2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)
October 8-12, 2022, Macau, China

An Online Traffic Simulation Modeling Framework for Real-Time Road Traffic Control and Management

Yuanqi Qin, Junchen Jin, *Member, IEEE*, Wen Hua, Xingyuan Dai, and Xiao Wang, *Senior Member, IEEE*

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922538

Abstract— **Online traffic simulation feeds from online information to simulate vehicle movement in real-time, which has recently seen substantial advancement in road traffic control and management. It has been a challenging problem due to three aspects: 1) the diversity of traffic patterns caused by heterogeneous layouts of urban intersections; 2) the complexity of spatiotemporal correlations; 3) the requirement of adjusting traffic model parameters in a real-time system. To cater to these challenges, this paper proposes an online traffic simulation modeling framework via a meta-learner. In particular, simulation models with various intersection layouts are automatically generated using an open-source simulation tool, SUMO, according to static traffic geometry attributes. Through a meta-learning technique, the proposed modeling framework enables an automated learning process for estimating model settings capable of adapting traffic model parameters according to dynamic traffic information in real-time. Such a process is featured with various traffic scenarios and different spatiotemporal correlations. Through computational experiments, we demonstrate that the meta-learning-based framework is able to self-adapt its effectiveness according to real-time traffic data.**

I. INTRODUCTION

Traffic simulation models are designed to characterize road infrastructure and transportation systems, which plays a crucial role in modeling, planning, management, and control of urban traffic systems [1]–[7]. In the past decades, traffic simulation is primarily performed offline mainly because building a traffic simulation model with high fidelity requires extensive work of model calibrations (e.g., estimating traffic flows, turning ratios, and origin-destination matrices) [8]–[11]. A set of model calibration parameters is merely suitable for a specific simulation scenario. However, an urban road network has complex and heterogeneous configurations for intersections. Meanwhile, traffic pattern changes dynamically because of the high stochasticity of traffic systems. This poses a big challenge for the practice of using traffic simulations in traffic management and control.

This work was supported by the National Natural Science Foundation of China (62173329). (Corresponding author: Xiao Wang.)

Y. Qin is with Zhejiang Lab, Hangzhou 310030, China.

J. Jin is with Zhejiang Supcon Information Technology Co., Ltd., Hangzhou 310030, China.

W. Hua is with ECARX Technology Co., Ltd., Shanghai, China.

X. Dai is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100049, China.

X. Wang is with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

An online traffic simulation framework dynamically creates suitable traffic simulation models according to real-time traffic information. Such a framework becomes increasingly important for implementing traffic control measures and analyzing their influence on various control measures. Recent studies began to draw attention to consecutive analyses of online traffic measures for real-time traffic management. Wang et al. solved an OD demand prediction problem via a graph convolution method to capture the dynamic mobility pattern within a traffic network [12]. In addition, Jin et al. proposed a non-parametric Bayesian framework for real-time estimate traffic states [13] and tested a real-time simulation for urban traffic control [14]. These models managed to extract features from multi-source data to gain knowledge of traffic patterns and traffic states. Nevertheless, previous traffic simulation studies usually rely on a large set of training samples and require a pre-defined simulation scenario [8], [15]–[18].

Besides, most online models rely on deployment of sophisticated traffic detectors and corresponding data collection in a large scale. The procedure is not a cost-effective solution for practical implementation [19]–[22]. Thus, online traffic simulation needs to encounter a continual stream of calibration tasks for heterogeneous urban networks with time-variant traffic patterns. This requires a meta-learning model with fast-learning adaptability from limited data [23], [24]. The meta-learning model is generalized by learning from a large range of various model calibration tasks to extract domain-general information which can act as an inductive bias to improve learning efficiency in new tasks. The model learns faster in new tasks with a small amount of data [25].

Consequently, according to the best of the authors' knowledge, few of the existing studies can support real-time traffic model estimations for heterogeneous road networks in simulation systems. To tackle the aforementioned challenges, we propose an online simulation framework for real-time road traffic control and management, which is developed based on an open-source microscopic traffic simulation software tool, called Simulation of Urban MObility (SUMO) [26]. The framework automatically generates static traffic simulation models with dynamic model settings according to real-time traffic observations. The core modeling component of the framework is a meta-learner [27], [28]. The meta-learner provides a paradigm to learn new tasks much faster than from scratch with an optimal learning model. The contributions of the framework are summarized below.

