






# GAN and Multi-Agent DRL Based Decentralized Traffic Light Signal Control

Zixin Wang, *Graduate Student Member, IEEE*, Hanyu Zhu , Mingcheng He , Yong Zhou , *Member, IEEE*, Xiliang Luo , *Senior Member, IEEE*, and Ning Zhang , *Senior Member, IEEE*

**Abstract**—Adaptive traffic light signal control (ATSC) is a promising paradigm for alleviating traffic congestion in intelligent transportation systems. Most of the existing methods require heavy traffic data exchange among neighboring intersections to achieve collaborative ATSC, which may not be supported by bandwidth-limited communication links in practice. In this article, we develop a communication-efficient decentralized ATSC framework for traffic networks with multiple intersections, where each intersection only exchanges traffic statistics with its neighboring intersections. In particular, the proposed framework consists of a generative adversarial network (GAN) based algorithm for traffic data recovery, and a multi-agent deep reinforcement learning (DRL) based decentralized ATSC algorithm for traffic efficiency enhancement. By adopting the value decomposition technique that establishes a nonlinear mapping from the local state-action values to the global reward, each intersection can independently determine its traffic light signal based on its local traffic data while achieving collaboration among neighboring intersections. Our proposed decentralized ATSC framework is scalable to large-scale traffic networks, and is also robust to traffic flow variations via interacting with the environment. Simulations show that our proposed algorithm can significantly reduce the vehicle travel time while maintaining high and stable traffic throughput.

**Index Terms**—Adaptive traffic light signal control, multi-agent deep reinforcement learning, generative adversarial network.

## I. INTRODUCTION

### A. Motivation

WITH the proliferation of vehicles, traffic congestion is becoming increasingly problematic in the metropolis

Manuscript received April 10, 2021; revised August 21, 2021 and November 3, 2021; accepted December 2, 2021. Date of publication December 13, 2021; date of current version February 14, 2022. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61971286. An earlier version of this paper was presented at *Proc. IEEE ICASSP*, Barcelona, Spain, May 2020. The review of this article was coordinated by Prof. Yao Ma. (Corresponding author: Yong Zhou and Xiliang Luo.)

Zixin Wang is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China, also with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wangzx2@shanghaitech.edu.cn).

Hanyu Zhu, Yong Zhou, and Xiliang Luo are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: zhuhy@shanghaitech.edu.cn; zhouyong@shanghaitech.edu.cn; luoxl@shanghaitech.edu.cn).

Mingcheng He is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: mingcheng.he@uwaterloo.ca).

Ning Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: ning.zhang@uwindsor.ca).

Digital Object Identifier 10.1109/TVT.2021.3134329

and has lead to severe consequences in many aspects, including huge financial loss, large carbon dioxide emission, and high risks of accidents [1]. Deployed in most of the existing traffic management systems, the traditional traffic light signal control strategies, e.g., fixed-time scheduling, where the traffic light signals are switched with fixed durations, and historical data-based scheduling, where the traffic light signals are switched according to pre-determined signal sequences, cannot adapt to the ever-changing traffic dynamics [2]. As we are heading towards the era of smart cities, artificial intelligence (AI) enabled intelligent transportation system (ITS) is emerging as a promising technique that is capable of achieving smart and flexible traffic control, thereby alleviating the long-lasting traffic congestion problem [3]–[5].

### B. Related Works

Building upon the demand-response mechanism, ITS improves the efficiency of transportation systems in terms of the average travel time by adopting adaptive traffic light signal control (ATSC). Specifically, according to the real-time traffic conditions, the decision system of ITS intelligently determines the next traffic light signal and its lasting time to ameliorate the traffic congestion. ATSC was initially realized by manually designing the traffic light signal plans [6]–[8]. However, due to the weak perception and the lack of learning ability, these manually designed schemes cannot react to real-time traffic patterns [9]. There were also some existing studies [10]–[14] focusing on optimization-based ATSC. In particular, the authors in [10] proposed a multi-resolution strategy to minimize the traffic delay by jointly optimizing the global traffic signal cycle and the local switching timing. The authors in [11] studied the joint optimization of traffic signal timing and driving trajectories via linear programming, where the traffic efficiency and energy efficiency are taken into account. The authors in [12] proposed a macroscopic traffic flow modeling approach for multi-intersection and multi-phase traffic signal systems, and developed a traffic signal switching and timing strategy for dwelling time minimization at intersections based on dynamic programming. The authors in [13] and [14] studied the traffic delay minimization problem for ATSC by utilizing tools from stochastic programming and linear-quadratic regulator theory, respectively. However, these optimization-based ATSC methods only considered simplified traffic models, and cannot be applied to achieve ATSC in large-scale traffic systems because of the

high computational complexity. Moreover, as the real-world traffic condition evolves in a complicated way with many uncertainties, e.g., driver's preference, weather, and road conditions, the traffic model considered by these model-based optimization methods can hardly capture these uncertainties accurately in practice [9].

Reinforcement learning (RL) [15], as a model-free method, can make decisions by learning from the observed data without relying on unrealistic assumptions on the model. By modeling the traffic light signal control at a single intersection as a Markov decision process (MDP), RL approaches have been applied to control the traffic light signals in [16], [17]. In particular, the authors in [16] established a Q-table to record the traffic state, traffic light signal, and the corresponding reward. By selecting traffic light signals according to the Q-table, they realized the ATSC in a relatively simplified scenario. As the size of the Q-table increases exponentially with the considered state space, the authors in [17] adopted a nonlinear function to approximate the expected action value under different traffic states. However, as the state space gets large, the difficulty in convergence makes it impractical to be adopted in the large-scale traffic system.

With the development of deep neural networks, by combining the deep learning technology with RL, deep reinforcement learning (DRL) [18], [19] has been recognized as a promising approach that is capable of finding an optimal or near-optimal policy through interacting with a complex environment. The application of DRL in achieving ATSC has been studied in [2], [20]–[24], where a single road intersection was considered. These studies cannot be directly extended to large-scale traffic networks as it is impractical to have a centralized agent to manage the whole traffic network. Thus, it is essential to view the traffic network with multiple intersections as a multi-agent system (MAS) and develop a decentralized scheme to control the traffic light signals in a distributed manner. In an MAS, the actions of all agents are typically coupled, rendering the design of collaboration strategies a main research problem [25]–[27]. Regarding the ATSC, many studies have recently been proposed to achieve collaboration among the neighboring intersections, e.g., [28]–[35]. In particular, the authors in [28] constructed a hierarchical structure consisting of a global agent and multiple regional agents. Based on the pre-determined actions generated by the regional agents, the authors proposed an algorithm to iteratively search for the solution that maximizes the global Q-value to realize the collaboration among all intersections. The authors in [29] proposed a multi-head attention mechanism to adjust the priority of cooperation with the neighboring intersections and built a graph attention neural network to optimize the traffic light signals. In [30], the authors proposed an actor-critic based method to coordinate the intersections under a partially observable environment. In [31], the authors proposed a hierarchical structure to alleviate the instability of the environment caused by the partially observable state. Besides, the authors in [32] took the waiting pedestrians into account and set different priorities for buses and ordinary vehicles. In [33], the authors proposed a region-aware cooperative strategy (RACS) based on graph attention network (GAT) and incorporated the spatial information of neighboring intersections. The

authors in [33] exploited the ubiquity of the IoV to accelerate traffic data collection. Despite the promising performance, the aforementioned algorithms relied on the heavy and real-time traffic data exchange among the intersections. Hence, these algorithms may not work well in practice as the communication links between the intersections are usually bandwidth-limited. In addition, these algorithms mainly adopt the hierarchical structure that relies on a centralized controller to realize the collaboration, and hence are not suitable for large-scale traffic networks.

### C. Contributions

In this paper, we aim to develop a communication-efficient decentralized ATSC framework for large-scale traffic networks consisting of multiple intersections, where the communication links among the neighboring intersections are bandwidth-limited. Developing such an framework is challenging due to the following reasons. First, since the effectiveness of ATSC at each intersection can be significantly affected by the traffic conditions at the neighboring intersections, each intersection is required to collaborate with other intersections to improve the traffic efficiency of the whole traffic network. Second, to achieve collaboration among neighboring intersections, each intersection needs to obtain real-time traffic data both at the local and the neighboring intersections, but only statistical information of traffic data can be shared between the intersections with bandwidth-limited communication links. Leveraging the traffic statistics to achieve collaboration among the neighboring intersections is a challenging task.

To this end, we develop a novel traffic light signal control framework, which consists of a generative adversarial network (GAN) based traffic data recovery algorithm and a multi-agent DRL based ATSC algorithm. Specifically, we solve the communication dilemma by applying the GAN technique to recovery the neighboring real-time traffic data with traffic statistics. As each intersection can only partially observe the environment in practice and determine the traffic light signals independently, we formulate the collaborative ATSC problem as a decentralized partially observable Markov decision process (Dec-POMDP) [36], and propose a decentralized multi-agent DRL based algorithm. The proposed framework is capable of achieving distributed training and distributed execution, as well as collaboration among the neighboring intersections. To the best of the authors' knowledge, this is the first work that proposes an efficient ATSC algorithm by utilizing both GAN and multi-agent DRL. The main contributions of this paper are summarized as follows.

- We propose a GAN based algorithm to recover the traffic data of the neighboring intersections. Specifically, by representing the traffic information with a matrix, we formulate the neighboring traffic data recovery problem as a data imputation problem. By exploiting the strong spatial correlation of the traffic flows among the neighboring intersections, we elaborately design loss functions for the generator and discriminator networks.

TABLE I  
DEFINITIONS OF FREQUENTLY USED NOTATIONS

Notation	Definition
$\Phi_i$	Set of neighboring intersections of intersection $i$
$\Phi_i^\cup$	Union of $\Phi_i$ and intersection $i$
$o_i(t)$	Observed local traffic data of agent $i$ in time slot $t$
$k_i$	Traffic statistics received from the neighboring intersections of intersection $i$
$\tilde{o}_j$	Estimated traffic data at intersections $j$
$\mathbf{G}_i$	Recovered traffic data matrix at intersections $i$
$\mathcal{G}(\cdot)$	Generator function of the GAN network
$\mathcal{D}(\cdot)$	Discriminator function of the GAN network
$\tilde{\mathbf{X}}_i$	Traffic data matrix for intersection $i$ and its neighboring intersections in set $\Phi_i$
$\Psi_i$	Index set of all lanes at intersection $i$
$y_{k,l}(t)$	Number of vehicles in the $k$ -th segment of the $l$ -th lane at the beginning of time slot $t$
$K$	Number of equal segments on each lane
$\theta_i^{\mathcal{G}}$	Parameters of the generator network at intersection $i$
$\theta_i^{\mathcal{D}}$	Parameters of the discriminator network at intersection $i$
$\theta_i^{\text{est}}$	Parameters of the estimator network of agent $i$
$\theta_i^{\text{tar}}$	Parameters of the target network of agent $i$
$N_b$	Size of the mini-batch

- We propose a novel decentralized multi-agent DRL based algorithm to control the traffic light signal at each intersection. We tackle the coupling between agents by introducing a regional reward. By exploiting the value decomposition technique, the proposed framework can maximize the joint state-action value by maximizing the local state-action value based on its local traffic data. With a combination of GAN and multi-agent DRL, the proposed framework is capable of achieving collaborative traffic signal control among the neighboring intersections in a scalable manner.
- We conduct extensive simulations to illustrate the superior performance of the proposed decentralized ATSC framework for large-scale traffic networks. In addition to significant reduction in the travel time and the accumulated waiting time, the proposed algorithm can also maintain stable traffic throughput under dynamic traffic flows. Moreover, we demonstrate that our proposed algorithm can coordinate multiple intersections such that the “greenwave” phenomenon is observed.

