2023 IEEE 26th International Conference on
Intelligent Transportation Systems (ITSC)
24-28 September 2023. Bilbao, Bizkaia, Spain

Parallel Learning Based Foundation Model for Networked Traffic Signal Control

Chen Zhao, Xingyuan Dai, Yuanyuan Chen, Yilun Lin, Yisheng Lv and Fei-Yue Wang[†]

2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3503-9946-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITSC57777.2023.10422161

***Abstract*— Networked Traffic Signal Control (NTSC) is a fundamental component of Intelligent Transportation Systems (ITS) and the broader vision of smart city development. While a plethora of intelligent strategies have been developed, the Sim2Real challenge often impedes their full realization. In response, this paper introduces the Parallel Learning-based Adaptive Network for Traffic Signal Control (PLANT) as a foundation model for NTSC. We employ the Wasserstein GAN with Gradient Penalty (WGAN-GP) to generate a wide range of artificial scenarios for robust PLANT training. Further, the Transformer-based Cooperation Mechanism (TCM) is integrated as the primary learner within PLANT, facilitating effective capture of traffic dynamics and knowledge accumulation. This knowledge is readily transferable to real-world applications through meticulous fine-tuning, equipping PLANT to adapt and evolve in alignment with shifting transportation paradigms. Our empirical study on the Hangzhou road network demonstrates PLANT's superiority over both traditional and emerging DRL-based approaches, emphasizing its viability as a potential foundation model for NTSC.**

I. INTRODUCTION

The continual increase in traffic congestion and the associated pollution has hampered urban development and posed a challenge to urban transportation management. To address this issue, researchers have been working on developing Intelligent Transportation Systems (ITS) with the goal of achieving the "6S" goals for smart cities: safety, security, sustainability, sensitivity, service, and smartness [1]. ITS is a broad research field that includes a variety of technologies, such as Networked Traffic Signal Control (NTSC), which aims to improve traffic conditions and reduce travel time by facilitating the cooperation of multiple traffic signals.

There are currently numerous intelligent NTSC algorithms available, with Deep Reinforcement Learning (DRL)-based algorithms in particular outperforming traditional methods [2], [3]. The DRL-based algorithms require virtual scenarios created on traffic simulators such as SUMO [4], Paramics, VISSIM, and CityFlow for training. In the simulations, DRL-based agents manage traffic signals by using trial and error to determine the most effective strategy [5], [6]. Despite their potential, the implementation of DRL-based methods has been difficult, particularly in the transition from simulation to reality (Sim2Real), in two major aspects.

The first challenge is adapting to the complexity and dynamic changes of actual transportation systems. Most DRL-based methods are currently trained and evaluated for a single scenario, as if playing the same game repeatedly. However, traffic systems are complex Cyber-Physical-Social Systems (CPSS) comprised of multiple factors including human and social aspects [7], [8], and traffic flows will vary across different times and locations. Consequently, these DRL-based models tend to overfit to specific scenarios and exhibit poor adaptability to new scenarios, negatively impacting the effectiveness of actual implementations. One potential solution is increasing the amount of training data, but obtaining sufficient real traffic flow data remains challenging due to data privacy and monopoly concerns [9].

The second challenge is effectively implementing DRL algorithms in the real world. To train effectively, the DRL algorithms requires extensive interaction with the actual system, as exploration and optimization are crucial for maximizing their potential [10]. However, such interactions can cause severe traffic congestion or even system crashes, while also demanding significant computing time. The current training process, which begins from scratch, limits the ability to meet real-time strategy development requirements. One possible solution to this problem is leveraging simulation and transfer learning to reduce the need for direct interaction with the actual system during the training phase. Unfortunately, this solution is constrained by the first challenge since the two problems are interconnected.

The parallel learning framework [11], derived from Artificial Systems, Computational Experiments and Parallel Learning (ACP), is an effective approach to addressing the Sim2Real challenge, bridging the modeling gap between artificial and actual systems through description, prediction, and prescription. This framework has been widely used in the fields of ITS [12]–[14], self-driving vehicles [15]–[17], and smart manufacturing [18]–[20]. This paper aims to develop a parallel learning-based foundation model to address the Sim2Real problem for NTSC [21], with the construction process inspired by the "pre-training + fine-tuning" paradigm used in Artificial Intelligence Generated Content (AIGC) models such as Generative Pre-trained Transformer (GPT). The main contributions of this paper are as follows.

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B090921003, in part by National Natural Science Foundation of China under Grant U1811463.

Chen Zhao, Xingyuan Dai, Yuanyuan Chen and Yisheng Lv are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China. They are also with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China.