- Traffic simulation models can be quickly created con-

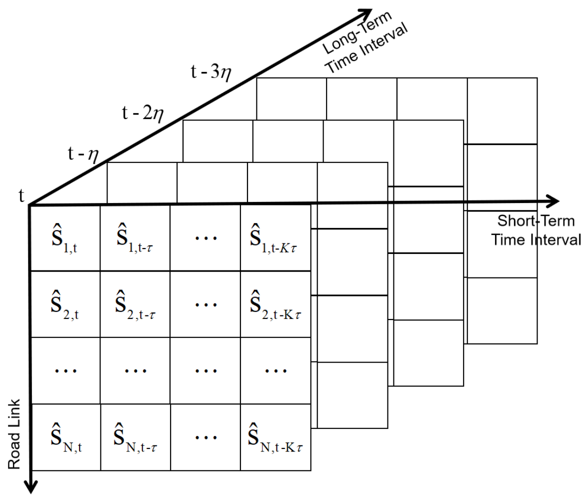


Fig. 1. Description of the tensor representation of the observed traffic states.

sidering heterogeneous layouts of urban intersections;

- Traffic spatiotemporal characteristics are mapped with a wide range of simulation scenarios (modeling tasks);
- The learned knowledge is fast transferred from source models across different simulation scenarios to improve the model estimation performance in a target simulation model with limited data samples;
- Dynamic simulation model settings are adapted in real time according to the observed data.

II. Problem Definition

The proposed online simulation modeling framework solves the fundamental turning movement estimation problem. The framework can estimate turning movements between any two approaches within an intersection given the associated observed traffic states. In terms of simulating an urban network, an online traffic simulation model is created by simultaneously performing estimation models associated with all intersections. The estimation problem is formulated as a classification problem using observed traffic states.

$$C_{X,t_1,t_2} = f(\mathbf{S}^{obs}_{t_1,t_2}), \tag{1}$$

where C_{X,t_1,t_2} refers to a class with respect to turning movement table X within time interval $[t_1, t_2]$. s^{obs}_{i,t_1,t_2} denotes the observed traffic state of link i within time interval $[t_1, t_2]$. As described in Equation 1, the model inputs and outputs are related to observed traffic states and turning movement table, respectively.

In the study, the model input is defined as a three-dimensional tensor to capture both short-term (e.g., one-minute) and long-term (e.g., thirty-minute) temporal characteristics. Figure 1 depicts the tensor representation of $\mathbf{S}^{obs}_{t-3\eta,t}$ that is established by stacking three two-dimensional matrices in long-term time interval order. A two-dimensional observed traffic state matrix at t is formulated as

$$\mathbf{s}_{t-K\tau,t} = \begin{bmatrix} \hat{s}_{1,t} & \hat{s}_{1,t-\tau} & \cdots & \hat{s}_{1,t-K\tau} \\ \hat{s}_{2,t} & \hat{s}_{2,t-\tau} & \cdots & \hat{s}_{2,t-K\tau} \\ \cdots & \cdots & \hat{s}_{i,t-j\tau} & \cdots \\ \hat{s}_{N,t} & \hat{s}_{N,t-\tau} & \cdots & \hat{s}_{N,t-K\tau} \end{bmatrix}, \tag{2}$$

where K represents the number of short-term time intervals. N denotes the number of road links associated with an intersection. $\hat{s}_{i,t-j\tau}$ is the moving average of traffic states at time interval $j\tau$ of road link i. $i = 1, 2, \cdots, N$ and $j = 0, 1, \cdots, K$. Here, τ represents a short time interval such as one minute and η denotes a long-term time interval and $\eta \gg \tau$ such as one hour.