The rest of this paper is organized as follows. In Section II, we describe the system model and the problem formulation. In Section III, we put forward a GAN-based algorithm for neighboring traffic data recovery. In Section IV, we present the details of the proposed algorithm for ATSC. Simulation results are provided in Section V. Finally, Section VI concludes this work. Table I lists the frequently used notations.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

### A. System Model

Consider a traffic network in an urban area consisting of  $N$  signalized intersections, as shown in Fig. 1. We denote the index set of intersections as  $\mathcal{N} = \{1, 2, \dots, N\}$ . We consider typical four-way intersections across the network, where each intersection has exactly four adjacent intersections that lie to the north, west, south, and east, respectively. Two adjacent



Fig. 1. Illustration of a typical city traffic network consisting of multiple signalized intersections.

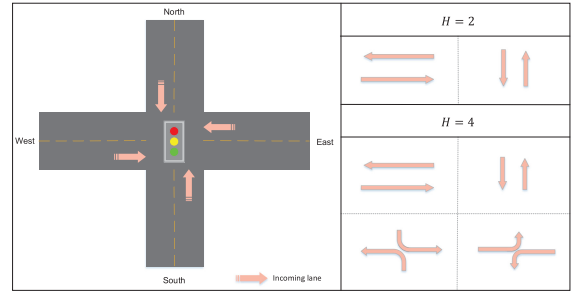


Fig. 2. Illustration of available traffic light signals at each intersection for different values of  $H$ .

intersections are connected via a two-way road, which contains two one-way lanes in the opposite direction. The traffic light signal at each intersection is controlled by an intelligent agent according to the real-time traffic. Hence, we use intersection  $i \in \mathcal{N}$  and agent  $i \in \mathcal{N}$  interchangeably hereafter. With the assistance of the nearby roadside sensors (e.g., cameras, speed radars), we assume that each agent  $i \in \mathcal{N}$  is capable of obtaining the real-time local traffic information (e.g., number, velocities, and positions of vehicles) at intersection  $i$ . We consider a practical yet challenging scenario, where each agent can only share the statistics of its local traffic information (i.e., number of vehicles and traffic light signal) with its neighboring intersections via bandwidth-limited communication links. We define the set of neighboring intersections of intersection  $i$  as

$$\Phi_i = \{j \in \mathcal{N}, j \neq i \mid \text{dis}(i, j) \leq d\}, \quad \forall i \in \mathcal{N}, \quad (1)$$

where  $\text{dis}(i, j)$  denotes the Manhattan distance between the  $i$ -th and the  $j$ -th intersections, and  $d$  is a predetermined threshold.

Each intersection has  $H$  legal traffic light signals, as shown in Fig. 2. For example, the value of  $H$  can be set to be 2 if we only consider the straight driving, and 4 if we consider both straight driving and right turning. Time is divided into slots with constant durations. The traffic light signal remains invariant within one time slot. We denote the traffic light signal at intersection  $i$  in time slot  $t$  as  $\rho_i(t)$ , which is determined at the beginning of time slot  $t$  by agent  $i$  according to the locally available traffic information. The determined traffic light signals



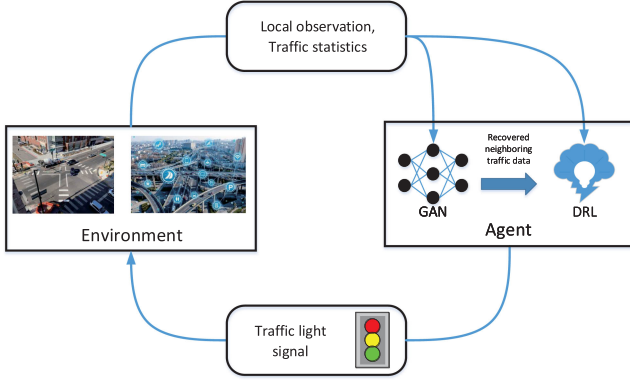


Fig. 3. Overall structure of the proposed framework. Each agent takes an observation of the real-time local traffic and receives traffic statistics from its neighboring intersections. Based on the local state extracted from the local observation, each agent determines its local traffic light signal for real-time traffic management. On the other hand, each agent applies GAN to recover the neighboring traffic data based on the local observation and the neighboring traffic statistics, and stores it for further collaborative policy updates.

$\{\rho_i(t)\}_{i \in \mathcal{N}}$  guide all vehicles from non-conflicting directions in time slot  $t$ . We assume that the time spent on both traffic condition observation and traffic light signal determination is negligible when compared to the time duration of one time slot.

### B. Problem Formulation

Our objective is to maximize the average velocity of the traffic network by dynamically adjusting the traffic light signal of each intersection for arbitrary traffic flows in a distributed and collaborative manner. This is a challenging task due to the following reasons. First, due to the complexity of traffic environment, the average velocity of the traffic network in terms of  $\{\rho_i(t)\}_{i \in \mathcal{N}}$  is very challenging, if not impossible, to be analytically characterized. Second, due to the continuity of the traffic flows along the roads, the decisions on the traffic light signals among the neighboring intersections are coupled. As a result, it is necessary for each agent to coordinate with its neighboring intersections to improve the traffic efficiency of the whole traffic network, which is proverbially challenging. Third, achieving collaboration among the neighboring intersections generally requires accurate real-time traffic information of these intersections. However, with bandwidth-limited communication links, only the traffic statistics can be shared between the neighboring intersections.

To overcome the aforementioned challenges, we resort to developing a novel traffic light signal control framework by exploiting tools from both GAN and multi-agent DRL. The overall structure of the proposed framework is shown in Fig. 3. The developed framework is capable of achieving both decentralized training and decentralized execution, and consists of the following two phases. First, we propose a GAN-based algorithm to recover the neighboring traffic data at each intersection  $i \in \mathcal{N}$  based on the local observation at intersection  $i$  and the received traffic statistics from neighboring intersections  $j \in \Phi_i$  to facilitate the design of collaborative ATSC. Second, with the recovered traffic data, we develop a low-complexity

multi-agent DRL based ATSC algorithm for each agent to make a local control decision, which inherently achieves effective collaboration among the neighboring intersections. The main ideas of the proposed traffic data recovery and traffic light signal control algorithms are summarized as follows.

1) *Neighboring Traffic Data Recovery*: By exploiting the spatial correlation of the traffic data between adjacent intersections [37], the traffic data recovery problem can be regarded as a data imputation problem, where the traffic data at the neighboring intersections are unobserved and regarded as the missing data that need to be imputed. For agent  $i$ , we denote the observed local traffic data in time slot  $t$  by  $o_i(t)$ , and the traffic statistics received from the neighboring intersections as  $k_i(t)$ , which will be defined in Section III-A. Note that the traffic data of the neighboring intersections in time slot  $t$ , i.e.,  $\{o_j(t)\}_{j \in \Phi_i}$ , are unobserved at intersection  $i$ . If the conditional distribution  $\mathbb{P}(\tilde{o}_j(t), j \in \Phi_i \mid o_i(t), k_i(t))$  is available, then agent  $i$  can impute the unobserved traffic data, where  $\{\tilde{o}_j(t)\}_{j \in \Phi_i}$  denote the estimated traffic data of the neighboring intersections. As the traffic data that need to be recovered are not always sparse, the compressed sensing (CS) technique is not applicable, rendering the recovery of the neighboring traffic data a challenging task. As GAN is capable of generating data by mimicking the distribution of the real data [38], we propose to use GAN to generate the unobserved neighboring traffic data by exploiting the strong spatial correlation of the traffic data. Specifically, we shall develop and train a generator of GAN to approximate the conditional distribution  $\mathbb{P}(\tilde{o}_j(t), j \in \Phi_i \mid o_i(t), k_i(t))$  based on the historical data. With the well-trained generator, each agent is able to recover the traffic data of its neighboring intersections. The details of the proposed GAN-based traffic data recovery method will be presented in Section III.

2) *Coordinated Decision Making Using Recovered Data*: Due to the coupling among the decisions of the neighboring agents, the continuity of the traffic flows, as well as the local availability of the traffic data, the traffic light signal control for the traffic network with multiple intersections can be regarded as a Dec-POMDP [36]. Specifically, at the beginning of time slot  $t$ , agent  $i$  takes an observation of its local traffic condition and obtains the local state, which is utilized to help agent  $i$  determine its traffic light signal in time slot  $t$ . After setting the traffic light signal at each intersection, the traffic environment (e.g., the positions and velocities of vehicles) transit to a new state according to the probability transition matrix. At the end of time slot  $t$ , agent  $i$  receives a reward from the traffic environment. Agent  $i \in \mathcal{N}$  interacts with the environment and aims to find a policy that maximizes the long-term cumulative reward. The cumulative reward takes into account the average velocity of all vehicles at intersection  $i$  and its neighboring intersections. For notational ease, we abbreviate  $\Phi_i \cup \{i\}$  as  $\Phi_i^U$ . To achieve collaboration among agents in set  $\Phi_i^U$ , the recovered traffic data are exploited for policy updating. The details of the DRL-based coordinated decision making method will be presented in Section IV.