Yilun Lin is with the Shanghai AI Laboratory, Shanghai 200232, China.

Fei-Yue Wang is with the State Key Laboratory for Management and Control of Complex Systems, Chinese Academy of Sciences, Beijing 100190, China, and also with the Macao Institute of Systems Engineering, Macau University of Science and Technology, Macao 999078, China.

[†]Corresponding Author.

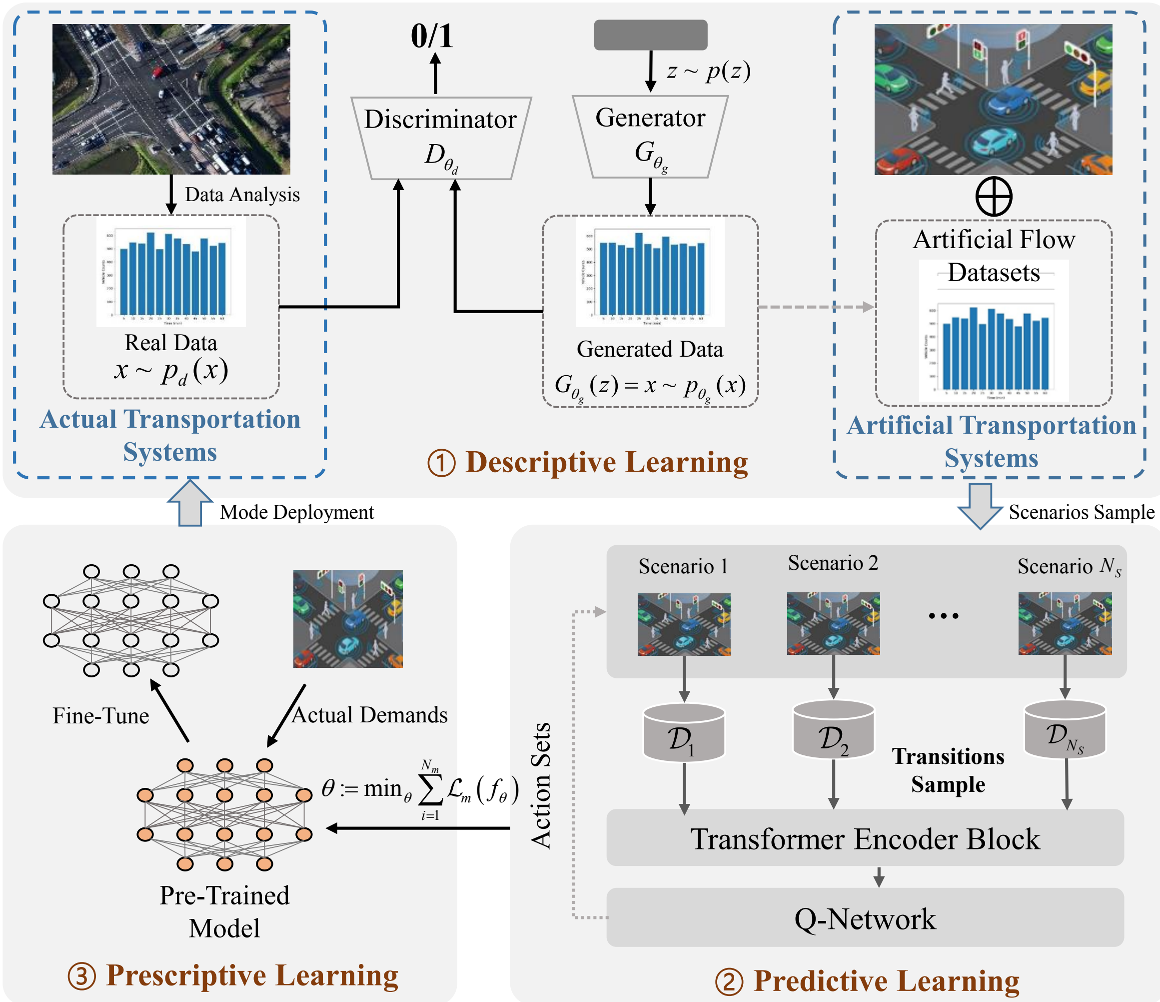


Fig. 1: The operational process of PLANT: Descriptive learning, predictive learning and prescriptive learning.