III. Methodology

A. Overall Framework

Figure 2 describes three core processes of the proposed framework . Note that the estimation model is intersection-based and we generate the online traffic simulation model by real-time estimate turning movement which is the core model parameters for an intersection. Firstly, the framework starts from an offline training process that establishes a base estimation model using a predefined meta-learner. In the training process, various scenarios are offline simulated using SUMO. Here, a traffic simulation scenario (or a modeling task) refers to a unique combination of intersection layout (e.g., a four-approach two-lane intersection, a three-approach three-lane intersection) and control setting (e.g., signal control type, signal control parameters). A large dataset for meta-training is generated by running multiple traffic simulations that continuously reports traffic states given an input turning movement table. Then, the model trains a base model for turning movement estimations via a meta-learner to capture the general characteristics across different estimation tasks.

Secondly, a particular road network is selected and each intersection in the network is required to build a specific turning movement estimation model. The framework quickly creates a static simulation model and generates a small-size meta-testing dataset by iteratively running the simulation model with predefined simulation scenarios. We design an online fine-tuning process to generate different models for intersections with different spatiotemporal features. During the fine-tuning process, the meta-learner online generates a specific estimation model with every intersection, which adjusts the parameters of the trained base estimation model. In principle, the fine-tuning process can be performed in parallel, and then be finished within few minutes.

As shown in Figure 2(c), each specific estimation model is deployed for providing estimated turning movement based on the observed traffic states in real-time. In return, with the estimated turning movement, different traffic control strategies and schemes are evaluated using the online traffic simulation. Thereafter, simulation-based optimization for proactive traffic operations can be carried out to consistently enhance traffic mobility efficiency and reduce environmental impacts [29], [30].

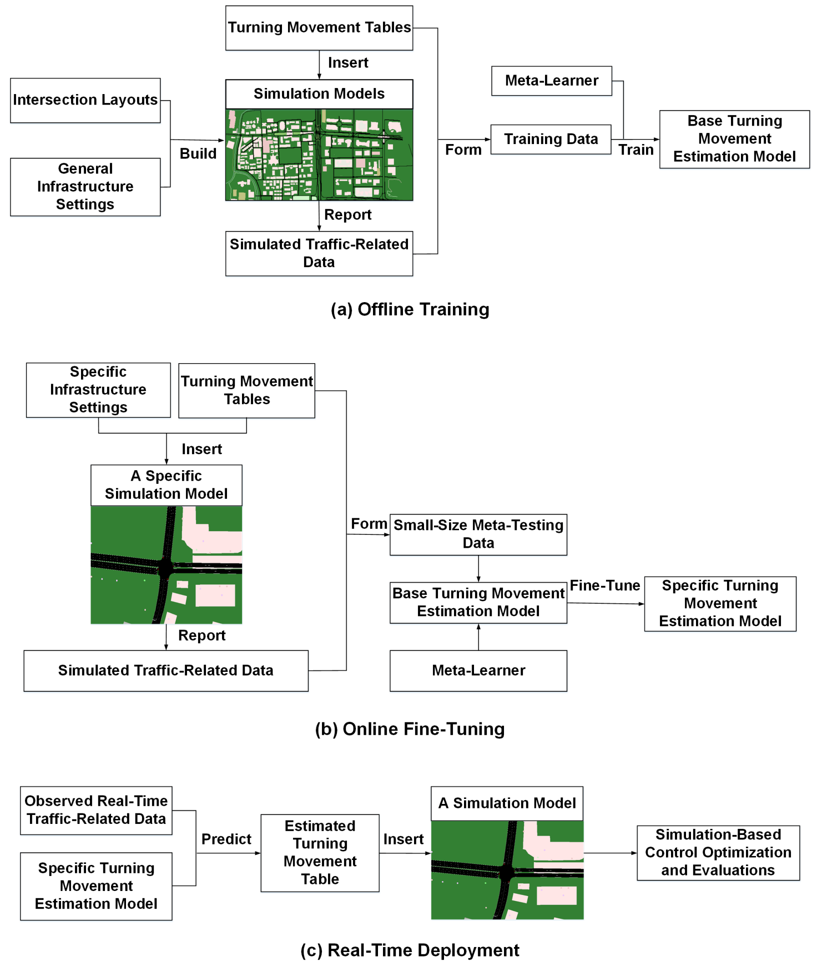


Fig. 2. Descriptions of the operational process including (a) offline training, (b) online fine-tuning, and (c) real-time deployment.