*Remark 1*: Note that there exists a trade-off between the communication overhead among neighboring intersections and the traffic efficiency. Specifically, considering a larger set of neighboring intersections enables a larger-scale collaboration,

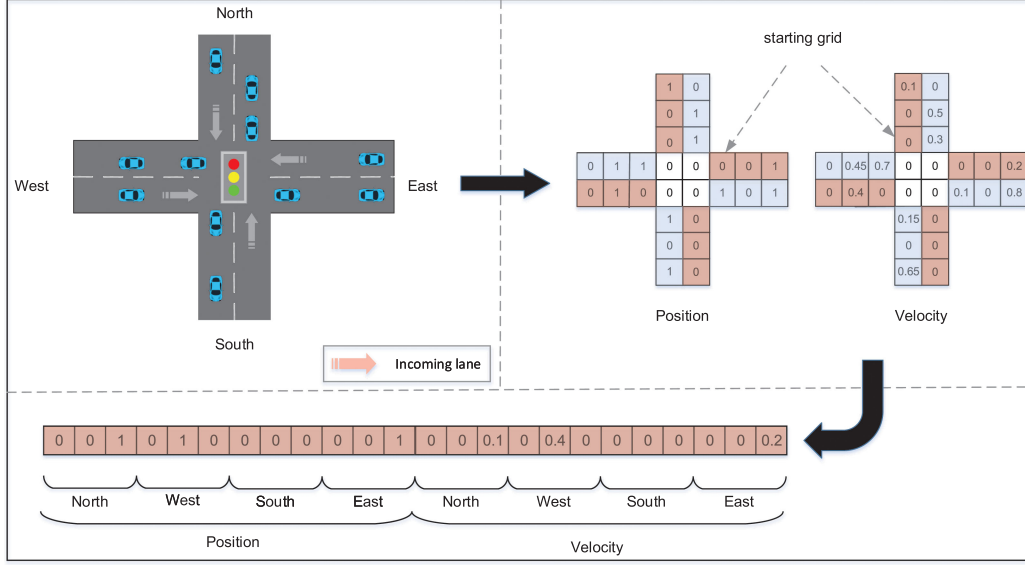


Fig. 4. Illustration of traffic data representation at a typical intersection.

leading to higher traffic efficiency but also incurring more communication overhead, and vice versa. To balance the trade-off between communication overhead and traffic efficiency, we consider a general scenario, where the collaboration level among neighboring intersections can be adjusted by appropriately setting the value of parameter  $d$ , defined in (1). In particular, when the value of  $d$  is large enough, our problem turns to be a global collaboration problem, where all agents collaborate with each other. When  $d$  is zero, our problem is reduced to a multiple independent traffic light signals control problem, where no collaboration among the agents is considered.

### III. GAN-BASED TRAFFIC DATA RECOVERY

In this section, we propose a GAN-based traffic data recovery algorithm, which exploits the strong spatial correlation of the traffic flows among the neighboring intersections.

#### A. Traffic Data Representation

To facilitate the traffic data recovery, we represent the traffic information of the whole network with a matrix, which can be constructed based on the position and the speed information of all vehicles at each intersection. Recall that each agent is only able to observe part of the traffic information with the help of the roadside sensors (e.g., speed radars, cameras) at the beginning of each time slot. We divide the traffic network into grids, where each grid is assumed to accommodate at most one vehicle. For sake of clarity, we take the grids at intersection  $i$  as an example. To avoid information redundancy, only the traffic data of the incoming lanes are considered. The four incoming lanes of intersection  $i$  are arranged in the order of {north, west, south, east}, which are indexed by  $l \in \{1, 2, 3, 4\}$ , respectively. Each incoming lane is divided into  $L$  grids, each of which is indexed by  $j \in \{1, \dots, L\}$ . Note that we count the grids from the center of the intersection to the end of the lane, i.e., the starting grid of each lane is the grid closest to the center

of the intersection, as shown in Fig. 4. The traffic information of the  $(j + (l - 1)L)$ -th grid at intersection  $i$  can be expressed as

$$e_{i,j+(l-1)L}(t) = (p_{i,j+(l-1)L}(t), v_{i,j+(l-1)L}(t)),$$

where  $p_{i,j+(l-1)L}(t) \in \{0, 1\}$  indicates the existence of a vehicle and  $v_{i,j+(l-1)L}(t) \in [0, 1]$  denotes the normalized speed of the corresponding vehicle.

After each grid is well defined, the traffic data at intersection  $i$  can be converted into a vector, denoted as  $o_i(t) \in \mathbb{R}^{1 \times 8L}$ , where the first and second halves of vector  $o_i$  represent the position and velocity information, respectively. Fig. 4 shows a simple example illustrating how the real-time traffic data at one intersection are represented by a vector. Therefore, the traffic data of intersection  $i$  and its neighboring intersections in set  $\Phi_i$  can be expressed by matrix  $\mathbf{O}_i(t) \in \mathbb{R}^{C \times 8L}$ , where the rows are composed by  $\{o_j(t)\}_{j \in \Phi_i^U}$ , and  $C$  is the cardinality of set  $\Phi_i^U$ .

Due to the bandwidth-limited communication links, each agent exchanges the statistical traffic information with its neighboring agents to assist the traffic data recovery. We denote the traffic statistics  $\mathbf{K}(t) \in \mathbb{R}^{C \times 6}$  of all intersections in set  $\Phi_i^U$  by

$$\mathbf{K}(t) = (\mathbf{K}^N(t), \mathbf{k}^L(t), \mathbf{k}^V(t)) = (\mathbf{k}_1^T(t), \dots, \mathbf{k}_C^T(t))^T, \quad (2)$$

where  $\mathbf{K}^N(t) \in \mathbb{R}^{C \times 4}$  denotes the number of vehicles on the four incoming lanes at all  $C$  intersections in set  $\Phi_i^U$ ,  $\mathbf{k}^L(t) \in \mathbb{R}^{C \times 1}$  denotes the traffic light signals at all  $C$  intersections in set  $\Phi_i^U$ , and  $\mathbf{k}^V(t) \in \mathbb{R}^{C \times 1}$  denotes the average velocity of vehicles on the incoming lanes at all  $C$  intersections in set  $\Phi_i^U$ . Vector  $\mathbf{k}_i(t) \in \mathbb{R}^{1 \times 6}$  denotes the  $i$ -th row vector of matrix  $\mathbf{K}(t)$ , which contains the traffic statistics from intersection  $i$ . The first four elements of  $\mathbf{k}_i(t)$  represent the number of vehicles in the four incoming lanes at intersection  $i$  and they are arranged in the order of {north, west, south, east}. The fifth element of  $\mathbf{k}_i(t)$  represents the traffic light signal at intersection  $i$ , while the sixth element represents the average speed of all vehicles at intersection  $i$ .

With all notations in place, we define the traffic data matrix (TDM) for intersection  $i$  and its neighboring intersections in set  $\Phi_i$  as

$$\tilde{\mathbf{X}}_i(t) = (\mathbf{O}_i(t), \mathbf{K}^N(t), \mathbf{k}^L(t)) \in \mathbb{R}^{C \times (8L+5)}. \quad (3)$$

Each agent  $i \in \mathcal{N}$  only has access to its local traffic data (i.e.,  $\mathbf{o}_i(t)$ ) and the received traffic statistics (i.e.,  $\{\mathbf{k}_i(t)\}_{i \in \Phi_i^U}$ ), while the rest of matrix  $\tilde{\mathbf{X}}_i(t)$  is unobserved and needs to be recovered.

### B. TDM Recovery With GAN

In this subsection, we develop a GAN-based algorithm to achieve TDM recovery by utilizing the historical data. The proposed GAN consists of two neural networks: a generator  $\mathcal{G}$  designed to generate the unobserved traffic data based on the observed local traffic data and the received traffic statistics, and a discriminator  $\mathcal{D}$  designed to judge the authenticity of the generated data. Note that the historical data contains both the local traffic data and the received traffic statistics from all neighboring intersections. As different agents have different knowledge on the traffic data, we define a specialized mask matrix  $\mathbf{M} \in \mathbb{R}^{C \times (8L+5)}$  for each agent to cover the unobserved traffic data. In particular, we use  $\mathbf{M}^{[i]}$  to denote the mask matrix of intersection  $i$ , i.e.,

$$\mathbf{M}_{m,n}^{[i]} = \begin{cases} 0, & m \neq i \text{ and } 1 \leq n \leq 8L, \\ 1, & \text{otherwise.} \end{cases}$$

Therefore, the available traffic data at intersection  $i$  can be expressed as  $\mathbf{M}^{[i]} \odot \tilde{\mathbf{X}}_i(t)$ , where  $\odot$  denotes the element-wise product. For the unobserved part of  $\tilde{\mathbf{X}}_i(t)$ , we fill each entry with a standard random Gaussian variable. Thus, the available traffic data at agent  $i$  are represented as

$$\hat{\mathbf{X}}_i(t) = \mathbf{M}^{[i]} \odot \tilde{\mathbf{X}}_i(t) + (\mathbf{1} - \mathbf{M}^{[i]}) \odot \mathbf{Z},$$

where  $\mathbf{Z}$  denotes a stand random Gaussian matrix that has the same size as matrix  $\mathbf{M}^{[i]}$ . The generator network  $\mathcal{G}$  takes the available traffic data matrix  $\hat{\mathbf{X}}_i(t)$  as the input and generates an estimated TDM based on its current learned conditional distribution, denoted as  $\mathcal{G}(\hat{\mathbf{X}}_i(t))$ , where  $\mathcal{G}(\cdot)$  denotes the generator function. As defined in (3), each row of  $\mathcal{G}(\hat{\mathbf{X}}_i(t))$  includes the estimated traffic observation of neighboring intersections, i.e.,  $\tilde{o}_j$ ,  $\forall j \neq i, j \in \Phi_i^U$ . Along with the locally observed traffic data, the recovered TDM is given by

$$\mathbf{G}_i(t) = \mathbf{M}^{[i]} \odot \tilde{\mathbf{X}}_i(t) + (\mathbf{1} - \mathbf{M}^{[i]}) \odot \mathcal{G}(\hat{\mathbf{X}}_i(t)). \quad (4)$$

The discriminator network  $\mathcal{D}$  takes  $\mathbf{G}_i(t)$  as the input and outputs a probability matrix, denoted as  $\mathbf{\Lambda}_i(t) = \mathcal{D}(\mathbf{G}_i(t)) \in \mathbb{R}^{C \times (8L+5)}$ , the elements of which are the probabilities that the corresponding elements in  $\mathbf{G}_i(t)$  are considered to be true by discriminator  $\mathcal{D}$ .