- We proposed PLANT (Parallel Learning-based Adaptive Network for Traffic Signal Control) as a foundation model for NTSC, utilizing diverse artificial scenarios to enhance control agents' traffic knowledge. This approach facilitates efficient knowledge transfer to actual systems, addressing the Sim2Real problem.
- We used the Wasserstein GAN with Gradient Penalty (WGAN-GP) model [22] to supplement the limited amount of real traffic flow data, resulting in an artificial system with a broader range of traffic scenarios for training PLANT.
- We employed the Transformer-based Cooperation Mechanism (TCM) [23] as a single-scene learner and integrated it with a multi-scene optimization mechanism to effectively use the Transformer for capturing traffic dynamics and acquiring more knowledge.
- A case study on the Hangzhou road network demonstrates the generality and robustness of PLANT, outperforming both traditional algorithms and cutting-edge DRL-based techniques.

The remainder of this paper is organized as follows. The formulation of multi-scenarios optimization for NTSC is described detailedly in Section II. The construction details of PLANT are presented in Section III. A case study of the proposed model is conducted in Section IV to verify the performance. Finally, this paper is summarized in Section V.

II. METHODOLOGY

A. Overall Architecture

As illustrated in Fig. 1, the framework synchronizes actual transportation systems and their counterparts using the using the pre-trained model, PLANT. The construction of PLANT are conducted through three steps of parallel learning: description, prediction, and prescription. In the first step of the process, we employ the WGAN-GP model to generate artificial traffic flow datasets that reflect the task distribution of real data, which are utilized in ATS to reproduce different traffic conditions [24]. The second step involves utilizing TCM across multiple scenarios, similar as multi-task learning [25], to create a robust model for predicting actions. The pre-trained model stores a wealth of traffic flow knowledge that can be transferred to new traffic scenarios. Finally, the pre-trained model is fine-tuned using only a few samples from the actual system to meet real-time traffic needs. This allows the model to provide an optimal signal plan in a timely manner, improving traffic efficiency and reducing congestion.

B. Descriptive Learning: WGAN-GP Based Scenarios Engineering

Collecting enough data on actual transportation systems can be difficult due to data monopolies and privacy issues, so there are only a limited number of samples which may not accurately reflect the complexity of traffic in the real world [26]. We choose to use WGAN-GP to create artificial traffic flow datasets and simulate different traffic conditions in ATS in order to guarantee that our model is versatile enough to address different traffic scenarios. We start by collecting traffic flow data and counting the number of vehicles entering the network during each time period. We then use WGAN-GP to fit the task distribution of the collected traffic flows and optimize the two loss functions.

$$\begin{aligned} \mathcal{L}_d = & \mathbb{E}_{\tilde{x}\sim P_g} D(\tilde{x}) - \mathbb{E}_{x\sim P_r} D(x) \\ & + \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}} \left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2 \right] \\ \mathcal{L}_g = & -\mathbb{E}_{\tilde{x}\sim P_g} D(\tilde{x}), \end{aligned} \tag{1}$$

where $\mathcal{L}_d$ and $\mathcal{L}_g$ denote the loss function of the of the discriminator D_{θ_d} and the generator G_{θ_g} respectively. P_r and P_g denote the distribution of real data and generated data respectively. It is noted that, in $\mathcal{L}_d$, the first half is same as the WGAN model, and the second half is a gradient penalty on discriminator D_{θ_d}, which is used to prevent the gradient of $\mathcal{L}_d$ from vanishing.

Once the generative model has been trained, we can use it to generate artificial traffic flow datasets that closely resemble real traffic flow data. To replicate various vehicle routes, we assemble a group of vehicle agents with varying driving tendencies that exhibit distinct behavior as they approach intersections [27]. We use various algorithms to build vehicle travel trajectories with different traffic patterns, allowing us to simulate a wide range of traffic scenarios in ATS.

C. Predictive Learning: Pre-Training with Multi-Scenarios

1) **Modeling:** Given a large-scale traffic road network $G(I, E)$, there are a set of N intersections $I = \{I_1, ..., I_N\}$ and a set of road edges $E \subseteq \{(x, y)|\ x, y \in I \text{ and } x \neq y\}$. Each intersection I_i is controlled by one agent i and its control process is formulated as a Markov decision process (MDP) $\langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{R}_i, \gamma_i \rangle$, which contains a finite set of states $\mathcal{S}_i$, a finite set of actions $\mathcal{A}_i$, a reward function $\mathcal{R}_i$, and a discounted factor γ_i. The specific definitions of relevant terms are provided here.