B. Meta-Learner

Meta-learner uses a general-purpose algorithm that tries to learn 'how to learn' by generalizing across various tasks within the same domain. Assume an estimation model is defined as

$$\hat{C} = f_\theta(\mathbf{S}) \tag{3}$$

where $\mathbf{S}$ and $\hat{C}$ denote model input and output, respectively. θ refers to model parameters. Let $\mathcal{D} = \{(\mathbf{S}_l, \hat{C}_l) : k = 1 \cdots, L\}$ denote a training dataset, where K denotes the number of samples. The model training process is operated by obtaining a minimum loss:

$$\theta^* = \arg\min_\theta \mathcal{L}_\mathcal{D}(f_\theta, \lambda) \tag{4}$$

where λ defines the hyperparameters of training an estimation model, such as function class for f. $\mathcal{L}$ refers to a loss function. Let a task $\mathcal{T}$ to be a dataset and loss function $\mathcal{T} = \{\mathcal{D}, \mathcal{L}\}$. Meta-learning performs the optimization above from distribution of tasks rather than from scratch, which is defined to minimize the expectation of loss evaluation performance over task distribution $p(\mathcal{T})$.

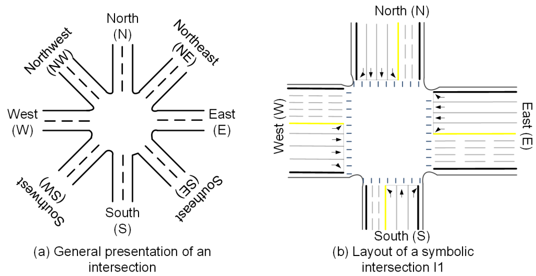
$$\min_\lambda \mathbb{E}_{\{\mathcal{D},\mathcal{L}\} \in p(\mathcal{T})} \mathcal{L}_\mathcal{D}(f_\theta, \lambda) \tag{5}$$

where $\mathcal{L}(\mathcal{D}, \lambda)$ measures the performance of a model trained using hyperparameter λ on dataset $\mathcal{D}$. Few-shot learning for online traffic simulation refers to estimating a turning movement class using a small amount of training data. Training data is designed as n-way and k-shot tasks, meaning that each task is a classification problem with n classes and k samples belong to a class. In meta-learning, two separate stages (meta-training and meta-testing) together with two different task sets (source and target sets) are defined. In the meta-training stage, each source task has both training and validation data, also called support and query datasets ($\mathcal{D}^{support}$ and $\mathcal{D}^{query}$), to learn across-task knowledge. Meta-training is formalized as follows to train a base model.

$$\lambda^* = \arg\min_\lambda \sum_{m=1}^{M} \mathcal{L}^{meta}_{\mathcal{D}^{query}_m}(f_{\theta^*_m}, \lambda) \tag{6}$$

$$\text{s.t. } \theta^*_i(\lambda) = \arg\min_\theta \mathcal{L}^{task}_{\mathcal{D}^{support}}(f_\theta, \lambda) \tag{7}$$

where M refers to the number of source tasks. $\mathcal{L}^{meta}$ and $\mathcal{L}^{task}$ represent the across-task and task-specific loss functions, respectively. The learned knowledge is then used in the meta-testing stage to train the base model concerning the previously unseen target tasks.



X=

	N	NE	E	SE	S	SW	W	NW
N	NA	NA	I	NA	II	NA	III	NA
NE	NA	NA	NA	NA	NA	NA	NA	NA
E	V	NA	NA	NA	IV	NA	V	NA
SE	NA	NA	NA	NA	NA	NA	NA	NA
S	IV	NA	II	NA	NA	NA	I	NA
SW	NA	NA	NA	NA	NA	NA	NA	NA
W	V	NA	III	NA	I	NA	NA	NA
NW	NA	NA	NA	NA	NA	NA	NA	NA

(c) An example of turning movement table for intersection I1

Fig. 3. General presentation of an intersection (a) and example of a symbolic intersection and its turning movement table (b and c).

Level	Traffic flow (Number of vehicles per hour per lane)	Description
I	[0,100]	Very low
II	[100,200]	Low
III	[200,300]	Medium
IV	[300,500]	High
V	[400,500]	Very high

TABLE I
LEVEL OF TRAFFIC FLOW FOR A TURNING MOVEMENT TABLE

Meta-learners are equipped with deep neural networks (DNN) capable of being sensitive to changes in the task. Meta-learners are parameterized by DNN parameters including λ and θ and are trained using gradient-based learning techniques. No assumption is made for the DNN model on its form such that all the tested meta-learners are based on a model-agnostic meta-learning (MAML) mechanism [31].

