieved excellent performance in enhancing traffic efficiencies. Nevertheless, these two methods are not flexible for human experts since they are both learned from the rule-based policy. Thus, the rule-based policy trajectories are not the most appropriate demonstration for a BC method. At the same time, the limited demonstrations are only a subset of the traffic state-action space, which is not conducive to finding the optimal policy. This brings the necessities of a data augmentation process.

GAN [22] and its variants [26]–[29] have been spotlighted as data augmentation methods, which are capable of generating virtual data with distribution being consistent with the real data. Specifically, the original GAN model achieves this goal by minimizing the Jensen-Shannon divergence between these two distributions. Least Square GAN (LSGAN) [26] improves the generating capacity of the model by changing the cross-entropy loss to mean square loss. To tackle the issue of mode collapses, Wasserstein GAN (WGAN) [27] proposed to use Wasserstein distance to replace the original loss function in GAN and combined with the technique of weight clipping. It greatly reduced the number of mode collapse. As an effective improvement of WGAN, WGAN with gradient penalty (WGAN-GP) [28] effectively avoids the disadvantages caused by the weight clipping through the gradient penalty. Recently, some new methods, such as an ensemble method [30], attention module [29], [31], have also been used to improve the performance.

Due to the excellent performance of data generation, GAN models have attracted extensive attention [32], [33]. Many studies pay attention to data augmentation with GANs to solve the problem of insufficient data size or unbalanced data categories [34], [35]. For example, because of the significant category imbalance in medical images, the data augmentation technique is essential for a reliable expert system. In the classification task of liver image data, a GAN model is leveraged to enhance the image data to obtain higher classification accuracy [36]. GAN is also used to enhance the brain CT image to improve segmentation accuracy [37]. In the field of self-driving, intelligent vehicles must be able to identify traffic signs to keep safe. To ensure that self-driving cars are competent for the task, traffic signs from different environments are imperative to train a robust classifier. However, it is difficult and expensive to satisfy this requirement. The data augmentation method makes it possible to improve the safety and comfort of self-driving vehicles. Recently, to improve traffic sign detection performance, GAN and deep reinforcement learning methods are used to generate night-time vision, poor lighting, and several other visual occlusions [38].

III. Methodology

A. Traffic-Control Operational Behavioral Modeling

To model the knowledge and experience of traffic engineers, we present a DADBC traffic control operations method under the parallel learning (PL) framework [19]. Three processes are specified in the PL framework: descriptive learning, predictive learning, and prescriptive learning. The descriptive learning models abstract systems to extract domain knowledge and experience from "big" artificial data. The predictive learning establishes a mapping from the collected information to unknown information. The prescriptive learning further optimizes the policy learned, based on the first two processes, and guides the system to ensure better performance. Specifically, the predictive learning process is used to solve the problem of insufficient human operation record data by using a large amount of generated virtual data. Then, the descriptive learning process is implemented to distill knowledge and experience from the enhanced data and imitate professional human traffic engineers.

Fig. 1 presents the workflow of the proposed DADBC method under the PL framework, containing two core blocks: the real traffic system and the DADBC module. A real traffic system is depicted by multi-dimensional data sources collected from professional traffic engineers and various traffic infrastructure. Since the manual operation records are limited compared with the traffic state data, a relatively "small" dataset is constructed after matching operational records and traffic state information. In the "small" dataset, "state" and "scheme" represent the traffic state and control scheme, respectively. Since traffic engineers only intervene in the regulation process when there are abnormal situations such as congestion, "label" are used to mark whether each <state-scheme> pair comes from professional traffic engineers. It is marked as "True" when the schemes come from the traffic engineers and as "False" otherwise. Using the data collected from a real traffic

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

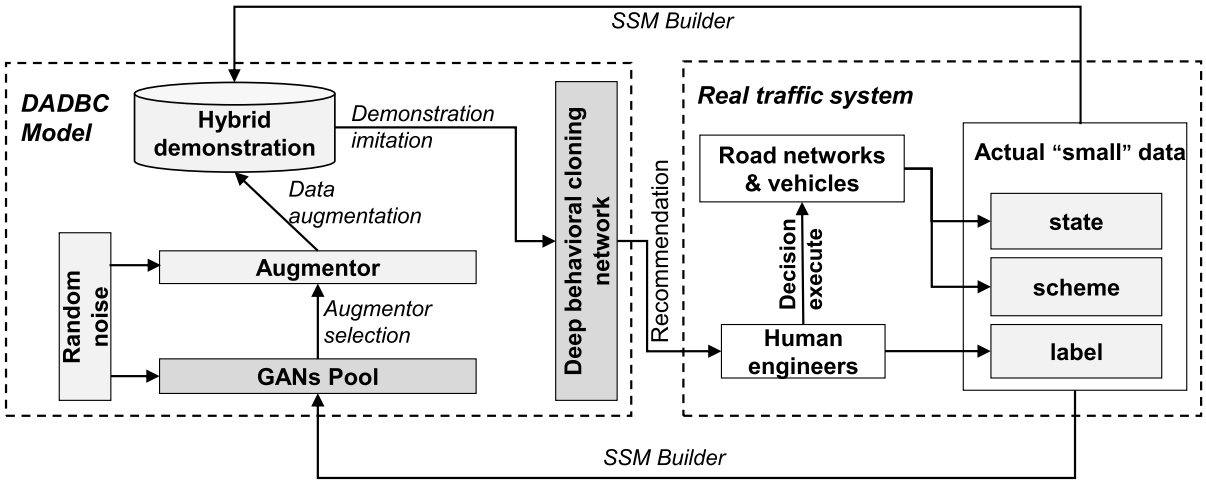

Fig. 1. The technical framework of the data augmented deep behavioral cloning urban traffic control operations model.

system, a GAN-based model is applied in the context of the predictive learning process to augment the actual data with a relatively small size. Thereafter, the augmented data prevents neural networks from overfitting and improves the performance of the deep behavioral cloning (DBC) model. The well-trained DADBC model can generate a unique scheme for each specific traffic situation. The output scheme will be used as a candidate to traffic engineers, who will evaluate the scheme based on their own observations and experience to decide whether to accept the scheme. The recommendation system will employ the scheme if it is approved by the traffic engineers; otherwise, the model needs to be optimized to provide a better solution.