In order to recover TDM as accurately as possible, we design the loss function by taking into account the difference between the real and recovered positions of each vehicle, as well as the difference between the real and recovered numbers of vehicles on each lane. By denoting the parameters of the generator network  $\mathcal{G}$  at intersection  $i$  as  $\theta_i^G$ , we adopt the following loss

### Algorithm 1: Proposed GAN-Based Traffic Data Recovery Algorithm.

**Input:** Historical data  $\mathcal{H}$ , mini-batch size  $N_b$ , hyperparameter  $\lambda$ , learning rate  $\mu$ .

- 1 Randomly initialize parameter  $\theta_i^G$  of the generator network and parameter  $\theta_i^D$  of the discriminator network.
- 2 **for** each agent  $i \in \mathcal{N}$  **do**
- 3     **while** generator network is not converged **do**
- 4         Draw  $N_b$  samples  $\{\tilde{\mathbf{X}}_i(t)\}^{N_b}$  from  $\mathcal{H}$ , randomly generate  $N_b$  mask matrices  $\{\mathbf{M}^{[i]}\}^{N_b}$  and  $N_b$  uniform noise matrices  $\{\mathbf{Z}\}^{N_b}$ .
- 5         Generator  $\mathcal{G}$  at intersection  $i$  outputs the TDM  $\mathcal{G}(\hat{\mathbf{X}}_i(t))$  and obtains the recovered TDM according to (4).
- 6         Discriminator  $\mathcal{D}$  at intersection  $i$  takes  $\mathcal{G}(\hat{\mathbf{X}}_i(t))$  as input and outputs the probability matrix  $\mathbf{\Lambda}_i(t)$ .
- 7         Calculate the loss of  $\mathcal{D}$  according to (7).
- 8         Given  $\theta_i^G$ , update  $\theta_i^D$  using Adam
 
$$\theta_i^D \leftarrow \theta_i^D - \mu \frac{1}{N_b} \sum_{k=1}^{N_b} \nabla(\mathcal{L}_D).$$
- 9         Calculate the loss of  $\mathcal{G}$  according to (5).
- 10         Given  $\theta_i^D$ , update  $\theta_i^G$  using Adam
 
$$\theta_i^G \leftarrow \theta_i^G - \mu \nabla_{\mathcal{G}}(\mathcal{L}_{\text{Elem}} + \lambda \mathcal{L}_{\text{Stat}}).$$

function to train the generator network  $\mathcal{G}$

$$\mathcal{L}_G(\theta_i^G) = \mathcal{L}_{\text{Elem}} + \lambda \mathcal{L}_{\text{Stat}}, \quad (5)$$

where  $\mathcal{L}_{\text{Elem}}$  represents the level in deceiving the discriminator  $\mathcal{D}$ ,  $\mathcal{L}_{\text{Stat}}$  represents the mean squared error (MSE) between the real and estimated traffic statistics,  $\lambda$  denotes a hyperparameter to balance the trade-off between  $\mathcal{L}_{\text{Elem}}$  and  $\mathcal{L}_{\text{Stat}}$ ,  $\mathbf{1} \in \mathbb{R}^{C \times (8L+5)}$  is an all-ones matrix, and

$$\begin{aligned} \mathcal{L}_{\text{Elem}}(\theta_i^G) &= -(\mathbf{1} - \mathbf{M}^{[i]}) \log(\mathbf{\Lambda}_i(t)), \quad \forall i \in \mathcal{N}, \\ \mathcal{L}_{\text{Stat}}(\theta_i^G) &= \sum_{n=1}^C \sum_{m=1}^4 \left( \tilde{\mathbf{X}}_i(t)_{n,m+8L} - \sum_{j=(m-1)L+1}^{mL} \mathcal{G}(\hat{\mathbf{X}}_i(t))_{n,j} \right)^2. \end{aligned} \quad (6)$$

Likewise, we denote the parameters of the discriminator network  $\mathcal{D}$  as  $\theta_i^D$  and train the discriminator network  $\mathcal{D}$  with the following loss function

$$\mathcal{L}_D = \mathbf{M}^{[i]} \log(\mathbf{\Lambda}_i(t)) + (\mathbf{1} - \mathbf{M}^{[i]}) \log(\mathbf{1} - \mathbf{\Lambda}_i(t)). \quad (7)$$

By iteratively calculating (5) and (7), the generator network  $\mathcal{G}$  and the discriminator network  $\mathcal{D}$  converge to a balance state [39]. Based on the aforementioned discussions, we summarize the proposed GAN-based traffic data recovery algorithm in Algorithm 10.

#### IV. MULTI-AGENT DRL-BASED ATSC

In this section, we propose a novel traffic light signal control algorithm named GAN-aided decentralized ATSC (GD-ATSC), which is capable of achieving collaboration among the neighboring intersections in a distributed and scalable manner. First, we model the traffic light signal control problem for traffic networks with multiple intersections as a Dec-POMDP. We then present the synergy mechanism based on the recovered traffic data. Finally, we describe the training and updating strategies of the proposed algorithm.

##### A. MDP Modeling

As the traffic light signal control of the considered traffic network with  $N$  intersections is essentially a multi-agent decision-making process and each agent only has access to its local traffic observation and the received traffic statistics, we model it as a Dec-POMDP, which can be defined by tuple  $G = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \mathcal{R}, \mathcal{O}, \mathcal{N}, \gamma \rangle$ . The details of the Dec-POMDP are described as follows.

- **Joint State Space  $\mathcal{S}$ :** The joint state is denoted as  $\mathbf{s}(t) = (s_1(t), \dots, s_N(t))$ , where  $s_i(t)$  denotes the state of intersection  $i$  at the beginning of time slot  $t$ . We define the local state as

$$s_i(t) = (\{y_{k,l}(t) \mid k \in \{1, \dots, K\}, l \in \Psi_i\}, \rho_i(t)),$$

where  $\Psi_i$  denotes the index set of all lanes at intersection  $i$ ,  $y_{k,l}(t)$  denotes the number of vehicles in the  $k$ -th segment of the  $l$ -th lane at the beginning of time slot  $t$ , and  $K \in \{1, \dots, L\}$  denotes the number of equal segments on each lane. Particularly, when  $K = 1$ ,  $y_{k,l}(t)$  represents the number of vehicles on the  $l$ -th lane. When  $K = L$ ,  $y_{k,l}(t)$  indicates the existence of vehicle in the  $k$ -th grid on the  $l$ -th lane. At the beginning of time slot  $t$ , agent  $i \in \mathcal{N}$  takes an observation  $o_i(t)$  via the deployed roadside sensors (e.g., speed radars, cameras) to obtain the current traffic conditions, which include the position of every vehicle at intersection  $i$ . Thus,  $s_i(t)$  is available at agent  $i$  at the beginning of time slot  $t$ .

- **Joint Action Space  $\mathcal{A}$ :** Agent  $i$  determines its local traffic light signal based on its local state  $s_i(t)$ . We denote the action space of agent  $i$  as  $\mathcal{A}_i$  and  $a_i(t) \in \mathcal{A}_i$  as the action of agent  $i$  in time slot  $t$ . In particular, we define the action of agent  $i$  as the traffic light signal of intersection  $i$ , i.e.,  $a_i(t) = \rho_i(t)$ . Thus, the action space can be expressed as

$$\mathcal{A}_i = \{1, \dots, H\}.$$

Note that the action spaces of all agents are the same, i.e.,  $\mathcal{A}_i = \mathcal{A}_j, \forall i \neq j, i, j \in \mathcal{N}$ . We define the joint action space as  $\mathcal{A} = \prod_{i=1}^N \mathcal{A}_i$ , where  $\prod$  denotes the Cartesian product of the sets.

- **Probability Transition Matrix  $\mathbf{P}$ :** After determining the traffic light signal at each intersection, the traffic environment transits to a new state according to the probability transition matrix  $\mathbf{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ . Note that the state transition probability  $\mathbb{P}(\mathbf{s}(t+1) \mid \mathbf{s}(t), \mathbf{a}(t))$  denotes the probability that given the current global state  $\mathbf{s}(t)$

and global action  $\mathbf{a}(t) = (a_1(t), \dots, a_N(t))$ , the environment moves from state  $\mathbf{s}(t)$  to state  $\mathbf{s}(t+1)$ . Under the considered system, the randomness of the state transition is due to the variation of traffic flows based on the current state  $\mathbf{s}(t)$  and global action  $\mathbf{a}(t)$ . The state transition probability matrix  $\mathbf{P}$  reflects the dynamics of the traffic environment.

- **Reward Function  $\mathcal{R}$ :** At the end of time slot  $t$ , agent  $i$  receives a reward denoted as  $r_i(t)$  from the environment according to the reward function  $\mathcal{R}$ . As the traffic light signals of the neighboring intersections (i.e.,  $\{\rho_j(t)\}_{j \in \Phi_i}$ ) have a significant impact on the traffic conditions at intersection  $i$  [29], we consider a regional reward, which is contributed by all agents in set  $\Phi_i^U$  and also reflects the level of collaboration among the agents. We define the regional reward of intersection  $i$  as the average velocity of vehicles at all intersections in set  $\Phi_i^U$ , given by

$$r_i(t) = \frac{2}{C} \sum_{j \in \Phi_i^U} \frac{\bar{v}_j(t)}{\bar{v}_j} - 1, \forall i \in \mathcal{N}, \quad (8)$$

where  $\bar{v}_j(t)$  denotes the average velocity of all vehicles on the incoming lanes of intersection  $j$  at the end of time slot  $t$ , and  $\bar{v}_j > 0$  denotes the maximum allowed speed of the vehicles on the incoming lanes of intersection  $j$ . In (8), we normalize the reward into the range of  $[-1, 1]$  for accelerating the policy convergence. We can obtain  $r_i(t)$  based on the received traffic statistics  $\{\mathbf{k}_j(t)\}_{j \in \Phi_i}$  and the local velocity information (i.e.,  $\{v_{i,l}(t)\}_{l \in \Psi_i}$ ).

- **Joint Observation Space  $\mathcal{O}$ :** The local observation  $o_i(t)$  taken by agent  $i$  is defined in Section III-A. We define  $\mathcal{O}_i$  as the set of all possible local observations at intersection  $i$ . Similarly, the joint observation space is defined as  $\mathcal{O} = \prod_{i=1}^N \mathcal{O}_i$ .
- **Discount Factor  $\gamma$ :** The objective of each agent is to maximize its long-term cumulative reward, while the discount factor  $\gamma \in [0, 1]$  is applied to determine the weight on the potential future reward. Setting a larger value of  $\gamma$  enables each agent to pay more attention to the potential future reward.

##### B. Value Decomposition Technique

With the specifically designed state, action, and reward in place, we optimize each agent's policy to maximize the expected cumulative reward, which is defined as follows.

$$\max_{\mathbf{a}_i^R(t)} \mathbb{E}_{\{\pi_j\}_{j \in \Phi_i^U}} \left[ \sum_{k=0}^{\infty} \gamma^k r_i(t+k) \mid \{\tilde{s}_j(t)\}_{j \in \Phi_i}, s_i(t), \mathbf{a}_i^R(t) \right], \quad (9)$$

where  $\mathbf{a}_i^R(t) = \{a_j\}_{j \in \Phi_i^U}$  denotes the regional joint action taken by the agents in set  $\Phi_i^U$ , and  $\{\tilde{s}_j(t)\}_{j \in \Phi_i}$  denotes the regional joint state extracted from the recovered neighboring traffic data. We denote the regional joint state of agents in set  $\Phi_i^U$  as  $\mathbf{s}_i^R(t) = \{\tilde{s}_j(t)\}_{j \in \Phi_i} \cup \{s_i(t)\}$ . As each agent (e.g.,  $i \in \mathcal{N}$ ) maximizes the average velocity of all vehicles on the incoming lanes of its own intersection and its neighboring intersections (i.e., the incoming lanes of intersection  $j \in \Phi_i^U$ ), the average velocity of all vehicles in the traffic network can be improved.