IV. EXPERIMENTS

A. Dataset Description

The experiment dataset was generated by running multiple simulations on a simulation platform SUMO. A SUMO simulation provides fundamental models for simulating different layouts of signalized intersections, where the number of lanes, the turning direction of lanes, and the signal phase can be designed. A simulation run in this study is designed to run 60 minutes while resampling the turning movement count every 10 minutes. Meanwhile, the traffic state is recorded every 3 minutes during the whole simulation process.

The turning movement estimation is defined as a classification problem such that model output is a turning movement class. A turning movement class is identified by defining the traffic flow level for each movement direction. We identify five levels of traffic flow (hour/lane) from very low to very high according to engineering practice as shown in Table I.

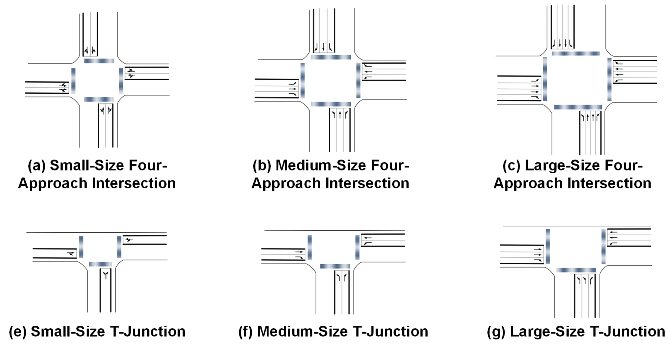


Fig. 4. Six typical layouts with respect to different sizes of four-approach intersections and T-junctions.

To make the estimation class applicable to all types of intersection layouts, we define a general presentation of an intersection that consists of eight origins/destination in Figure 3(a). Figure 3 (b) draws a typical intersection with four approaches and Figure 3 (c) gives an example of a turning movement table where each entry of the matrix is represented by traffic flow level. Since traffic flow is not available for non-existing movement direction, NA in the matrix means that the corresponding traffic flow is not available for the intersection. In principle, $5^{8^2} \simeq 5.421\times10^{44}$ movement classes are generated if an intersection has eight approaches. An intersection usually has four approaches and the size of movement classes is $5^{4^2} \simeq 1.526 \times 10^{11}$ for a four-approach intersection which is the most commonly deployed in real-world applications. To simply our problem, we assume that the same level of traffic flow is assigned to the identical turning direction of each approach at a specific intersection. Besides, the movement of turning around is not considered in this experiment.The total number of turning movement classes for an intersection is reduced to 5^3.

We split the dataset into the source and test sets, and divide a source set into a training set and a validation set. The model gets to see the entire training and validation sets, and then it tries to classify a randomly chosen sample from the test set. In the experiments, we tested four types of classification tasks for turning movement estimation, including 5-way 1-shot, 5-way 5-shot, 20-way 1-shot, and 20-way 5-shot. Specifically, if we trained a model for 5-way, 1-shot classification tasks and defined one target sample, then the model is trained on 5 examples (one per class) and test on another example.

B. Experimental Settings

1) Traffic simulation settings: To examine the model effectiveness under different spatiotemporal traffic environments, six different sizes of intersections are built and simulated using SUMO, three of which are identified as four-approach intersections and the other three are three-approach junctions (see Figure 4). The synthetic data can be obtained by running plenty of offline simulations as required. During simulations, all signal cycle lengths are set as 160 seconds with an equal division on the green time, while amber time and all-red time are 6 seconds and 4 seconds, respectively. We use link space mean speed as the observed data, which is measured by calculating the harmonic mean of passing vehicles speeds. This measure is in accordance with the data collection situations in the real-world since consecutive pictures or videos of a road segment or floating cars are universally used to track the speed of individual vehicles.