At time t, the state of agent i is defined as a partial observation of the environment state: $s_i^t = [[v_l]_{l=1}^{l_n}, [r_l]_{l=1}^{l_n}, p] \in \mathcal{S}_i,\ i = 1, 2, ..., N$, where l_n is the number of incoming lanes at intersection I_i, and the variables r_l, k_l represent the average speed of vehicles and the road occupancy along each incoming lane l. The variable p is a one-hot encoding for the

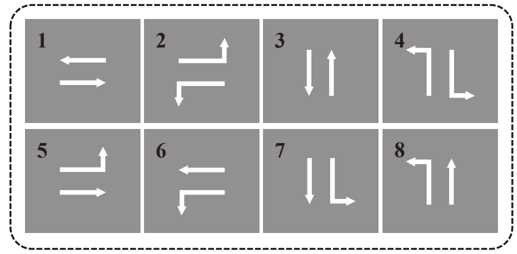


Fig. 2: Eight typical signal phases in NTSC problems.

current signal phase. Note that the speed v is normalized by dividing it by the maximum speed limit for each lane.

Consistent with previous work [23], the action of each agent is to choose from eight permissible phases shown in Fig. 2 for the next time interval. Given the current state s_i^t, agent i selects a phase p_i^t from the candidate set $\mathcal{A}_i$ as its action a_i^t based on the control policy π_i. Each action is executed for a time interval Δt with a yellow light time t_y between phase switching to ensure safety.

At time t, agent i takes an action a_i^t and receives immediate feedback $r_i^t \in \mathcal{R}_i$ from the environment. This paper aims to minimize the travel time for all vehicles in the system, which is difficult to optimize directly. Therefore, the reward for agent i is defined as $r_i^t = \sum_{l=1}^{l_n} u_{i,l}^t$, where $u_{i,l}^t$ represents the queue length on incoming lane l at time t.

To achieve the long-term optimal traffic policy, a discount factor $\gamma_i \in (0, 1]$ is set to balance exploration and utilization in policy optimization.

We consider a road network $G(I, E)$ sampled from scenario sets $\mathcal{M} = \{\mathcal{M}_1, ..., \mathcal{M}_{N_s}\}$, which have similar flow distributions $\mathcal{E}$. Each agent i is modeled as a MDP in each scenario m. At time t, agent i observes state $s_{i,m}^t \in \mathcal{S}_{i,m}$ from the scenario m, and takes an action $a_{i,m}^t \in \mathcal{A}_{i,m}$. The reward $\mathcal{R}_{i,m}^t(s, a)$ in time t is defined as $\mathcal{R}_{i,m}^t(s, a) = \mathbb{E}\left[\mathcal{R}_{t+1} \mid \mathcal{S}_{i,m}(t) = s, \mathcal{A}_{i,m}(t) = a\right]$. The objective of all agents is to learn the optimal policies $\{\pi_{i,m}(a \mid s)\}_{i=1}^N$ to optimize the traffic conditions across all scenarios. To achieve this, we introduce a shared model and define the learner f with learnable parameters θ to map the state space $\mathcal{S}_{i,m}$ to the action space $\mathcal{A}_{i,m}$. The loss function $\mathcal{L}_m(f_\theta)$ is used to measure the deviation between the predicted actions and the actual actions taken. Our goal is to minimize $\mathcal{L}(f_\theta)$ across all scenarios to find the optimal parameters θ, which is defined as:

$$\theta := \min_\theta \sum_{i=1}^{N_m} \mathcal{L}_m(f_\theta). \tag{2}$$

2) **Training Process:** In the predictive learning process, we use TCM as the base learner, training it in multiple scenarios to obtain a more robust control agent. In each episode, we sample a signal control task $\mathcal{M}_m$ from the set of traffic scenarios [28]. At each step, each control agent obtains states $s_{i,m}^t$ from the interactive scenario and embeds them into the latent space using Eq. (3).

$$h_i^m = \sigma(s_{i,m}^t W_e + b_e). \tag{3}$$

The generated hidden state h_i^m denotes the current traffic information of the i-th intersection. We then use a Transformer [29] to facilitate the communication of status information between multiple intersections, enabling us better capture the global dynamic changes of traffic flows. It consists of three main steps: obtaining the importance scores e_{ij}^m between different intersections using Eq. (4), distributing attention using Eq.(5), and integrating key information using Eq. (6). Through these steps, we can obtain a dynamic global perception of each intersection and achieve better cooperation with other agents.

$$e_{ij}^m = \frac{(h_i^m W^Q)(h_j^m W^K)^{\mathrm{T}} + p_{ij}}{\sqrt{d_z}}. \tag{4}$$

$$\alpha_{ij}^m = \frac{\exp(e_{ij}^m/\beta)}{\sum_{j\in N}\exp(e_{ij}^m/\beta)}. \tag{5}$$

$$z_i^m = \sigma(W_z(\frac{1}{\mathrm{H}}\sum_{h=1}^{\mathrm{H}}\sum_{j=1}^{\mathrm{N}}\alpha_{ij}^m(h_j^m W^K + p_{ij}) + b_z)). \tag{6}$$