We denote the expectation in (9) as the regional joint state-action value  $Q^R(s_i^R(t), a_i^R(t))$ , given by

$$Q^R(s_i^R(t), a_i^R(t)) = \mathbb{E}_{\{\pi_j\}_{j \in \Phi_i^U}} \left[ \sum_{k=0}^{\infty} \gamma^k r_i(t+k) | \{\tilde{s}_j(t)\}_{j \in \Phi_i}, s_i(t), a_i^R(t) \right]. \quad (10)$$

In order to estimate the regional joint state-action value  $Q^R(s_i^R(t), a_i^R(t))$ , it is necessary for agent  $i$  to know the regional joint state  $s_i^R(t)$  and the regional joint action  $a_i^R(t)$ . However, at the beginning of time slot  $t$ , each agent only has access to its local traffic data, and the traffic light signal at each intersection is yet to be decided. Therefore, it is impractical for each agent to obtain the regional joint state-action value. To address this issue, we propose to use the value decomposition technique to factorize the regional joint state-action value into multiple local state-action values, which enables each agent to determine its local traffic light signal according to its local state  $s_i(t)$  only, thereby facilitating the distributed design of the GD-ATSC algorithm.

We adopt a multi-layer perceptron (MLP) to approximate the policy that determines the traffic light signal of each agent. For different traffic light signals  $\{\rho_i(t)\}$ , the MLP outputs different values to estimate the corresponding contribution of its own action on the regional joint state-action value. Specifically, for action  $a_i(t) \in \mathcal{A}_i$ , agent  $i \in \Phi_i^U$  calculates a local state-action value, denoted by  $\tilde{Q}_i(s_i(t), a_i(t))$ , based on its MLP. To establish the relationship between the local state-action value estimated by the MLP of each agent and the regional joint state-action value, we use a mapping function  $f(\cdot)$  in the value decomposition networks (VDNs) architecture [40], [41] as follows.

$$Q^R(s_i^R(t), a_i^R(t)) = f\left(\tilde{Q}_1(\tilde{s}_1(t), a_1(t)), \dots, \tilde{Q}_i(s_i(t), a_i(t)), \dots, \tilde{Q}_C(\tilde{s}_C(t), a_C(t))\right). \quad (11)$$

Following the principle of monotonicity in [41], under a given state, the function in (11) should satisfy the following condition

$$\frac{\partial Q^R(s_i^R(t), a_i^R(t))}{\partial \tilde{Q}_i(s_i(t), a_i(t))} \geq 0, \quad \forall i \in \Phi_i^U. \quad (12)$$

Any mapping function  $f(\cdot)$  that satisfies the condition given in (12) leads to the following lemma.

**Lemma 1:** With the optimal regional joint action, denoted as  $a_i^{R*}(t) \in \prod_{j \in \Phi_i^U} \mathcal{A}_j$ , under the regional joint state  $s_i^R(t)$  for problem (9), we have

$$Q^R(s_i^R(t), a_i^{R*}(t)) = Q^R\left(s_i^R(t), \arg \max_{a_1(t)} \tilde{Q}_1(\tilde{s}_1(t), a_1(t)), \dots, \arg \max_{a_C(t)} \tilde{Q}_C(\tilde{s}_C(t), a_C(t))\right). \quad (13)$$

*Proof:* By denoting  $a_i^{R*}(t)$  as the optimal regional joint action under the regional joint state  $s_i^R(t)$ , we have

$$Q^R(s_i^R(t), a_i^{R*}(t)) \geq Q^R(s_i^R(t), a_i^R(t)),$$

$$\forall a_i^R(t) \in \prod_{j \in \Phi_i^U} \mathcal{A}_j.$$

Since  $\frac{\partial Q^R(s_i^R(t), a_i^R(t))}{\partial \tilde{Q}_i(s_i(t), a_i(t))} \geq 0$ , we have the following inequality,

$$\begin{aligned} Q^R(s_i^R(t), a_i^R(t)) &= f(\tilde{Q}_1(\tilde{s}_1(t), a_1(t)), \dots, \tilde{Q}_i(s_i(t), a_i(t)), \dots, \tilde{Q}_C(\tilde{s}_C(t), a_C(t))) \\ &\leq f(\tilde{Q}_1(\tilde{s}_1(t), a_1(t)), \dots, \tilde{Q}_i(s_i(t), a_i^*(t)), \dots, \tilde{Q}_C(\tilde{s}_C(t), a_C(t))) \\ &= Q^R(s_i^R(t), a_1(t), \dots, a_{i-1}(t), a_i^*(t), a_{i+1}(t), \dots, a_C(t)). \end{aligned} \quad (14)$$

where  $a_i^*(t) = \arg \max_{a_i(t) \in \mathcal{A}_i} \tilde{Q}_i(s_i(t), a_i(t))$ . Similarly, we have the inequality in (15).

$$\begin{aligned} Q^R(s_i^R(t), a_1(t), \dots, a_j(t), \dots, a_C(t)) &\leq Q^R\left(s_i^R(t), \arg \max_{a_1(t) \in \mathcal{A}_1} \tilde{Q}_1(\tilde{s}_1(t), a_1(t)), \dots, \arg \max_{a_i(t) \in \mathcal{A}_i} \tilde{Q}_i(s_i(t), a_i(t)), \dots, \arg \max_{a_C(t) \in \mathcal{A}_C} \tilde{Q}_C(\tilde{s}_C(t), a_C(t))\right). \end{aligned} \quad (15)$$

As a result, we obtain (13).

According to Lemma 1, the maximal regional joint state-action value can be achieved by maximizing each local state-action value. In other words, the regional traffic efficiency can be maximized if each agent individually adjusts its traffic light signal to maximize the local state-action value  $\tilde{Q}_i(s_i(t), a_i(t))$ , thereby enhancing the traffic efficiency of the whole traffic network. Hence, agent  $i \in \mathcal{N}$  determines its action based on the following rule

$$a_i^*(t) = \arg \max_{a_i(t) \in \mathcal{A}_i} \tilde{Q}_i(s_i(t), a_i(t)). \quad (16)$$

To obtain  $a_i^*(t)$  in (16), agent  $i$  feeds the local state  $s_i(t)$  into its MLP, which outputs the action that corresponds to the largest local state-action value. Note that the selection of the local action  $a_i(t)$  does not require the states and actions of other agents. Such a design enables each agent to make its own decision based on the local observation, while achieving collaboration with its neighboring intersections in a distributed manner.

### C. Proposed Structure, Training and Algorithm

In this subsection, we present the details of the proposed GD-ATSC algorithm, where multiple distributed agents collaboratively control the traffic light signals to maximize the traffic efficiency with limited traffic data exchange. Each agent recovers the neighboring traffic data with the received traffic statistics and utilizes the recovered traffic data for decentralized collaborative training. In addition, each agent locally determines



its traffic light signal based on its local traffic state, thereby achieving decentralized execution. Our proposed framework contains three main modules: one local decision-making module, one traffic data recovery module, and one local training module. Specifically, the local training module consists of four major components (i.e., one memory buffer,  $C$  decision networks, one hyperparameter network, and one mixing network) and is responsible for training the parameters of the decision networks. Note that the proposed GD-ATSC algorithm is an off-policy algorithm and does not require the frequent exchange of the traffic statistics (e.g.,  $\{k_j\}_{j \in \Phi_i}$ ) with other neighboring intersections. We discuss these modules in detail as follows:

1) *Decentralized Execution*: Each agent determines its local traffic light signal according to a decision network, which is an MLP with three fully-connected layers. The MLP takes the local state as the input and generates the traffic light signal as the output. The decentralized execution process can be described as follows. After obtaining the local state  $s_i(t)$ , agent  $i$  determines its traffic light signal  $\rho_i(t)$  at the beginning of time slot  $t$ . At the end of time slot  $t$ , agent  $i$  obtains the average velocity  $\bar{v}_i(t)$ , and then the traffic state updates according to the state probability transition matrix  $\mathbf{P}$ . This observation-decision-transition process repeats over time slots. After extracting the traffic statistics  $k_i$  from the local observation  $o_i(t)$ , agent  $i$  shares  $k_i$  with its neighboring agents in set  $\Phi_i$ . Based on the received traffic statistics and its local observation, each agent obtains the recovered TDM (e.g.,  $\mathbf{G}_i(t)$ ). Further, by extracting the average velocity  $\{\bar{v}_j(t)\}_{j \in \Phi_i}$  from the received traffic statistics, agent  $i$  obtains the regional reward according to (8). After obtaining the recovered TDM  $\mathbf{G}_i(t)$ , the traffic light signals  $\{\rho_j(t)\}_{j \in \Phi_i}$ , and the reward  $r_i(t)$ , agent  $i$  time-stamps the data and stores them as an experience in the memory buffer  $\mathcal{B}$  with the size of  $|\mathcal{B}|$ . Specifically, the stored data are packed as a tuple in the form of  $(\mathbf{G}_i(t-1), \mathbf{a}_i^R(t-1), r_i(t-1), \mathbf{G}_i(t))$ . The memory buffer stores the experiences based on first-in-first-out (FIFO) policy. It is worth noting that the traffic statistics are not required to be exchanged in a real-time manner, as the training and execution processes can be asynchronous.

2) *Decentralized Training*: Each agent  $i \in \mathcal{N}$  possesses  $C$  decision networks, where one decision network is employed to determine the local traffic light signal and others are utilized for mimicking the decision networks of neighboring intersections. By utilizing the sampled experiences from the memory buffer  $\mathcal{B}$ , each agent updates its policy via updating the parameters of  $C$  decision networks and one hyperparameter network. The training process is summarized as follows. According to the TDM  $\mathbf{G}_i(t)$ , agent  $i$  obtains its local state  $s_i(t)$  and the recovered states of the neighboring intersections  $\{\tilde{s}_j(t)\}_{j \in \Phi_i}$ . Given the traffic light signals  $\{\rho_i(t)\}_{i \in \Phi_i^\cup}$ ,  $C$  decision networks of agent  $i$  outputs  $C$  local state-action values, i.e.,  $\{\tilde{Q}_j(s_j(t), a_j(t))\}_{j \in \Phi_i^\cup}$ . For notational ease, we abbreviate  $\tilde{Q}_j(s_j(t), a_j(t))$  and  $Q_i^R(s_i^R(t), \mathbf{a}_i^R(t))$  as  $\tilde{Q}_j$  and  $Q_i^R$ , respectively, in the rest of the paper. By mixing  $\{\tilde{Q}_j\}_{j \in \Phi_i^\cup}$  with dynamic weights, the mixing network approximates the mapping function  $f$  and generates the regional joint state-action value  $Q_i^R$ , where the dynamic weights are generated by the

hyperparameter network with TDM  $\mathbf{G}_i(t)$ . Based on the received reward  $r_i(t)$ , agent  $i$  calculates the training loss via exploiting the regional joint state-action value  $Q_i^R$  and temporal difference. The parameters of the decision networks and the hyperparameter networks are updated according to the gradient of the training loss. The process flow is illustrated in Fig. 5. We discuss the main components of the local training module as follows.