For all models, we use cross-entropy error between the predicted and true class as both across-task and task-specific loss functions. Classification accuracy is applied to evaluate meta-learners, implying the rate of correct classifications. The accuracy metric is applied to all of the training, validation, and test datasets.

$$\mathcal{L}_{cross} = -\sum_{u=1}^{U} b_{\hat{C}_u,C_u} \log(p_{\hat{C}_u,C_u}) \quad (8)$$

$$ACC = \frac{\sum_{u=1}^{n} b_{\hat{C}_u,C_u}}{n_{samples}} \quad (9)$$

where L_{cross} and ACC denote the cross-entry loss and model accuracy. $\hat{C}_u$ and C_u represents predicted class label and actual class label. $b(\cdot)$ denotes a binary indicator where its value is 1 when the predicted class label is correct, otherwise its value equals zero. $p_{\hat{C}_u,C_u}$ is the predicted probability that predicted class label $\hat{C}_u$ is the actual label C_u. n denotes the total number of all available classes and n is the identical meaning for n-way tasks in few-shot learning. $n_{samples}$ refer to the total number of samples.

For data split, we use 64% of all data for training, 16% data for validation, and the remaining 20% data for testing. Each model experiment consisted of 150 training epochs. 400 iterations were carried out in each epoch. According to the properties of meta-learning, a small number of gradient steps were applied for inner-loop updates. We used five training steps per iteration.

During training, the training tasks were continually generated without repeating previously sampled tasks, while 200 unique evaluation tasks were applied for validating the model at the end of each epoch. Upon completion of 200 training epochs, five best models, ever performing on the validation set, were used on the test set, and produces the final test performance. The models were trained with the Adam optimizer and used a batch size of 8. The data shape is resized to $32 \times 32 \times 4$.

The deep neural network architecture was implemented using PyTorch, containing four convolutional neural network (CNN)-based stages. Each stage constituted of a 2D convolution layer, a ReLU layer, a pooling layer, and a normalization layer. An identical number of filters, kernel size, and stride are set as 48, 3 and 2 for all convolutional layers, respectively.

C. Case Study

We further illustrate the effectiveness of the framework by performing a simulation-based case study carried out on a typical intersection layout shown in Figure 4(b). MAML++ was applied as the meta-learner in the modeling framework. The estimation task is designed as a 5-way 5-shot case. The trained model runs every 15 minutes while a 2-hour time horizon is considered for each run, which is divided into detection horizon and simulated horizon. The observed traffic states over the detection horizon are obtained for collecting the input data for the meta-learner, and the turning movement can be predicted for the simulated horizon. Consequently, the predicted traffic states can be used to compare with the collected traffic states in the next run 15 minutes later, which realizes the online testing of the effectiveness of the proposed traffic simulation model.

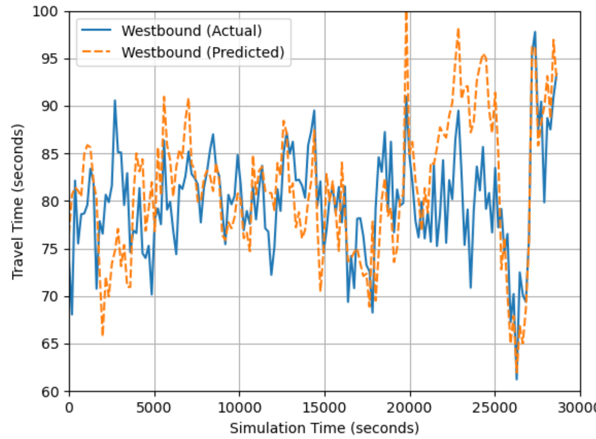


Fig. 5. Actual and predicted travel time for westbound approach.

Table II demonstrates the execution results of an eight-hour simulation experiment. The traffic flow is randomly generated to design a real scenario and the value of which varies every 30 minutes following a trend from increasing to decreasing. This is grounded in reality, for instance, when traffic reaches morning peak or evening peak and gradually returns to normal. During the whole simulation process, the proposed framework runs every 15 minutes after the first-hour warm-up and the predicted turning movement class is obtained to compare with the actual situation. By comparing the two columns of 'Class ID' and 'Predicted Class ID' in Table II, it can be found that only three out of 30 classes are inaccurately predicted, indicating a 90% prediction accuracy. To conceptualize this, we also provide random-generated predicted traffic flows based on the predicted class ID (see Table I for the classification).