The DADBC module consists of three major steps: hybrid demonstration construction, augmentation model selection, and demonstration imitation. The proposed model first obtains the control operation record (<state-scheme> pair) from the real traffic system. Before the collected data are used to train the GAN models, the state-scheme map (SSM) builder integrates the state and scheme information and then organizes the data in a particular form. An application-adapted GAN model is picked from the GAN pool by the augmentation selection and thereafter the random noise is used as the input to enhance the actual "small" dataset to construct a hybrid demonstration, from which the DBC network distills the knowledge and experience of professional traffic engineers. Finally, the recommended scheme will be generated by the well-trained DADBC model and the traffic engineers make their decisions to operate the traffic control system. Alg. 1 summarizes the operational process of the DADBC module.

In general, the SSM should reasonably obtain necessary information from both traffic states and control schemes in a time-window. Analogical to the definition of image data in the computer vision field, the SSM (Fig. 2) is designed as a multi-dimensional tensor, including a state channel and a scheme channel. Since an intersection with four arms is a universal form of the real traffic cross, each channel is divided into four corners representing data collected from four different directions, i.e., northbound, southbound, eastbound, and westbound. There are nine different features in each of these four directions, donated by $F1, F2, \cdots, F9$ and $f1, f2, \cdots, f9$, to describe the traffic state and control scheme. Note that the data corresponding to the non-existent entrance is set to be zero. Through the arrangement of each feature value in the tensor, the spatial characteristics of different directions can also be considered. Moreover, it is also easy to decompose the <state-scheme> pair from the SSM sample, which conveniently serves subsequent imitation learning processes.

Algorithm 1 Data Augmented Deep Behavioral Cloning Algorithm

1: Collect the actual data D_{true} from the real traffic system;
2: **for** each model $\mathcal{G}$ in GANs pool **do**
3: Train the model $\mathcal{G}$ with D_{true};
4: Leverage the trained $\mathcal{G}$ to generate virtual data D_{fake};
5: Mix D_{true}, D_{fake} and construct a hybrid demonstration D_{Demo}
6: Train the DBC model $\mathcal{B}$ with D_{Demo} and evaluate it on the test set;
7: Store the test result and the model:$\mathcal{M} = \mathcal{G} + \mathcal{B}$
8: **end for**;
9: Select $\mathcal{M}^*$ who gets the best result as the DADBC model

B. Hybrid Demonstration Construction

Fig. 3 shows the process of hybrid demonstration construction. The main idea behind GAN is to set up a zero-sum game and acquire learning through the adversarial learning process [22]. A typical GAN contains two parts, a generator, and a discriminator, which are generally composed of deep neural networks (DNNs). The core idea of the generator is to generate samples that mimic the distribution of the collected samples in the train data set. And noise vector is usually used as the input of the generator. The purpose of a discriminator is to distinguish as accurately as possible the source of the input

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

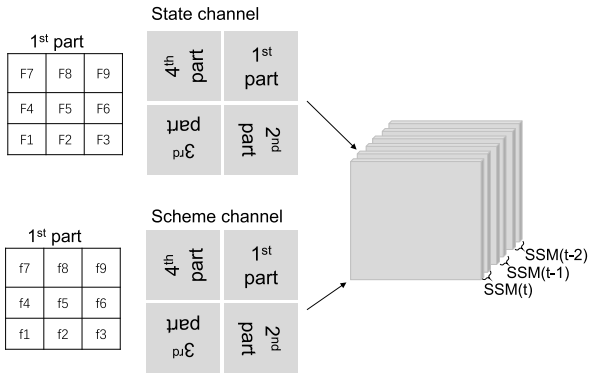

Fig. 2. State-Scheme Map and the data set. The second, third and forth part have the same shape as the first part.

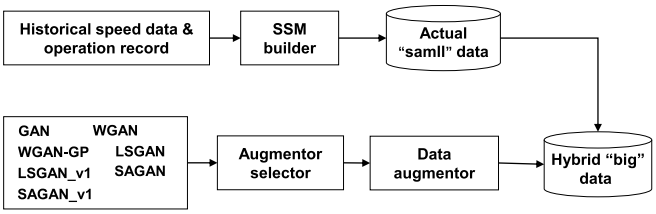

Fig. 3. The operational process of the data augmentation module.

data. After a GAN model is well-trained, virtual SSM samples are generated by feeding noise vector into its generator. The generated data is combined with the collected data to construct a hybrid demonstration data set.

GAN models update the parameters of the networks through a unified min-max objective function (Eq. 1).

$$\min_{G}\max_{D}\{f(D,G)=\mathrm{E}_{x\sim p_r(x)}[\log D(x)] + \mathrm{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]\} \quad (1)$$

In Eq. 1, z ($z \in \mathbb{R}^{100}$) represents noise data obtained from a standard normal distribution, and x represents the data collected from the actual traffic system. Here, x is composed of SSM samples at different times, and $x \in \mathbb{R}^{6\times6\times2}$. G represents the generator, which accepts the noise vector z, and generates virtual SSM samples. The generated virtual samples and the true data are both used as the input of the discriminator, D. If the discriminator cannot effectively distinguish the origin of the input data, it indicates a high similarity between the generated data and the real data, and the generator can be used to enhance the data and construct a hybrid demonstration.