- **Decision Network**: In each agent, there are  $C$  independent MLPs corresponding to  $C$  individual decision networks. Specifically, each MLP is a three-layer fully-connected neural network, where the activation layers between the linear layers are set as rectified linear unit (ReLU). It is worth noting that among these  $C$  MLPs, only one MLP corresponds to the decision network of agent  $i$ , while other MLPs are designed to mimic the decision networks of agents in set  $\Phi_i$ , which will be further discussed in the mixing network. We let the  $i$ -th MLP of agent  $i$  represent its decision network, which first takes  $s_i(t)$  as the input and calculates the local state-action values for all possible traffic light signals and then chooses the traffic light signal that corresponds to the largest state-action value as the output. Besides, exploration during the training is achieved by adopting the  $\epsilon$ -greedy strategy. Specifically, in each time slot, agent  $i$  chooses a signal randomly from its action space  $\mathcal{A}_i$  with probability  $\epsilon(t)$ , and chooses the traffic light signal that corresponds to the largest state-action value with probability  $1 - \epsilon(t)$ . We initialize the value of  $\epsilon(t)$  with a relatively large value and reduce it over time slots until reaching the pre-determined minimum threshold.
- **Mixing Network**: The mixing network is designed to obtain a desired combination of the local state-action values  $\{\tilde{Q}_j\}_{j \in \Phi_i^\cup}$ . This is calculated at each agent locally. Each agent  $i$  obtains the chosen traffic light signals of neighboring intersections  $\{\rho_j(t)\}_{j \in \Phi_i}$  via the received traffic statistics  $\{k_j\}_{j \in \Phi_i}$ . With the recovered neighboring states  $\{\tilde{s}_j(t)\}_{j \in \Phi_i}$ , each agent utilizes other  $(N-1)$  MLPs, which are designed to mimic the decision networks of the neighboring intersections, to generate the local state-action values. In particular, the state-action value  $\tilde{Q}_j$  is calculated based on the recovered state  $\tilde{s}_j(t)$  and the actual traffic light signal  $\{\rho_j(t)\}_{j \in \Phi_i^\cup}$ . After obtaining  $\{\tilde{Q}_j\}_{j \in \Phi_i^\cup}$ , the weights and bias that are generated by the hyperparameter network are mixed to approximate the regional joint state-action value  $Q_i^R$ . In order to incorporate the nonlinear relationship between  $Q_i^R$  and  $\{\tilde{Q}_j\}_{j \in \Phi_i^\cup}$ , we consider the following mapping function

$$f(\tilde{\mathbf{Q}}) = \delta(\tilde{\mathbf{Q}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{w}_2 + \mathbf{b}_2, \quad (17)$$

where  $\tilde{\mathbf{Q}} \triangleq (\tilde{Q}_1, \dots, \tilde{Q}_C)$  denotes the input vector of the mixing network and  $\delta(\cdot)$  denotes the nonlinear activation function. The weights  $\{\mathbf{W}_1, \mathbf{w}_2\}$  and biases  $\{\mathbf{b}_1, \mathbf{b}_2\}$  in (17) are the inputs of the mixing network and they are generated by the hyperparameter network. By further taking  $\tilde{\mathbf{Q}}$  as the input, the mixing network generates  $Q_i^R$  as the output. As the nonlinear activation function has the

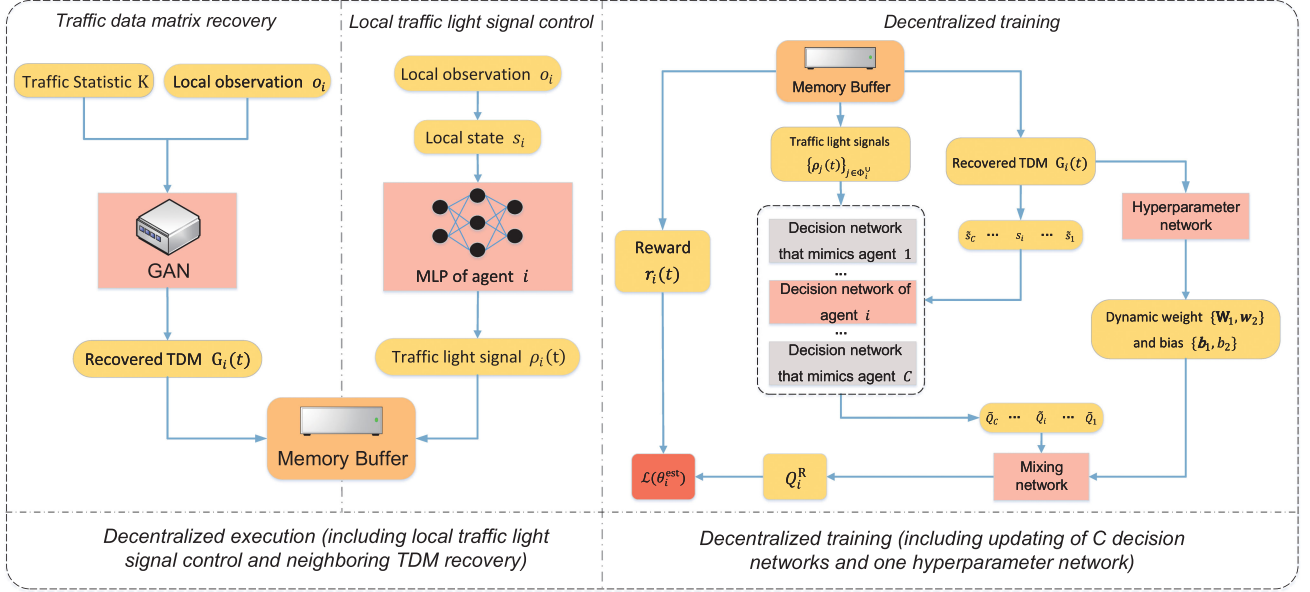


Fig. 5. Illustration of the proposed GD-ATSC framework consisting of a memory buffer, a hyperparameter network, a mixing network, and  $C$  decision networks. Each agent  $i \in \mathcal{N}$  samples the stored experiences from the memory buffer, which are used to train  $C$  decision networks of agent  $i$ . The hyperparameter network generates two pairs of weight and bias based on the recovered TDM. By taking the output of  $C$  decision networks and the hyperparameter networks as the input, the mixing network generates the regional joint state-action value  $Q_i^R$ . The training loss is calculated according to (18). The gradient of the loss function with respect to the weights is calculated and the parameters of both  $C$  decision networks and the hyperparameter network are updated.

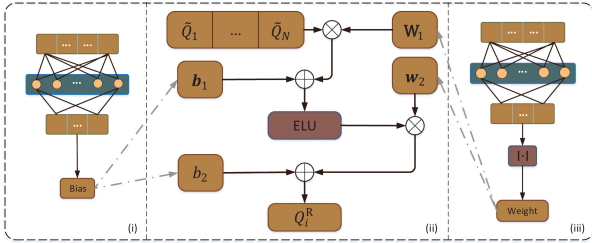


Fig. 6. Structure of the mixing network in one agent, where (ii) illustrates the function of mixing network, and (i) and (iii) show the network structures for generating the bias and the weight, respectively.

potential to build a robust neural network, we use the following nonlinear function to handle the inputs

$$\mathbf{q}_{\text{temp}} = \text{ELU}(\tilde{\mathbf{Q}}\mathbf{W}_1 + \mathbf{b}_1),$$

where the exponential linear unit (ELU) function is defined as  $\text{ELU}(\cdot) = \max(0, x) + \min(0, \beta(e^x - 1))$ , and the value of coefficient  $\beta$  is set to 1 in this paper. As a result, the output of the mixing network can be expressed as

$$Q_i^R = \mathbf{q}_{\text{temp}}\mathbf{w}_2 + b_2.$$

Note that the main operation of the mixing network is the matrix multiplication, as shown in Fig. 6. Since the input of the mixing network is  $\tilde{\mathbf{Q}}$ , the proposed GD-ATSC framework can be extended to incorporate more agents by linearly increasing the dimension of the mixing network.

- **Hyperparameter Network:** Due to the variation of traffic conditions over time, the contribution of each agent to the regional traffic efficiency also varies. This variation is

captured by the weights and biases in the mapping function  $f$ . The hyperparameter network is capable of generating dynamic weights and biases based on the recovered TDM  $\mathbf{G}_i(t)$ . The hyperparameter network contains two weight generators and two bias generators. Specifically, each weight generator is built with two-layer full-connected linear layers, and the activation layer between these two linear layers is also set as ReLU. According to (12), we restrict the weights to be non-negative to reduce the time required to search for the optimal weight, by applying an absolute value function  $|\cdot|$  after the output layer. Each bias generator is also a neural network with the same structure as the weight generator except for the absolute value function. It is worth emphasizing that even though the input dimensions of the two weight generators are the same, the output dimensions are different. In particular,  $\mathbf{W}_1$  is a  $C \times D$  matrix and  $\mathbf{w}_2$  is a  $D \times 1$  vector, where  $D$  is a hyperparameter that determines the size of  $\mathbf{q}_{\text{temp}}$ . Similarly, for the bias generators,  $\mathbf{b}_1$  is an  $1 \times L$  vector, while  $b_2$  is a scalar.

We sample a mini-batch of data from the memory buffer, and calculate the MSE via the temporal difference (TD) as follows:

$$\mathcal{L}(\theta_i^{\text{est}}) = \frac{1}{N_b} \sum_{j=1}^{N_b} (y_{\text{sp}}^R - Q_i^R(s_{\text{sp}}, \mathbf{a}_{\text{sp}}; \theta_i^{\text{est}}))^2, \quad (18)$$

$$y_{\text{sp}}^R = r_{\text{sp}} + \gamma \max_{\mathbf{a}_{\text{sp}}} Q_i^R(s'_{\text{sp}}, \mathbf{a}'_{\text{sp}}; \theta_i^{\text{tar}}),$$

where  $N_b$  denotes the size of the mini-batch,  $s'_{\text{sp}}$  and  $\mathbf{a}'_{\text{sp}}$  are the next regional joint state and next regional joint action in the sampled experience, respectively,  $r_{\text{sp}}$  denotes the regional reward in the sampled experience, and  $\theta_i^{\text{est}}$  and  $\theta_i^{\text{tar}}$  denote the

**Algorithm 2:** Proposed GD-ATSC algorithm.

---

**Input:** Mini-batch size  $N_b$ , discount factor  $\gamma$ , size of memory buffer  $|\mathcal{B}|$ , target network updating interval  $\mathcal{T}$ , and well-trained generator  $\mathcal{G}$ .