Next, we adopt a Q-network to evaluate the Q value of state-action pairs using Eq.(7) and the action of agent i can be predicted by maximizing $Q(s_{i,m}^t) \in \mathbb{R}^{|\mathcal{A}|}$. The state transitions $\langle s_{i,m}^t, a_{i,m}^t, r_{i,m}^t, s_{i,m}^{t+1}\rangle$ are stored in the replay buffer $\mathcal{D}_m$ for the corresponding scenario.

$$Q(s_{i,m}^t) = z_i^m W_p + b_p. \tag{7}$$

During the multi-scenario training process, the parameters θ are updated at every episode by gradient descent on a sampled scenario. One example of taking a gradient step is as follows:

$$\theta \leftarrow \theta - \alpha\nabla_\theta \mathcal{L}_m\left(f_\theta; \mathcal{D}_m\right), \tag{8}$$

where α is the step size, $\mathcal{L}_m$ is the loss function to optimize θ on scenario $\mathcal{M}_m$:

$$\mathcal{L}_m\left(f_\theta; \mathcal{D}_m\right) = \mathbb{E}_{s,a,r,s'\sim\mathcal{D}_m}\left[\left(r + \gamma\max_{a'} Q\left(s', a'; f_{\theta^-}\right) - Q\left(s, a; f_\theta\right)\right)^2\right], \tag{9}$$

where γ is the discount factor for future reward, θ denotes all the trainable parameters in PLANT, and θ' is a periodically updated copy of θ in target Q-network.

D. Prescriptive Learning: Fine-Tuning for Real-World

During the prescriptive learning phase, we use the pre-trained PLANT model as the foundation module for NTSC and fine-tune it to meet the actual demands in the real world. In actual transportation systems, we can use cameras, induction coils, and other sensing devices [30] to gather the data we need to fine-tune our model. These data are organized into transition pairs $\langle s, a, r, s'\rangle$ and stored in $\mathcal{D}_r$. We can then re-optimize the model using Eq.(10).

$$\theta \leftarrow \theta - \alpha\nabla_\theta \mathcal{L}_r\left(f_\theta; \mathcal{D}_r\right). \tag{10}$$

The "local simple remote complex" design principle [31] is used to build and deploy the PLANT model. This entails using the massive computational capacity of cloud-based servers to offline train the model with artificial scenario data and fine-tune the parameters before deploying it to the actual system. Once the models have been fine-tuned, we download them to the actual transportation systems with fixed-weight parameters to support real-time signal control. This cycle can be repeated every hour or every day until the parameters are fine-tuned. We can avoid instability and ensure the real-time reliability and robustness of our application by using this mechanism. It is evident that the proposed framework in this paper can be seamlessly integrated with other DRL-based algorithms.

III. CASE STUDY

A. Scenario

In this paper, we use a public real-world dataset [1] ($\mathcal{M}_r$) to evaluate our model, where the road network is shown in Fig. 3 and traffic flows are captured from intersection surveillance cameras. We have made statistical analysis of the real-world traffic flow, and the findings are depicted in Table I. Subsequently, we utilize WGAN-GP to fabricate artificial datasets $\mathcal{M}_a^t$ and $\mathcal{M}_a^e$ for training and validation, replicating the actual traffic flow patterns. Two samples are exhibited in Fig. 4. Lastly, these traffic flow datasets are fed into Cityflow to construct diverse artificial traffic scenarios to support our computational experiments.

TABLE I: Data statistics of the actual and artificial traffic flow datasets.

Datasets	Agents	Arrival Rate (Vehicles / Min)			
		Min	Mean	Max	Std
$\mathcal{M}_r$		75	109	134	19.87
$\mathcal{M}_a^t$	16	$\mathcal{M}_a^t = \{\mathcal{M}_1, ..., \mathcal{M}_{30}\} \sim P_{\mathcal{M}_r}$			
$\mathcal{M}_a^e$		$\mathcal{M}_a^t = \{\mathcal{M}_{31}, ..., \mathcal{M}_{35}\} \sim P_{\mathcal{M}_r}$			