Different GAN models leverage different loss functions and neural network architectures to improve generation ability. The original GAN mimics the distribution of the real data by minimizing the cross-entropy loss function of the discriminator and maximizing the same loss function of the generator. LSGAN chooses the least square error as the loss function to stable the training process of the model. Wasserstein distance (Eq. 2) can better measure the distance between two distributions in more complex situations, such as when two distributions do not overlap. This property, combined with the weight clipping technique, is used in WGAN.

$$W(P_r,P_g)=\inf_{\gamma\sim\Pi(P_r,P_g)}\mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|] \quad (2)$$

WGAN-GP attempts to address the issue of gradient vanishing or exploding caused by the weight clipping in a WGAN model by providing a loss function as

$$L=\underbrace{\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})]-\mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)]}_{\text{WGAN loss}}+\underbrace{\lambda\mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[\left(\left\|\nabla_{\hat{x}}D(\hat{x})\right\|_2-1\right)^2\right]}_{\text{gradient penalty}} \quad (3)$$

where $\tilde{x}$ represents the generated SSM sample, x represents the real SSM sample, and $\hat{x} \leftarrow \epsilon x + (1-\epsilon)\tilde{x}, \epsilon \sim U(0,1)$. Self-attention GAN (SAGAN) model contains the self-attention module, which can automatically establish a similarity measure between any two elements in the feature map, to the generator and discriminator to determine the importance of different features in the SSM samples. Thereby the efficiency of data utilization and model performance has been both improved.

Additionally, a variant of each of LSGAN and SAGAN is also considered. In particular, the self-attention module is added to the LSGAN to construct the LSGAN_v1 model. The fully-connected (FC) layers in the discriminator in the SAGAN model are replaced with a CNN block and called SAGAN_v1. Table I provides a summary of the tested model architecture in the GAN pool. The linear block is among them and includes: 1 × (FC layer, LeakyRelu) + 4 × (FC layer, BN_1D, LeakyRelu) + 1 × (FC layer, Tanh). The CNN block includes: 1 × (Conv [in, out, 3, 2, 1], LeakyRelu, Dropout2D) + 3 × (Conv [in, out, 3, 2, 1], LeakyRelu, Dropout2D, BN_2D) + Sigmoid. FC layers include: 2 × (FC layer, LeakyRelu) + FC layer. BN presents the batch-normalization technique and attention refers to the self-attention module introduced above.

TABLE I
THE STRUCTURE OF ALL GAN MODELS

Model	Generator	Discriminator
GAN [22]	Linear block	FC layers
WGAN [27]	Linear block	FC layers
WGAN-GP [28]	Linear block	FC layers
LSGAN [26]	Linear block	FC layers
SAGAN [29]	Linear block	FC layers
LSGAN_v1	Linear block+attention	Attention+FC layers
SAGAN_v1	Linear block+attention	Attention+CNN block

C. Augmentation Model Selection

Although various GAN models have different characters, it is not clear which GAN model is the most appropriate one for the traffic data augmentation task. In this study, GAN, LSGAN, WGAN, WGAN-GP, SAGAN (and its variant) are selected to constitute a GAN pool. The GAN pool aims to generate a large amount of virtual data for constructing hybrid demonstrations and complete the data augmentation task. The goal of the DADBC model is to approximate the scheme of human experts as closely as possible by learning from the hybrid demonstration. Thus, the DADBC model can be

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

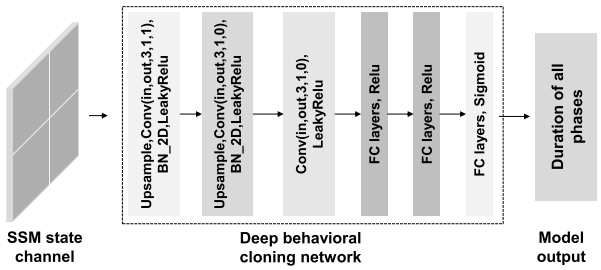

Fig. 4. The neural network architecture of deep behavioral cloning model.

considered as a regression problem and $L2$ distance in Eq. 4 is chosen to justify the difference between the model output and human behaviors.

$$J(\theta) = \|\pi_\theta(s_t) - a_t\|^2 \tag{4}$$

where s_t is the state and a_t is the scheme generated by humans, π_θ represents the policy that generates imitated schemes. The GAN model who gets the smallest $J(\theta)$ is selected as the final data augmentor.

D. Demonstration Imitation

After the hybrid demonstration is obtained through the real traffic control operation system and data augmentation module, the DBC model needs to distill the policy from the hybrid demonstration. Fig. 4 describes the structure of the DBC model that is used in the DADBC model. The state channel of the SSM sample (which represents the traffic state) is used as the input of the neural network. An eight-dimensional vector, extracted from the scheme channel, is used as the ground truth. According to the SSM scheme channel configuration rules, there are eight different phases in the scheme channel, namely the north-south straight phase, the north-south left-turn phase, the east-west straight phase, the east-west left-turn phase, the south straight left-turn phase, the north straight left-turn phase, the west straight left-turn phase, and the east straight left-turn phase. The eight specific durations of each phase are extracted from the scheme channel and form the eight-dimensional vector $[P1, P2, P3, P4, P5, P6, P7, P8]$.

The state-action mapping relationship determines the behavior of the system in different traffic states and it can be distilled from the demonstration. The expert policy can thus be described as a mapping from state to action, $\pi_\theta : s \mapsto a$. By establishing this mapping, the goal of imitating the expert policy can be achieved. Deep neural networks have powerful feature-extraction and function-fitting capabilities. The DBC model utilizes these characteristics and establishes a deep neural network with parameters θ, to represent the policy π_θ. The mapping is learned gradually by continuously adjusting the parameter of the neural networks, θ.

The DBC method is essentially a regression task, which achieves the purpose of learning policy by minimizing the distance, $Dist(\pi_\theta(s), a)$, between the output of the action by the model $\pi_\theta(s)$ and the actions a in the demonstration. In this paper, we let $Dist(\pi_\theta(s), a) = L2(\pi_\theta(s), a)$. Since the behavioral cloning method is a supervised learning method, it is able to learn the policy contained in the demonstration quickly. The convergent model is called π^*_θ, which is able to generate a unique scheme for each traffic state. The generated scheme closely resembles the scheme provided by the professional traffic signal engineers. In such a circumstance, we believe that the DADBC model has learned the hidden knowledge and experience behind the demonstration data set. For the testing traffic state, the performance of the proposed model is measured by comparing the difference between the solutions produced by the model and those provided by professional traffic engineers.