- 1 Randomly initialize the network parameters of each agent.
- 2 **for each episode do**
- 3   **for each step  $t$  of episode do**
- 4     **for agent  $i \in \mathcal{N}$  do**
- 5       Take an observation  $o_i(t)$ , transform  $o_i(t)$  into local state  $s_i(t)$ .
- 6       Choose action  $a_i(t)$  according to the decision network.
- 7       Receive the traffic statistics  $\{k_j\}_{j \in \Phi_i}$  from neighboring intersections.
- 8       Generate TDM  $\mathbf{G}_i(t)$  via GAN.
- 9       Store transition  $(\mathbf{G}_i(t-1), \mathbf{a}_i^R(t-1), r_i(t-1), \mathbf{G}_i(t))$  into the memory buffer of agent  $i$ .
- 10      **if  $t > |\mathcal{B}|$  then**
- 11       Uniformly sample  $N_b$  experiences from the memory buffer of agent  $i$ .
- 12       Update the estimator network of agent  $i$  by using Adam optimizer according to (18).
- 13       **if  $t \bmod \mathcal{T}$  then**
- 14           $\theta_i^{\text{tar}} \leftarrow \theta_i^{\text{est}}$

---

parameters of the estimator network and the target network of agent  $i$ , respectively. In particular, both  $\theta_i^{\text{est}}$  and  $\theta_i^{\text{tar}}$  include the parameters of the decision network and the hyperparameter network. The mixing network does not provide any parameters as it is designed to combine the dynamic weights and the local state-action values. After obtaining the gradient of  $\theta_i^{\text{est}}$  according to backpropagation based on the MSE, where the MSE is calculated according to (18) with the sampled experiences, the parameters of the estimator network  $\theta_i^{\text{est}}$  can be optimized by using the stochastic gradient descent method. The parameters of the target network  $\theta_i^{\text{tar}}$  are fixed when optimizing  $\theta_i^{\text{est}}$ . For every fixed interval  $\mathcal{T}$ , we copy the parameters of the estimator network  $\theta_i^{\text{tar}}$  to update the parameters of the target network. It is worth noting that the interval  $\mathcal{T}$  is generally assumed to be much longer than one time slot. In Algorithm 2, we summarize the above proposed GD-ATSC algorithm to solve the problem of traffic management in large-scale traffic networks.

## V. SIMULATION RESULTS

In this section, we present the simulation results to validate the effectiveness of the proposed GAN-based traffic data recovery algorithm and multi-agent DRL-based ATSC algorithm.

### A. Parameter Settings

The simulations are conducted on the open source platform CityFlow [42], which supports the swift and efficient simulations

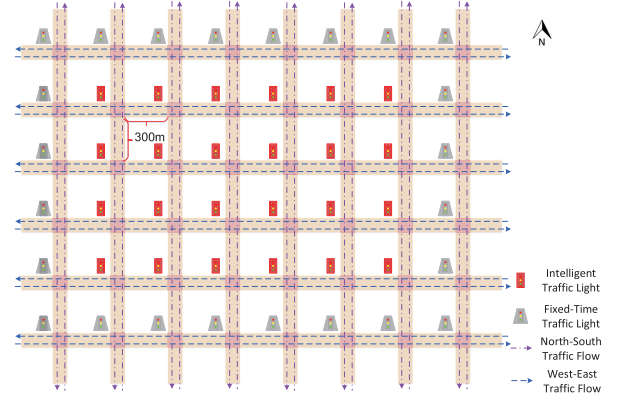


Fig. 7. Bird-view of an urban traffic network. There are 24 intersections with intelligent traffic lights and 24 intersections with fixed-time traffic lights. The length of each road is equal to 300 m. The maximum speed is 10 m/s. For simplicity, the yellow signal of the traffic light is ignored.

of large-scale traffic light signal control. In the simulations, we consider a traffic network with  $6 \times 8$  intersections, as shown in Fig. 7. In particular, the predetermined threshold  $d$  is set to be 1 and there are 24 intelligent agents deployed at the 24 internal intersections that are equipped with intelligent traffic lights. In addition, the traffic lights in the outermost intersections of the traffic network adopt the fixed-time scheduling with random initiations. The outermost intersections are deployed to guarantee that each intelligent agent has four neighboring intersections in one Manhattan distance. Therefore, only the 24 internal intersections are dynamically controlled. The duration of each traffic light phase is 10 s. The road between the adjacent intersections is a two-way road and its length is 300 m. On each direction of each road, there is one lane. The vehicle length is 3.5 m and the minimal gap with the leading vehicle is 1.5 m. The grid length is set to be 3.5 m, and thus the number of grids in each lane is 86. As shown in 7, there are 12 North-South straight traffic flows, 16 West-East straight traffic flows, and 192 non-straight traffic flows.

In addition, we consider dynamic traffic flows with double peaks for traffic flows in the North-South direction and West-East direction, where the arrival rates of vehicles over time are specified by the following formula

$$s_{\text{straight}}(t) = \alpha \exp\left(-\frac{(t - \mu_1)^2}{\sigma_1^2}\right) + \beta \exp\left(-\frac{(t - \mu_2)^2}{\sigma_2^2}\right) + \zeta,$$

where  $t$  denotes the time,  $\{\mu_1, \mu_2\}$  denote the locations of peaks,  $\{\sigma_1^2, \sigma_2^2\}$  denote the variances,  $\zeta$  is a constant, and  $\{\alpha, \beta\}$  are the hyperparameters that determine the peak values. The arrival rate of vehicles that take left and right turning is specified by  $s_{\text{turning}}(t) = \omega \exp(-\frac{(t - \mu_3)^2}{\sigma_3^2}) + \zeta$ , where  $\omega$ ,  $\mu_3$ , and  $\sigma_3^2$  denote the hyperparameter, the location of the peak, and the variance, respectively. Fig. 8 shows the variation of the vehicle arrival rate over time. In particular, for the straight traffic flows in the West-East (North-South) direction, we set  $\mu_1 = 600$  (600),  $\mu_2 = 1600$  (1800),  $\sigma_1^2 = 144$  (144),  $\sigma_2^2 = 144$  (225),  $\alpha = 150$



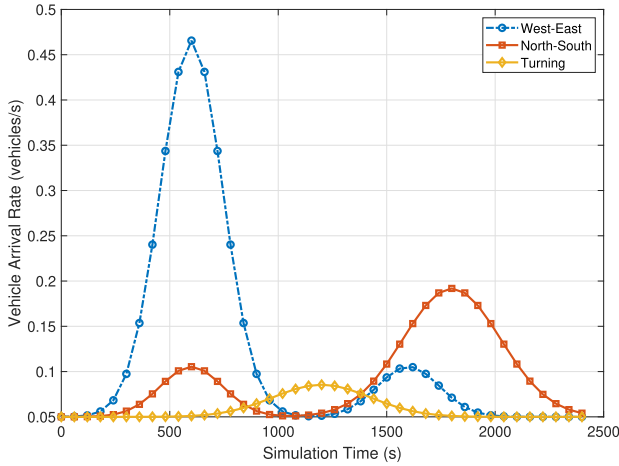


Fig. 8. Traffic flows versus simulation time within the network. The legends “North-South” and “West-East” represent the straight traffic flows in North-South direction and West-East direction, respectively, and the “Turning” represents the traffic flows with the left and right turning.

(20), and  $\beta = 20$  (80). For the traffic flows with the left and right turning, we set  $\mu_3 = 1200$ ,  $\sigma_3^2 = 225$ , and  $\omega = 20$ . For all the traffic flows, constant  $\zeta$  is set to be 0.05. With the above settings, the mean (maximum) arrival rates of each straight traffic flow in the West-East and North-South directions are 0.1191 (0.4656) and 0.0906 (0.1918) vehicles/s, respectively. The mean and maximum arrival rates of the traffic flow with the left and right turning are 0.0581 and 0.0855 vehicles/s, respectively.

Given the traffic flow data, each vehicle moves toward its destination according to the predetermined traffic flow setting. The car following model used in CityFlow is a modification of the model proposed by Stephen Krauß which enables the vehicle to drive as fast as possible subject to perfect safety regularization (e.g. being able to stop even if the leading vehicle stops using the maximum deceleration) [42], [43]. For all vehicles, the maximum speed is 10.0 m/s, the maximum accelerating acceleration is 2.0 m/s<sup>2</sup>, and the maximum decelerating acceleration is 4.5 m/s<sup>2</sup>.

### B. Effectiveness of Proposed GAN-Based Traffic Data Recovery Algorithm

In this subsection, we verify the effectiveness of the proposed GAN-based traffic data recovery algorithm. The historical data, which is used for the training of the GAN-based algorithm, is generated by randomly adjusting the traffic light signals. We record the global TDM and add it to the dataset every 10 s. To improve the convergence performance of the network training, the mini-batch stochastic gradient descent (SGD) is adopted, where a mini-batch of 10 samples are randomly chosen from the dataset for each update period. The average gradients among the 10 mini-batch samples are used to update discriminator  $\mathcal{D}$  and generator  $\mathcal{G}$  with a learning rate of  $\mu = 0.01$ .

An example of traffic data recovery using the proposed GAN-based traffic data recovery algorithm is depicted in Fig. 9, where the recovered traffic data and the actual traffic data are compared. In particular, the observed local traffic data, which is highlighted

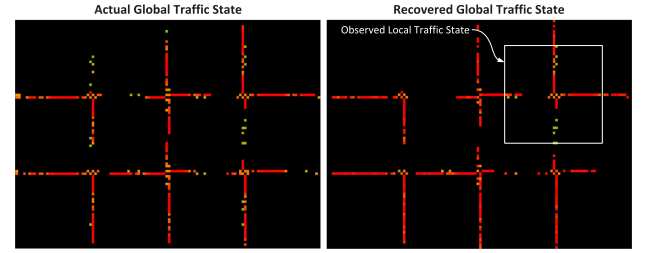


Fig. 9. An example of traffic data recovery using the proposed GAN-based algorithm. A black pixel means that the corresponding grid is empty. A non-black pixel implies the existence of a vehicle within the grid. The vehicle speed is normalized to be within [0,1] and is mapped to a color within { RED, GREEN }.