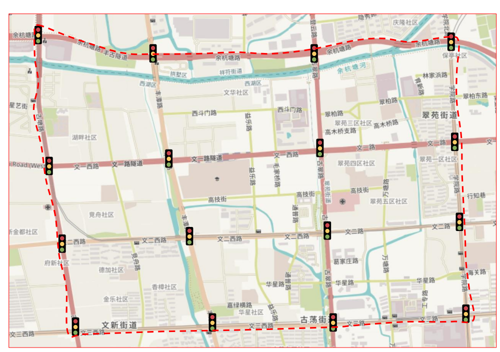
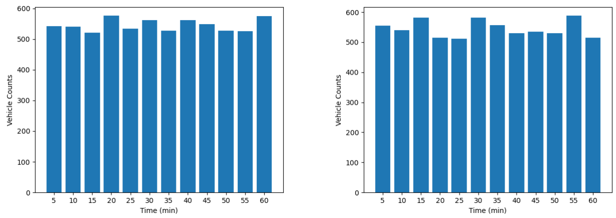


Fig. 3: The field testing area in Hangzhou.

[1] https://traffic-signal-control.github.io/#open-datasets



(a) (b)

Fig. 4: Samples of artificial traffic flow distributions in $\mathcal{M}_a^t$.

B. Experimental Settings

In our experiments, we compared two conventional algorithms, Fixed-Time and MaxPressure, and three DRL-based algorithms, Prsslight [32], Colight [33], and TCM [23], using all parameters detailed in the original research and open source code. The RL-based models were trained with $\mathcal{M}_r$ in one scenario, while the PLANT model was trained with $\mathcal{M}_a^t$ in a variety of scenarios, and their models were tested with both $\mathcal{M}_r$ and $\mathcal{M}_a^e$. It is noted that the TCM can be regarded as a single-scenario training version of PLANT.

In the experiments, we set the simulation time $T = 3600s$, control period $\Delta t = 10s$ and yellow time $t_y = 3s$. For each epoch in the multi-scene training procedure of PLANT, we randomly select a scenario from $\mathcal{M}_r$ to interact with the control agent, perform 5 episodes, and store the state transition pairs in the relevant replay buffer. The rest of parameters in PLANT are in accordance with TCM.

C. Performance Comparison

We use two metrics, similar to those in existing work [32], [33], to provide a comprehensive overview of the spatio-temporal journey relationship of vehicles entering the network. Table II displays the average travel time of vehicles, which is a metric for assessing the duration of vehicles in the network. Table III displays the throughput of vehicles, providing an understanding of how many vehicles successfully complete their journeys in the network.

It is shown that traditional algorithms are more reliable and adaptable in multi-scenario testing than DRL-based algorithms. The lack of transferability is caused by the high probability of over-fitting when models are only trained on one scenario. PressLight is more susceptible to overfitting than Colight and TCM because it lacks the ability to dynamically recognize traffic flow patterns, whereas the latter two have GAT module and Transformer mechanism for global information integration.

TABLE II: Results on $\mathcal{M}_a^e$ and $\mathcal{M}_r$ w.r.t average travel time.

Algorithms	$\mathcal{M}_a^e$				$\mathcal{M}_r$
	Max	Mean	Median	Min	
Fixed-Time	611.92	599.48	598.34	583.27	627.40
MaxPressure	478.66	474.04	473.91	467.42	439.70
PressLight	1102.83	1050.62	1053.80	980.84	427.29
Colight	730.76	637.63	625.19	590.83	410.81
TCM	678.48	590.19	578.09	534.01	**361.82**
PLANT	**374.91**	**370.56**	**370.28**	**367.52**	378.78

TABLE III: Results on $\mathcal{M}_a^e$ and $\mathcal{M}_r$ w.r.t throughput.

Algorithms	$\mathcal{M}_a^e$				$\mathcal{M}_r$
	Max	Mean	Median	Min	
Fixed-Time	27175	27021	27050	26811	26693
MaxPressure	29481	29222	29205	29056	28475
PressLight	17123	15918	15679	15077	28407
Colight	27491	26362	26788	23777	28116
TCM	28350	27576	27805	25655	**28635**
PLANT	**30378**	**30148**	**30129**	**29969**	28050

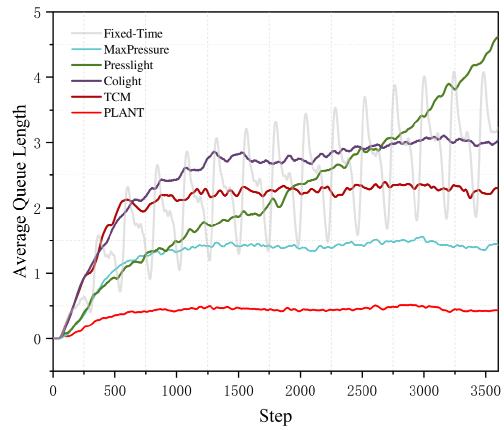


Fig. 5: The average queue length of PLANT (red) and other compared methods measured on $\mathcal{M}_a^e$.