TABLE II
HYPERPARAMETERS SETTING

Hyperparameters	value
Learning rate of GANs	0.001
Optimizer of GANs	Adam (WGAN: RMSprop)
Batchsize of GANs	256
Noise of GANs	$z \in \mathbb{R}^{100}, z \sim \mathcal{N}(0,1)$
WGAN-GP lambda-gp	10
WGAN&WGAN-GP clip-value	0.01
Learning rate of DBC	0.1
Optimizer of DBC	SGD
Batchsize of DBC	64
Random seed	100

IV. CASE STUDY

Three different measurement indicators, including mean squared error (MSE), the standard deviation of MSE (MSE_STD), and mean absolute percentage error (MAPE), are used to measure the difference between the scheme output by the model and the scheme provided by the engineers. The definition of MSE error and the MAPE error are detailed in Eq. 5 and Eq. 6. In both equations, a_t and $\tilde{a}_t$ represent the t^{th} ground truth and the t^{th} output of the model, respectively. All models are implemented on a workstation with the following specifications: an 8-core Intel® Core™ i7-6700K CPU @ 4.00GHz and 2 * Titan V GPUs.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(a_t - \tilde{a}_t)^2 \tag{5}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{a_t - \tilde{a}_t}{a_t}\right| \tag{6}$$

The hyperparameter values used in the experiments are detailed in Table II, which are determined by manual trial-and-errors.

A. Data Descriptions

An actual traffic dataset collected from Hangzhou, China (Fig. 5), is used for the verification of the proposed method. Two main categories of data sources, including state data and signal scheme data, were used in this study. 16 signalized intersections are picked to collect the state data from floating cars and the time interval is from 2020-03-01 to 2020-03-31. Simultaneously, a SCATS system is employed to take command of the traffic system and it can provide interfaces for manual intervention and historical records of manual operations. Overall, more than 720,000 signal cycles were

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

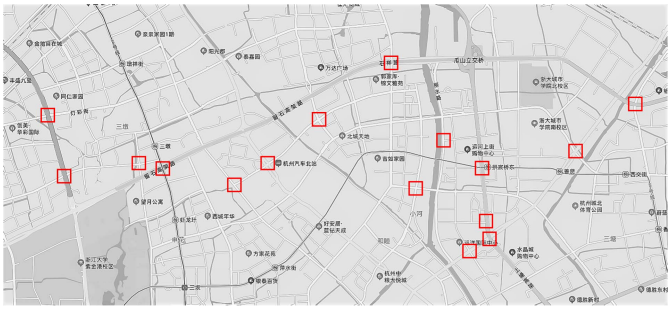

Fig. 5. All raw traffic state data and control operational records are from the actual intersection marked by the red rectangle in the figure.

performed concerning the tested intersections, and merely 5% among them were adjusted by humans. The duration of different phases ranging from 7 to 110 seconds and the maximum and minimum cycle lengths are 34 and 300 seconds. Min-max data normalization was used to rescale the original traffic state and signal scheme such that all feature values were within the range of 0 to 1. Additionally, these two types of data are corresponding to the two channels of SSM, respectively. In each part of the channels, three different kinds of features were used, including the collected data, the moving average data, and the deviation value between the collected data and the moving average data.

Table III introduces the meaning of the features that the state channel and scheme channel of SSM contain. The state channel is constructed using two types of link speed data of an intersection. They are the average of all vehicle speeds associated with a link and the average speed for the vehicles that do not stop. To take the historical information and change trend into consideration, moving average values of speeds and the deviation value between the current link speed and the moving average link speed are also integrated. Regarding the signal scheme channel, each link of an intersection is assumed to have three types of possible signalized directions: straight, left, and straight-left. The phase duration, the cycle length, and the moving average of the two variables above are employed in the scheme channel. Moreover, the minutes of a day are included in the scheme channel to infer to a rough operation pattern during a day.

B. Offline Evaluations

Seven GAN models (GAN, WGAN, WGAN-GP, LSGAN, LSGAN_v1, SAGAN, and SAGAN_v1) have been tested to generate virtual SSM samples, which are merged into the real data. All models are implemented, and the GAN models are trained under the same SSM data set. Different sizes of virtual SSM samples are generated by the generators in the GAN pool to uncover the relationship between model performance and data augmentation. Each generator is used to generate 5,000, 10,000, 20,000, and 30,000 virtual samples, respectively. The corresponding state and action are then extracted out to form a <state-action> pair and these virtual samples and real data constitute several hybrid demonstrations. Using these demonstrations, different DBC models (they have the same network architecture) are trained under the early stopping strategy.

TABLE III
DESCRIPTION OF FEATURES IN SSM SAMPLE

Channel definition	Symbol	Description
State channel	F1	Average speed: v
	F2	Average speed with no stop: v'
	F3	F1 − F2
	F4	MA30(F1)
	F5	MA30(F2)
	F6	F4 − F5
	F7	F1 − F4
	F8	F2 − F5
	F9	F7 − F8
Scheme channel	f1	Duration of straight phase
	f2	Duration of left phase
	f3	Duration of straight-left phase
	f4	Cycle length
	f5	MA10(f1)
	f6	MA10(f2)
	f7	MA10(f3)
	f8	MA10(f4)
	f9	Minute of the day

F1 − F2 means F1 minus F2 and MA10(*) means 10 periods moving average value of *.