For a clearer comparison with the real traffic states, we compared the actual and predicted average travel times when the model is in use every 180 seconds for the four approaches. Two forecast error terms, mean absolute percentage error (MAPE) and mean absolute error (MAE), were measured. Table III summarizes these two error results for four approaches after the an eight-hour simulation presented in Table II. We have several observations: 1) The proposed framework provides a reasonable representation of urban

Time Interval	Traffic flow (vehs/hr)			Class ID	Predicted Class ID		Predicted traffic flow (vehs/hr)		
	Left	Straight	Right		1^{th} 15 mins	2^{rd} 15 mins	Left	Straight	Right
Warm-up (1 hr)	38	125	66	–	–	–	–	–	–
$0-30$ mins	38	125	66	6	6	6	67	135	66
$30-60$ mins	79	178	102	7	7	7	55	137	148
$60-90$ mins	123	188	180	32	32	32	143	166	135
$90-120$ mins	178	209	192	37	37	37	180	242	141
$120-150$ mins	234	289	288	63	38	63	220	264	257
$150-180$ mins	241	276	320	64	64	64	253	258	367
$180-210$ mins	192	221	247	38	63	38	215	250	241
$210-240$ mins	188	182	129	32	32	32	155	148	161
$240-270$ mins	92	173	124	7	7	7	64	165	118
$270-300$ mins	123	89	144	27	27	27	179	65	170
$300-360$ mins	193	102	122	32	32	32	139	173	168
$360-390$ mins	99	89	83	1	26	1	101	67	62
$390-420$ mins	122	102	89	31	31	31	175	149	47
$420-450$ mins	64	78	122	2	2	2	78	92	160
$450-480$ mins	125	102	35	31	31	31	146	164	54

TABLE II

AN EXAMPLE OF PERFORMING TRAFFIC FLOW CHANGE PATTERN, COMPARED WITH ITS PREDICTED TRAFFIC FLOW RESULTED BY THE PROPOSED MODELING FRAMEWORK.

Error	Westbound	Eastbound	Southbound	Northbound
MAPE	6.24%	5.99%	5.68%	6.01%
MAE	4.94	4.75	4.53	4.80

TABLE III

PREDICTION ERRORS OF TRAVEL TIME FOR APPLYING THE FRAMEWORK IN A CASE STUDY

traffic operation in the near future due to the resulted low MAPE and MAE values. 2) As shown in Figure 5, sudden changes in traffic pattern deteriorate the prediction performance, but the model can adjust its prediction within a short time. This demonstrates that our proposed method is able to automatically build an online traffic simulation model and self-adapt its effectiveness according to real-time traffic data.

V. CONCLUSIONS

Online traffic simulation requires an automated, real-time, and efficient approach for adapting model parameters, especially in the context of a complicated urban network with heterogeneous road attributes. In this study, a meta-learning based modeling framework was proposed for online traffic simulation because of its advantages in dealing with the insufficient real-time data and the complexity of online model training. Computational experiments were conducted to evaluate the proposed framework with different meta-learners, and it is shown that the prediction has a good fitness with observations. One limitation of this experiment lies in the fact that we reduce the number of classes via a grouping of turning directions. However, the actual traffic pattern could be quite complex, especially when the number of intersections increases. Further investigation can be carried out to solve the classification problem using the continual learning method [32], which enables learning online to adapt to class increase without forgetting its past learning experiences.

REFERENCES

[1] J. Barceló, "Fundamentals of traffic simulation," 2010.

[2] 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 Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1616–1626, 2021.

[3] F.-Y. Wang, "Real-time social transportation with online social signals," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 909–914, 2014.

[4] Y. Zhao, H. Gao, S. Wang, and F.-Y. Wang, "A novel approach for traffic signal control: A recommendation perspective," *Intelligent Transportation Systems Magazine, IEEE*, vol. 9, no. 3, pp. 127–135, 2017.

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

[6] J. Jin, D. Rong, Y. Pang, P. Ye, Q. Ji, X. Wang, G. Wang, and F.-Y. Wang, "An agent-based traffic recommendation system: Revisiting and revising urban traffic management strategies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2022.