Comparing the TCM algorithm with PLANT can be viewed as an ablation experiment of the multi-scene optimization approach. Our findings indicate that training TCM on a specific scenario $\mathcal{M}_r$ only resulted in the best evaluation results on $\mathcal{M}_r$, but performed poorly in the multi-scene testing on $\mathcal{M}_a^t$, suggesting that the TCM model overfits data from a single scene. In contrast, the PLANT model trained on multiple scenarios performed more robustly, achieving comparable effectiveness to TCM on a specific scenario ($\mathcal{M}_r$) and also performing well on the multi-scene testing ($\mathcal{M}_a^t$). These results demonstrate that PLANT can serve as a robust and dependable foundation model for NTSC.

In a more intuitive way, Fig. 5 shows the average queue length at all intersections during the evaluated episode. The PLANT-controlled scenario exhibits fewer congested intersections, resulting in reduced vehicle travel time and improved operational efficiency.

IV. CONCLUSIONS

This paper presents the Parallel Learning-based Adaptive Network for Traffic Signal Control (PLANT) as a foundation model to tackle the Sim2Real problem. PLANT employs a three-stage learning process, including descriptive, predictive, and prescriptive learning, to effectively generalize and transfer transportation knowledge across different scenarios and real-world applications. The integration of WGAN-GP in the descriptive learning stage generates diverse artificial scenarios, enriching PLANT's training data. The Transformer-based Cooperation Mechanism (TCM) in the predictive learning stage extracts valuable transportation knowledge across varying conditions. Finally, the prescriptive learning stage applies the acquired knowledge to actual transportation systems, enabling PLANT to adapt and evolve in response to ever-changing transportation requirements. A case study on the Hangzhou road network highlights PLANT's potential as a reliable foundation model for networked traffic signal control. Future research may focus on enhancing PLANT's adaptability and performance in more complex scenarios.

REFERENCES

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transport. Syst.*, vol. 11, no. 3, pp. 630–638, 2010.

[2] S. D. Kumaravel, A. A. Malikopoulos, and R. Ayyagari, "Optimal coordination of platoons of connected and automated vehicles at signal-free intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 186–197, 2022.

[3] E. Moradi-Pari, D. Tian, H. N. Mahjoub, and S. Bai, "The smart intersection: A solution to early-stage vehicle-to-everything deployment," *IEEE Intell. Transport. Syst. Mag.*, vol. 14, no. 5, pp. 88–102, 2022.

[4] 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 *2018 ITSC*, 2018, pp. 2575–2582.

[5] W. Qiu, W. ShangGuan, L. Chai, B. Cai, and J. Chen, "Parallel hierarchical control-based efficiency enhancement for large-scale virtual reality traffic simulation," *IEEE Intell. Transport. Syst. Mag.*, vol. 14, no. 4, pp. 145–162, 2022.

[6] Z. Wang, K. Han, and P. Tiwari, "Digital twin-assisted cooperative driving at non-signalized intersections," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 198–209, 2022.

[7] F.-Y. Wang, Y. Lin, P. A. Ioannou, L. Vlacic, X. Liu, A. Eskandarian, Y. Lv, X. Na, D. Cebon, J. Ma, L. Li, and C. Olaverri-Monreal, "Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems," *IEEE Trans. Intell. Transport. Syst.*, vol. 24, no. 10, 2023.

[8] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transport. Syst.*, vol. 16, no. 2, pp. 865–873, 2014.

[9] X. Li, P. Ye, J. Jin, F. Zhu, and F.-Y. Wang, "Data augmented deep behavioral cloning for urban traffic control operations under a parallel learning framework," *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 6, pp. 5128–5137, 2021.

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

[11] Q. Miao, Y. Lv, M. Huang, X. Wang, and F.-Y. Wang, "Parallel learning: Overview and perspective for computational learning across Syn2Real and Sim2Real," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 603–631, 2023.

[12] C. Zhao, Y. Lv, J. Jin, Y. Tian, J. Wang, and F.-Y. Wang, "DeCAST in TransVerse for parallel intelligent transportation systems and smart cities: Three decades and beyond," *IEEE Intell. Transport. Syst. Mag.*, vol. 14, no. 6, pp. 6–17, 2022.

[13] J. Wang, T. Shen, Y. Tian, Y. Wang, C. Gou, X. Wang, F. Yao, and C. Sun, "A parallel teacher for synthetic-to-real domain adaptation of traffic object detection," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 441–455, 2022.