Table IV illustrates the results of different models with various sizes of virtual samples, on the MSE and MSE_STD of phase duration and the MAPE of cycle length. Compared with the original DBC model, almost all the tested DADBC models obtain improvement according to the three indicators. Model_3 (the DADBC model trained with the demonstration augmented by the generator of LSGAN_v1 model) generates the best performance by achieving the minimum MSE and MAPE index values. The MSE and MAPE indices of the best model can be particularly decreased by around 30% and 20%, respectively, compared with the original DBC model. Since the enhanced hybrid demonstration obtained by the DADBC model can cover more traffic situations, the MSE_STD generated by the DADBC model is also much smaller than the original DBC model.

The experimental results also demonstrate that two of the models, Model_5 and Model_6, do not perform as well as the original model. This difference proves that models with different characteristics have different effects on the performance of the proposed method. The neural network structure of GANs, the loss function, and the self-attention module all have a direct impact on the data augmentation result. For example, the only difference between Model_2 and Model_3 is that the latter uses the self-attention module, which improves performance significantly. Different GAN structures are needed for different data types and data forms to get better augmentation results. The data augmentation method can influence the performance of subsequent DBC models. In our traffic control operations problem, the LSGAN and self-attention module outperforms other models.

Fig. 6 displays an example of the statistical characteristics of the SSM sample generated by the LSGAN model. Specifically, Fig. 6 (a) and (b) demonstrate the speed and phase duration distribution of the north entrance of all intersections, respectively. Both the speed and scheme in the virtual data generated by the model are highly consistent with the distribution of true data. Likewise, Fig. 6 (c) and (d) clearly show that the speed and phase duration change generated by the model and real data are consistent in terms of trend and value.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
THE RESULTS OF DIFFERENT MODELS WITH VARIOUS SIZES OF VIRTUAL SAMPLES

Model	DBC(-)	Model_1 (DBC-GAN)				Model_2 (DBC-LSGAN)				Model_3 (DBC-LSGAN_v1)			
Size of virtual samples	0	5K	10K	20K	30K	5K	10K	20K	30K	5K	10K	20K	30K
MSE	33.42	38.14	32.9	28.99	31.48	26.41	25.5	28.61	24.08	29.27	22.81	22.44	23.68
MSE_STD	9.63	7.92	7.49	7.27	7.53	6.87	6.33	6.11	6.17	6.46	6.53	6.14	6.79
MAPE	0.150	0.166	0.145	0.142	0.149	0.139	0.140	0.145	0.130	0.123	0.118	0.127	0.129

Model	Model_4 (DBC-SAGAN)				Model_5 (DBC-SAGAN_v1)				Model_6 (DBC-WGAN)				Model_7 (DBC-WGAN_GP)			
Size of virtual samples	5K	10K	20K	30K	5K	10K	20K	30K	5K	10K	20K	30K	5K	10K	20K	30K
MSE	25.85	25.44	26.56	25.21	49.92	40.9	44.66	56.89	37.05	53.83	48.71	45.95	29.37	37.39	38.8	26.15
MSE_STD	6.66	6.01	6.11	5.9	11.08	9.22	9.93	11.5	9.76	12.8	11.18	11.69	7.11	7.72	8.41	6.57
MAPE	0.132	0.136	0.136	0.136	0.208	0.207	0.215	0.212	0.170	0.183	0.229	0.203	0.162	0.197	0.183	0.140

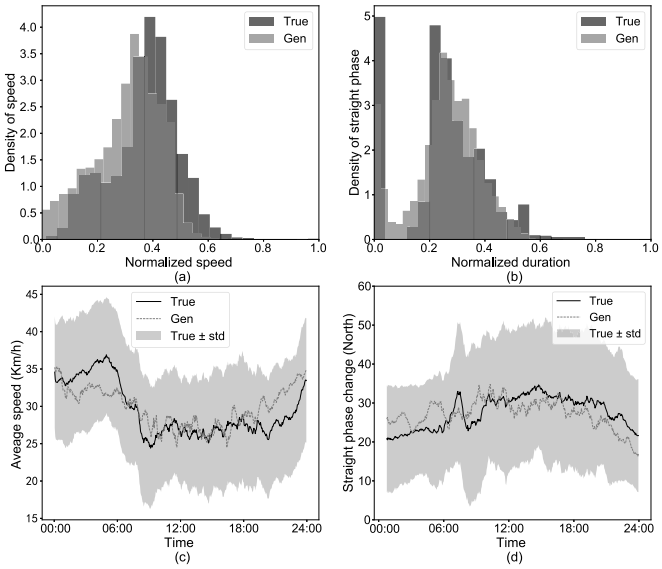

Fig. 6. Statistical characteristics of generated SSM samples and true data. (a) speed distribution. (b) phase duration distribution. (c) average speed. (d) duration of straight phase in a day.

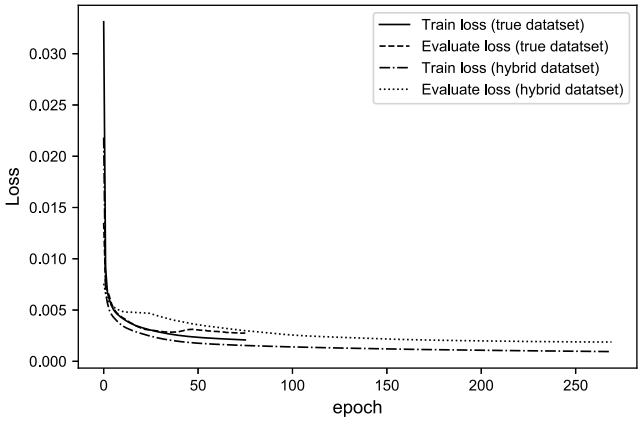

Fig. 7. Training process of deep behavioral cloning model.