[7] F.-Y. Wang, J. Guo, G. Bu, and J. J. Zhang, "Mutually trustworthy human-machine knowledge automation and hybrid augmented intelligence: mechanisms and applications of cognition, management, and control for complex systems," *Frontiers of Information Technology & Electronic Engineering*, pp. 1–16, 2022.

[8] B. Park and J. Won, "Microscopic simulation model calibration and validation handbook." 2006.

[9] F.-Y. Wang and S. Tang, "A framework for artificial transportation systems: From computer simulations to computational experiments," in *Proceedings of 2005 IEEE International Conference on Intelligent Transportation Systems (ITSC'05)*, 2005, pp. 1130–1134.

[10] C. Ou and F. Karray, "Enhancing driver distraction recognition using generative adversarial networks," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 385–396, 2019.

[11] Y. Chen, X. Wang, L. Li, and F.-Y. Wang, "Traffic situational awareness research and development enhanced by social media data: the state of the art and prospects," *Chinese Journal of Intelligent Science and Technology*, vol. 4, no. 1, p. 1, 2022.

[12] Y. Wang, T. Wo, H. Yin, J. Xu, H. Chen, and K. Zheng, "Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1227–1235, 2019.

[13] J. Jin and X. Ma, "A non-parametric bayesian framework for traffic-state estimation at signalized intersections," *Information Sciences*, vol. 498, pp. 21–40, 2019.
[14] J. Jin, X. Ma, and I. Kosonen, "An intelligent control system for traffic lights with simulation-based evaluation," *Control engineering practice*, vol. 58, pp. 24–33, 2017.
[15] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.
[16] X. Li, Z. Guo, X. Dai, Y. Lin, and F.-Y. Wang, "Deep imitation learning for traffic signal control and operations based on graph convolutional neural networks," in *Proceedings of 2020 IEEE International Conference on Intelligent Transportation Systems (ITSC'20)*, 2020.
[17] N. Khairdoost, M. Shirpour, M. A. Bauer, and S. S. Beauchemin, "Real-time driver maneuver prediction using lstm," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 714–724, 2020.
[18] Y. Li, X.-Y. Lu, J. Wang, and K. Li, "Pedestrian trajectory prediction combining probabilistic reasoning and sequence learning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 461–474, 2020.
[19] Y. F. Zhao, Q. J. Kong, H. Gao, F. H. Zhu, and F.-Y. Wang, "Parallel management for traffic signal control," in *Proceedings of 2014 IEEE International IEEE Conference on Intelligent Transportation Systems (ITSC'14)*, 2014, pp. 2888–2893.
[20] A. Pell, A. Meingast, and O. Schauer, "Trends in real-time traffic simulation," *Transportation research procedia*, vol. 25, pp. 1477–1484, 2017.
[21] X. Li, Y. Liu, K. Wang, and F.-Y. Wang, "A recurrent attention and interaction model for pedestrian trajectory prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 5, pp. 1361–1370, 2020.
[22] Z. Li, G. Xiong, Z. Wei, Y. Lv, N. Anwar, and F.-Y. Wang, "A semisupervised end-to-end framework for transportation mode detection by using gps-enabled sensing devices," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7842–7852, 2021.
[23] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
[24] C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang, and Y. Lv, "Taxi demand prediction using parallel multi-task learning model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 794–803, 2022.
[25] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 539–546.
[26] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
[27] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *arXiv preprint arXiv:2004.05439*, 2020.
[28] J. Jin, D. Rong, Y. Pang, F. Zhu, H. Guo, X. Ma, and F.-Y. Wang, "Precom: A parallel recommendation engine for control, operations, and management on congested urban traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
[29] J. Jin, X. Ma, and I. Kosonen, "A stochastic optimization framework for road traffic controls based on evolutionary algorithms and traffic simulation," *Advances in Engineering Software*, vol. 114, pp. 348–360, 2017.
[30] Z. Li, G. Xiong, Y. Tian, Y. Lv, Y. Chen, P. Hui, and X. Su, "A multi-stream feature fusion approach for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1456–1466, 2022.
[31] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 1126–1135.
[32] G. Gupta, K. Yadav, and L. Paull, "La-maml: Look-ahead meta learning for continual learning," *ArXiv*, vol. abs/2007.13904, 2020.