[14] C. Zhao, X. Dai, Y. Lv, J. Niu, and Y. Lin, "Decentralized autonomous operations and organizations in transverse: Federated intelligence for smart mobility," *IEEE Trans. Syst. Man Cybern, Syst.*, vol. 53, no. 4, pp. 2062–2072, 2023.

[15] L. Li, X. Wang, K. Wang, Y. Lin, J. Xin, L. Chen, L. Xu, B. Tian, Y. Ai, J. Wang, D. Cao, Y. Liu, C. Wang, N. Zheng, and F.-Y. Wang, "Parallel testing of vehicle intelligence via virtual-real interaction," *Sci. Robot.*, vol. 4, no. 28, p. eaaw4106, 2019.

[16] K. Liu, L. Li, Y. Lv, D. Cao, Z. Liu, and L. Chen, "Parallel intelligence for smart mobility in cyberphysical social system-defined metaverses: A report on the international parallel driving alliance," *IEEE Intell. Transport. Syst. Mag.*, vol. 14, no. 6, pp. 18–25, 2022.

[17] L. Chen, Y. Zhang, B. Tian, Y. Ai, D. Cao, and F.-Y. Wang, "Parallel driving OS: A ubiquitous operating system for autonomous driving in CPSS," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 886–895, 2022.

[18] F.-Y. Wang, J. Yang, X. Wang, J. Li, and Q.-L. Han, "Chat with ChatGPT on industry 5.0: Learning and decision-making for intelligent industries," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 4, pp. 831–834, 2023.

[19] X. Wang, J. Yang, Y. Wang, Q. Miao, F.-Y. Wang, A. Zhao, J.-L. Deng, L. Li, X. Na, and L. Vlacic, "Steps toward industry 5.0: Building "6S" parallel industries with cyber-physical-social intelligence," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 8, pp. 1692–1703, 2023.

[20] F.-Y. Wang, "The DAO to MetaControl for metasystems in metaverses: The system of parallel control systems for knowledge automation and control intelligence in CPSS," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 11, pp. 1899–1908, 2022.

[21] C. Zhao, X. Dai, Y. Lv, Y. Tian, Y. Ren, and F.-Y. Wang, "Foundation models for transportation intelligence: ITS convergence in TransVerse," *IEEE Intell. Syst.*, vol. 37, no. 6, pp. 77–82, 2022.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein GANs," in *NIPS'17*, 2017, pp. 5767–5777.

[23] C. Zhao, X. Dai, X. Wang, L. Li, Y. Lv, and F.-Y. Wang, "Learning Transformer-based cooperation for networked traffic signal control," in *2022 ITSC*, 2022, pp. 3133–3138.

[24] E. de Gelder, J.-P. Paardekooper, A. K. Saberi, H. Elrofai, O. O. d. Camp, S. Kraines, J. Ploeg, and B. De Schutter, "Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 300–314, 2022.

[25] E. Kargar and V. Kyrki, "Increasing the efficiency of policy learning for autonomous vehicles by multi-task representation learning," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 701–710, 2022.

[26] V. Fors, B. Olofsson, and E. Frisk, "Resilient branching MPC for multi-vehicle traffic scenarios using adversarial disturbance sequences," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 838–848, 2022.

[27] W. Wang, J. Wang, D. Zhao, and K. Jin, "A revised logit model for stochastic traffic assignment with a relatively stable dispersion parameter," *IEEE Intell. Transport. Syst. Mag.*, vol. 14, no. 2, pp. 92–104, 2022.

[28] X. Tang, K. Yang, H. Wang, J. Wu, Y. Qin, W. Yu, and D. Cao, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 7, no. 4, pp. 849–862, 2022.

[29] Y. Tian, J. Wang, Y. Wang, C. Zhao, F. Yao, and X. Wang, "Federated vehicular Transformers and their federations: Privacy-preserving computing and cooperation for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 456–465, 2022.

[30] R. Singh, D. Saluja, and S. Kumar, "R-Comm: A traffic based approach for joint vehicular radar-communication," *IEEE Trans. Intell. Veh.*, vol. 7, no. 1, pp. 83–92, 2022.

[31] Z. Li, C. Chen, and K. Wang, "Cloud computing for agent-based urban transportation systems," *IEEE Intelligent Systems*, vol. 26, no. 1, pp. 73–79, 2011.

[32] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *KDD'19*, 2019, pp. 1290–1298.

[33] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *CIKM'19*, 2019, pp. 1913–1922.