Fig. 7 describes the training process for two arbitrarily models. True and hybrid datasets are separately utilized for training a DBC model. As the training process goes on, the values of all loss functions decrease, tending towards 0, which means that the models have learned the information in the demonstration and that the outputs are getting closer and closer to the true

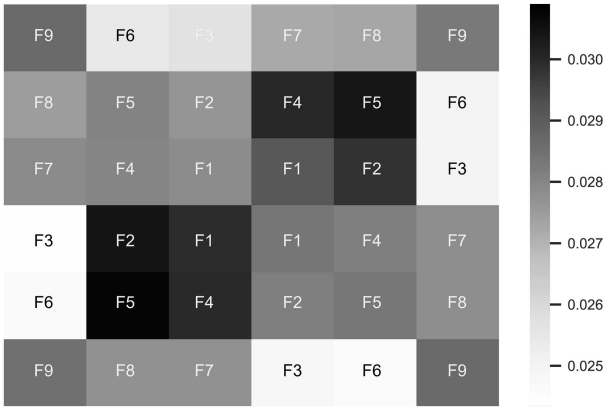

Fig. 8. Attention map of the state channel in the generator of LSGAN_v1.

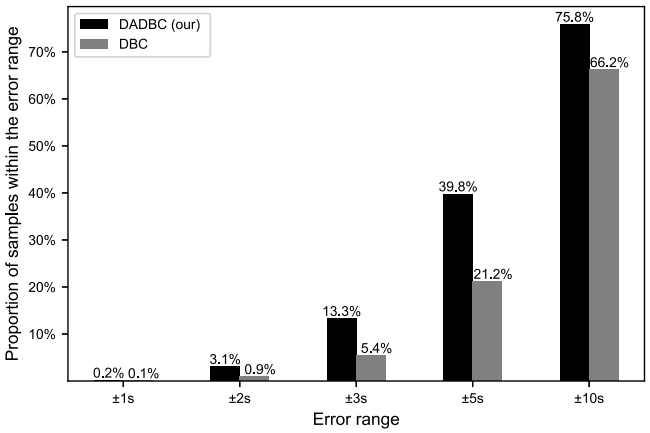

Fig. 9. Proportion of samples within different error ranges.

scheme provided by the professional traffic engineers. In the training process of deep neural networks, the early stopping strategy [39] can be used to avoid overfitting. In this paper, the DADBC model stops training when the evaluation loss increases by three consecutive epochs.

Under the actual small demonstration, the model quickly triggers the early stopping condition and ends the training, while the model can be more fully trained under the hybrid demonstration, and the evaluation loss converges to a smaller value. This proves that deep neural networks are prone to overfitting earlier when a smaller data set is used to train the model. The data augmentation can effectively alleviate

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

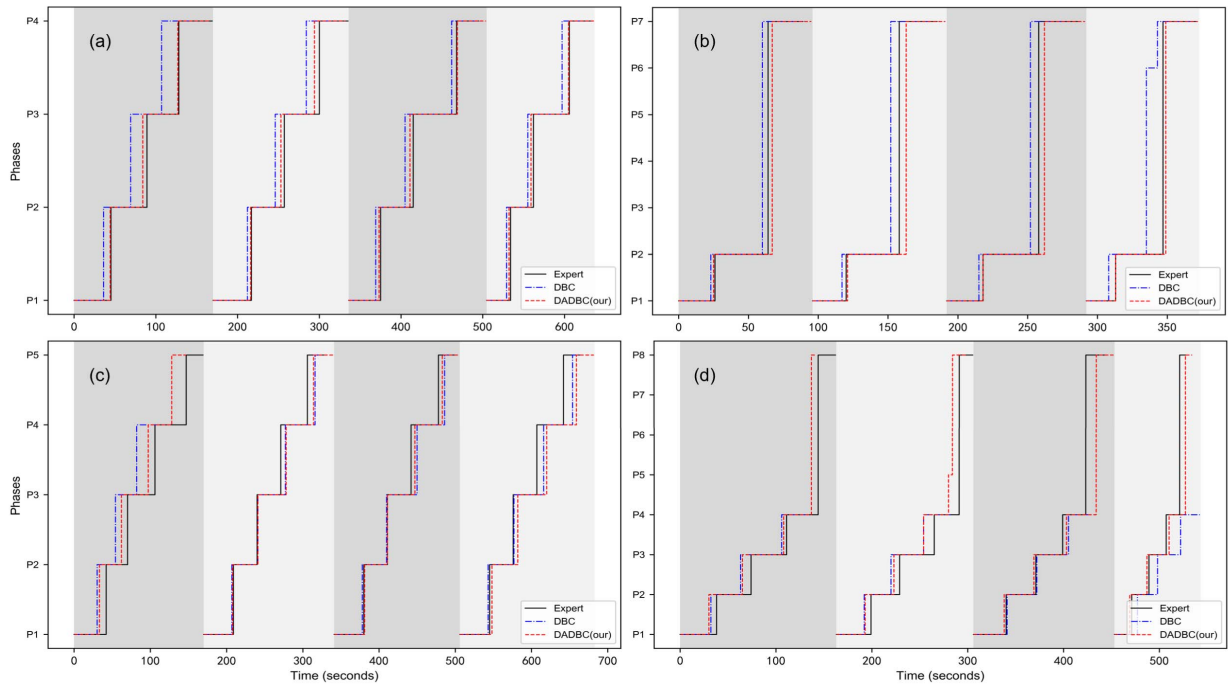

Fig. 10. Scheme trajectories for different phase groups. $P1$-$P4$ represent straight phases and left-turn phases, $P5$-$P8$ represent straight and left-turn combined phases. (a) phase groups: $P1$, $P2$, $P3$, $P4$, (b) phase groups: $P1$, $P2$, $P7$, (c) phase groups: $P1$, $P2$, $P3$, $P4$, $P5$, (d) phase groups: $P1$, $P2$, $P3$, $P4$, $P8$.

the overfitting of the model. At the same time, a lower validation loss indicates that the DADBC model can obtain a better policy compared with the conventional DBC model. The data augmentation module increases the heterogeneity of the samples in the demonstration, which means that the neural network may not fall into the local extremum and may converge to the global optimum more easily.

The LSGAN with the self-attention module can learn the importance weight of all features in the state channel, and Fig. 8 is the attention map of the self-attention module in the generator. In Fig. 8, the string in the box represents the feature ID and the color represents the weight of this feature. The darker the color, the greater the weight value. The attention map demonstrates that different features take different weights in each entrance. However, different entrances all present a similar pattern of weight distribution. Within each entrance, the speed and phase of the present moment, as well as their respective moving averages, corresponding to $F1$, $F2$, $F4$, $F5$, carry more weight. The difference between the two speeds and the difference between the corresponding moving average values carries significantly less weight. In human knowledge and experience, the most important information/observations used to determine the current scheme are the values of speed and the duration of the phase, in the past and at present. The fact that these characteristics have a greater weight keeps consistent with the knowledge and experience of humans.

C. Online Evaluations

In order to evaluate the performance of the DADBC model further, new schemes for the testing state are generated. If the errors of all phases in a scheme are less than a certain error range, the scheme is considered to be within the current error range. With this in mind, we compute the proportion of samples in different error ranges of the output schemes of the DADBC model and the conventional DBC model, as shown in Fig. 9. Five different error ranges are set and within the same error range, the number of schemes from the DADBC model is always more than the DBC model. And within a smaller error range, the advantages of the DADBC model are more obvious. This fact confirms that, in most cases, the proposed DADBC method can provide schemes that are closer to traffic engineers for a given state. Compared with the conventional DBC model, the DADBC model is more effective for the imitation of professional traffic engineers.

To adapt to different traffic demands, different intersections need to adopt different phase groups to improve traffic efficiency. The schemes trajectories of several different phases groups are displayed in Fig. 10. Compared to the traditional DBC approach, the trajectory of the DADBC approach is closer to the trajectory of experts across different phase groups. It approved that our DADBC model imitates the expert better. MAPE errors of the DADBC model and the DBC model of all phases are also counted separately. The results show that the MAPE error of the DADABC method is 0.42% to 4.65% (average 1.66%) smaller than that of the DBC method overall eight phases. In general, for different unknown states in the test set, the scheme output by the DADBC model is closer to the scheme provided by human experts than the conventional DBC model.

Table V illustrates the average speed of the road network at different times of the day. Compared to the speed at which the traffic engineers adjust the traffic control parameters, the average speed of the road network increases after the adjustment. Regardless of the time (in the morning or evening rush hour, or during off-peak hours), the adjustment always improves

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
AVERAGE SPEED (km/h) AFTER THE ADJUSTMENT

Case	7:00-9:30	13:00-14:00	17:00-19:00	Whole day
Adjustment	24.138	23.959	23.950	27.110
10 minutes after adjustment	24.643	25.414	24.777	27.632
30 minutes after adjustment	24.534	25.931	25.073	27.715
60 minutes after adjustment	24.538	26.019	25.125	27.661

the average speed. This phenomenon indicates that traffic engineers certainly do improve traffic efficiency. Our DADBC model imitates traffic engineers effectively and demonstrates promising applications.

V. CONCLUSION AND DISCUSSION

In this study, a data augmented deep behavioral cloning approach is proposed for urban traffic control under the parallel learning framework. Through the utilization of advanced GAN models and the DBC method, historical operational records are effectively enhanced and mined, with the aim of learning and imitating the control operation patterns of professional human experts. Specifically, the method proposed in this paper addresses the issue of insufficient human operation record data by using the LSGAN model with a self-attention module. A hybrid demonstration data set, including the recorded and generated data, is used to train a DBC model to imitate the control operation patterns of human traffic engineers. In the case study, the proposed method, using the real traffic control operation data of Hangzhou in China, has been proved that the scheme of our method is highly consistent with human experts. Such a consequence is able to effectively reduce the workload of human experts and shorten their response time of the traffic system when a congestion event occurs.

In practice, professional traffic engineers need a long time to accumulate experience and knowledge, and there may be significant differences between different traffic engineers. Physiological limitations of traffic engineers, such as fatigue, can also impact the control performance. The demonstration provided by professional traffic engineers is imperfect. In the parallel learning theory, the prescriptive learning process aims to improve the policy by emerging technologies, such as reinforcement learning. Reinforcement learning has the ability to learn autonomously and improve decision-making ability through interaction with the environment. In order to overcome the disadvantages of the human expert demonstration and the behavioral cloning method, we will try to combine the deep reinforcement learning methods with the DADBC method in a follow-up study.

REFERENCES

[1] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers' routing," *Transp. Res. B, Methodol.*, vol. 38, no. 4, pp. 329–342, 2004.

[2] J. Garcia-Nieto, A. C. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 823–839, Dec. 2013.

[3] J. Jin and X. Ma, "A multi-objective agent-based control approach with application in intelligent traffic signal system," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3900–3912, Oct. 2019.

[4] J. Jin, H. Guo, J. Xu, X. Wang, and F.-Y. Wang, "An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 20, 2020, doi: 10.1109/TITS.2020.2973736.

[5] A. G. Sims and K. W. Dobinson, "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," *IEEE Trans. Veh. Technol.*, vol. 29, no. 2, pp. 130–137, May 1980.

[6] J. Jin, X. Ma, and I. Kosonen, "An intelligent control system for traffic lights with simulation-based evaluation," *Control Eng. Pract.*, vol. 58, pp. 24–33, Jan. 2017.

[7] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[8] W. Xu *et al.*, "Internet of vehicles in big data era," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan. 2018.

[9] D. Nallaperuma *et al.*, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4679–4690, Dec. 2019.

[10] X. Zhou, W. Liang, K. I.-K. Wang, and S. Shimizu, "Multi-modality behavioral influence analysis for personalized recommendations in health social media environment," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 5, pp. 888–897, Oct. 2019.

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

[12] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, 2017.

[13] T. Zhang *et al.*, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.

[14] X. Li *et al.*, "Deep imitation learning for traffic signal control and operations based on graph convolutional neural networks," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[15] D. Silver *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, p. 484, 2016.

[16] R. P. Bhattacharyya, D. J. Phillips, B. Wulfe, J. Morton, A. Kuefler, and M. J. Kochenderfer, "Multi-agent imitation learning for driving simulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1534–1539.

[17] M. Alibeigi, M. N. Ahmadabadi, and B. N. Araabi, "A fast, robust, and incremental model for learning high-level concepts from human motions by imitation," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 153–168, Feb. 2017.

[18] B. Wang, E. Adeli, H.-K. Chiu, D.-A. Huang, and J. C. Niebles, "Imitation learning for human pose prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7124–7133.

[19] L. Li, Y. Lin, N. Zheng, and F.-Y. Wang, "Parallel learning: A perspective and a framework," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 3, pp. 389–395, Jul. 2017.

[20] F.-Y. Wang, X. Wang, L. Li, and L. Li, "Steps toward parallel intelligence," *IEEE/CAA J. Automatica Sinica*, vol. 3, no. 4, pp. 345–348, Oct. 2016.

[21] S. K. Saksena, N. B. S. Hegde, P. Raja, and R. M. Vishwanath, "Towards behavioural cloning for autonomous driving," in *Proc. 3rd IEEE Int. Conf. Robotic Comput. (IRC)*, Feb. 2019, pp. 560–567.

[22] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[23] I. Markelic *et al.*, "The driving school system: Learning basic driving skills from a teacher in a real car," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1135–1146, Dec. 2011.

[24] Y. Xiong, G. Zheng, K. Xu, and Z. Li, "Learning traffic signal control from demonstrations," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2289–2292.

[25] Y. Huo, Q. Tao, and J. Hu, "Tensor-based cooperative control for large scale multi-intersection traffic signal using deep reinforcement learning and imitation learning," 2019, *arXiv:1909.13428*. [Online]. Available: http://arxiv.org/abs/1909.13428

[26] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[27] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875
[28] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
[29] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018, *arXiv:1805.08318*. [Online]. Available: http://arxiv.org/abs/1805.08318
[30] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "AdaGAN: Boosting generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5424–5433.
[31] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[32] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 588–598, Sep. 2017.
[33] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.
[34] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," 2018, *arXiv:1803.09655*. [Online]. Available: http://arxiv.org/abs/1803.09655
[35] C.-T. Lin, S.-W. Huang, Y.-Y. Wu, and S.-H. Lai, "GAN-based Day-to-Night image style transfer for nighttime vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 6, 2020, doi: 10.1109/TITS.2019.2961679.
[36] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, Dec. 2018.
[37] C. Bowles *et al.*, "GAN augmentation: Augmenting training data using generative adversarial networks," 2018, *arXiv:1810.10863*. [Online]. Available: http://arxiv.org/abs/1810.10863
[38] S. Roy Chowdhury *et al.*, "Automated augmentation with reinforcement learning and GANs for robust identification of traffic signs using front camera images," 2019, *arXiv:1911.06486*. [Online]. Available: http://arxiv.org/abs/1911.06486
[39] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 402–408.

Xiaoshuang Li received the B.S. degree in automation from Wuhan University, Wuhan, China, in 2017. He is currently pursuing the Ph.D. degree in control theory and control engineering with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include intelligent transportation systems, deep learning, and deep reinforcement learning.

Peijun Ye received the Ph.D. degree from the University of Chinese Academy of Sciences in 2013.

He was a Visiting Scholar with the Department of Cognitive Science, University of California at San Diego, from 2017 to 2018. He is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, and an Advanced Researcher in Parallel Intelligence Research Center, Qingdao Academy of Intelligent Industries, China. His research interests mainly focus on multiagent systems, cognitive computing, artificial intelligence, and intelligent transportation systems. He is the author/coauthor of more than 40 papers, eight patents, and one monograph. He serves as an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and the Secretary for the Social and Economic Computing ACM Chapter. He is also an invited reviewer of the *Journal of Artificial Societies and Social Simulation*, *Knowledge and Information Systems*, and the IEEE International Conference on Intelligent Transportation Systems (2011–2019).

Junchen Jin (Member, IEEE) received the B.Eng. degree in traffic engineering from Beijing Jiaotong University, Beijing, China, and the M.Sc. and Ph.D. degrees in transport science from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2014 and 2018, respectively. He was the Vice Director of the Smart Transportation Research Institute, Enjoyor Co., Ltd., Hangzhou, China. He is also a Post-Doctoral Researcher with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include intelligent transport systems, traffic simulation and control, recommender systems, artificial intelligence, deep learning, and reinforcement learning.

Fenghua Zhu (Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. He is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, China. His research interests are artificial transportation systems and parallel transportation management systems.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined The University of Arizona in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council. In 2002, he was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems. His current research focuses on methods and applications for parallel intelligence, social computing, and knowledge automation. He is a fellow of the INCOSE, IFAC, ASME, and AAAS. In 2007, he received the National Prize in Natural Sciences of China and became an Outstanding Scientist of ACM for his work in intelligent control and social computing. He received the IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, respectively. In 2014, he received the IEEE SMC Society Norbert Wiener Award. Since 1997, he has been serving as the General or Program Chair of over 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, the American Zhu Kezhen Education Foundation from 2007 to 2008, the Vice President of the ACM China Council from 2010 to 2011, and the Vice President and the Secretary General of the Chinese Association of Automation from 2008 to 2018. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, the *IEEE ITS Magazine* from 2006 to 2007, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA from 2014 to 2017, and the *China's Journal of Command and Control* from 2015 to 2020. He was the EiC of the IEEE INTELLIGENT SYSTEMS from 2009 to 2012 and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2009 to 2016. He has been the EiC of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS since 2017, and the Founding EiC of *China's Journal of Intelligent Science and Technology* since 2019. He is currently the President of the CAA's Supervision Council, IEEE Council on RFID, and the Vice President of the IEEE Systems, Man, and Cybernetics Society.

Authorized licensed use limited to: INSTITUTE OF AUTOMATION CAS. Downloaded on May 11,2022 at 13:14:24 UTC from IEEE Xplore. Restrictions apply.