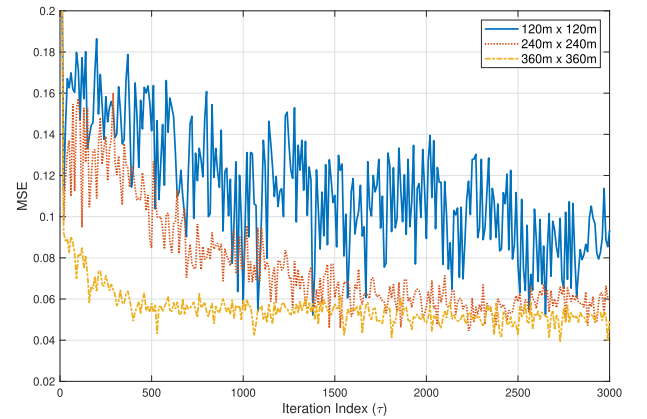


Fig. 10. Convergence behavior of the generator in recovering the global TDM during the GAN training.

with the white square (240 m  $\times$  240 m), is the input to the generator  $\mathcal{G}$ . As shown in Fig. 9, we observe that our proposed algorithm achieves good traffic data recovery performance especially for those congested intersections. On the other hand, due to the fact that the cases with high-speed vehicles occupy a very small portion of the dataset, it is challenging to accurately recover the traffic data of the high-speed vehicles.

In Fig. 10, the recovery performance of the trained generator  $\mathcal{G}$  is plotted after each update iteration during training. It is worth noting that we also compare the performance when the local traffic observation at one particular intersection covers the areas of different sizes, i.e., 120 m  $\times$  120 m, 240 m  $\times$  240 m, and 360 m  $\times$  360 m. The MSE between the real and the recovered traffic information matrices are plotted in Fig. 10. It can be seen that a larger observed area leads to a faster convergence speed and a better recovering performance during the GAN training. This is because a larger observed area provides more traffic data, which improves the accuracy of the traffic data recovery.

Fig. 11 shows the comparison between our proposed traffic data recovery algorithm and other two algorithms. The first baseline algorithm is the Bayesian Gaussian CANDECOMP/PARAFAC (BGCP) algorithm proposed in [44]. The second baseline algorithm is termed as Primitive-GAN, which only utilizes the locally observed state without the traffic statistics collected from the neighboring intersections. Both the cumulative distribution functions (CDFs) of the MSE and the queueing

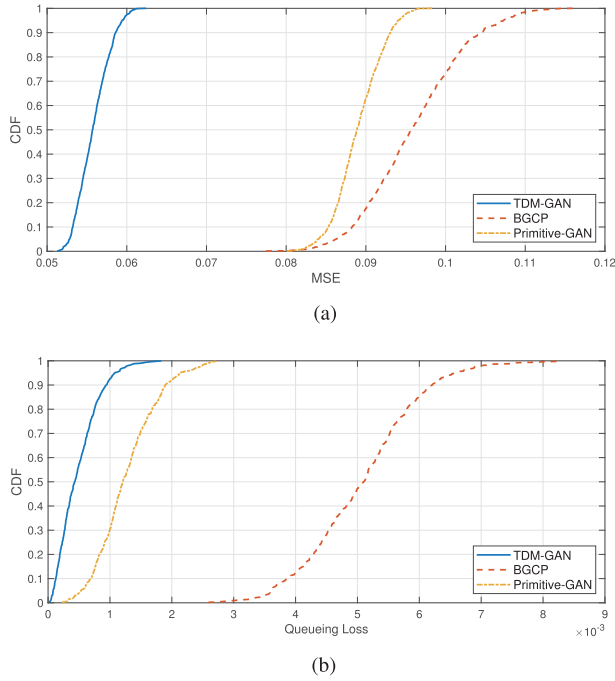


Fig. 11. Recovery performance of different algorithms. (a) CDFs of MSE, (b) CDFs of queueing loss.

loss are compared in Fig. 11. It is worth noting that the queueing loss is an important performance metric in the traffic light signal control system [45], and refers to the average difference between the queue size based on the recovered traffic data and the actual queue size in the traffic network. For each particular road, the traffic queue consists of all vehicles on the road with speed below 0.1 m/s. Note that the queue size is normalized to be within [0,1] in the simulations. As shown in Fig. 11(a), by exploiting the traffic statistics from other intersections, our proposed GAN-based traffic data recovery algorithm achieves better performance than the Primitive-GAN and BGCP algorithms. In Fig. 11(b), both the GAN-based traffic data recovery algorithm and the Primitive-GAN algorithm achieve better performance in terms of the queueing loss compared to the BGCP algorithm. Additionally, the superiority of our proposed GAN-based traffic data recovery algorithm corroborates that the GAN model efficiently learns the spatial correlations in the traffic data. Furthermore, through exchanging a small amount of statistical information among different intersections, the recovery performance is significantly improved.

### C. Effectiveness of Proposed GD-ATSC Algorithm

In this subsection, we evaluate the performance of our proposed GD-ATSC algorithm. The main performance metric is the **average travel time**, which is the average time of all vehicles that have spent on traveling through the network. In addition, the average accumulated waiting time is also the performance metric adopted in the following simulations. The accumulated waiting time of one vehicle is defined as the summation of the time this vehicle spent in waiting while traveling through the network, while the average accumulated waiting time refers to the average

TABLE II  
PERFORMANCE OF DIFFERENT ALGORITHMS

Methods	GD-ATSC	RLight	Fixed-time	Random-time	PressLight
Ave. tra. time (s)	<b>444.3</b>	495.1	643.9	1708.4	554.6
Med. tra. time (s)	<b>460.0</b>	480	630	1650	540
Ave. acc. wait. time (s)	<b>185.0</b>	264.8	419.5	1465.4	304.6
Med. acc. wait. time (s)	<b>190</b>	240	420	1410	280

accumulated waiting time of all vehicles in the network. In the simulations, we compare our proposed GD-ATSC algorithm with the following benchmarks:

- **Random-time scheduling:** In the random-timing method, the phases of the 24 intelligent traffic light signals are switched randomly. Specifically, the action is uniformly and randomly selected from the corresponding action space [28], and is updated every 30 s.
- **Fixed-time scheduling:** The traffic light signals are switched sequentially every 30 s, and the initial phases are selected randomly [28].
- **PressLight:** PressLight is an RL based method proposed in [24]. This method combines the traditional traffic light signal control algorithm and deep reinforcement learning, and can form the local collaboration between the adjacent intersections. It uses a reward called the “pressure”, which is defined in [24].
- **RLight:** RLight is an independent multi-agent DQN based algorithm, which is a special case of our proposed algorithm. Specifically, both the state and action of each agent of RLight are set to be the same as that of the proposed algorithm, while the reward is defined as the average velocity of all vehicles on the incoming lanes of the local intersection.

Figs. 12(a) and (b) depict the CDFs of the travel time and the accumulated waiting time of vehicles. Under dynamic traffic flows, the traditional approaches, i.e., fixed-time scheduling and random-time scheduling, achieve longer travel time and accumulated waiting time than the RL-based methods. In particular, only about 45% of the vehicles’ travel time are less than 600 s for the traditional approaches, while more than 95% of the vehicles’ travel time are less than 600 s by applying our proposed algorithm. Compared with PressLight, our proposed algorithm reduces the travel time and the accumulated waiting time by 24.03% and 39.52%, respectively. Additionally, our proposed algorithm reduces the median of the travel time by about 30 s. Our proposed algorithm also outperforms the RLight algorithm in terms of both the average travel time and the average accumulated waiting time by 10.25% and 30.14%, respectively. This is because, each agent of RLight can only access to the local traffic data and receive the reward from the local intersection. However, our proposed algorithm can recover the traffic data in the neighboring intersections and achieve cooperation among the agents, thereby reducing the average travel and accumulated waiting time. Moreover, Table II summarizes the performance of different algorithms under different performance metrics.

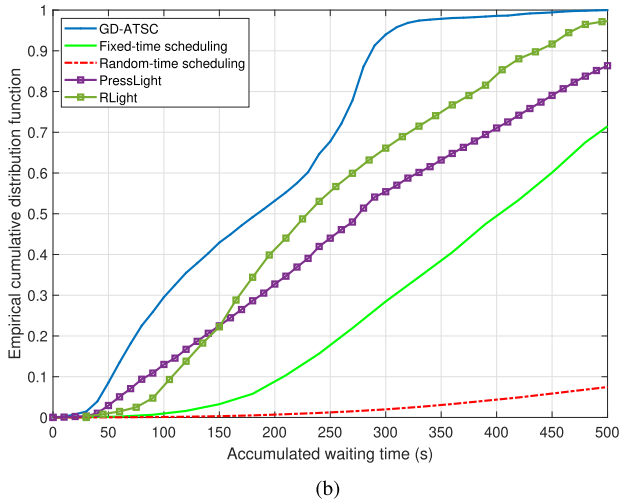
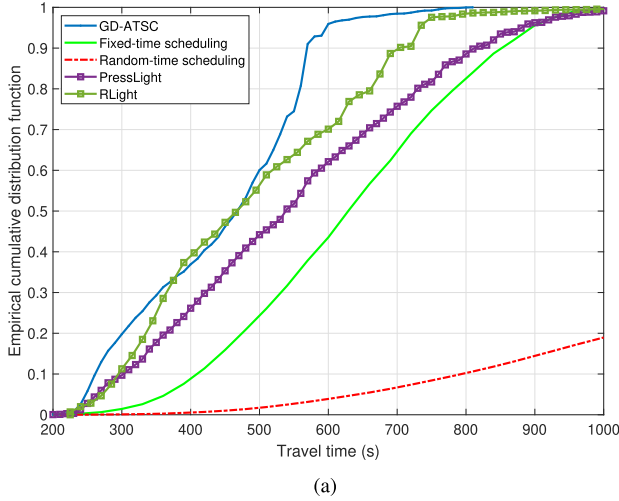


Fig. 12. CDF of travel time and accumulated time between different algorithms. (a) CDF of travel time of all vehicles under different algorithms, (b) CDF of accumulated waiting time of all vehicles under different algorithms.

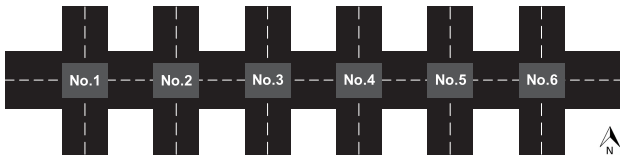


Fig. 13. An arterial road with six intersections. The length of each road in the network is equal to 300 m. For simplicity, the signal for each traffic light is restricted to two: {pass for West-East direction and stop for North-South direction} or {pass for North-South direction and stop for West-East direction}.

#### D. Case Study

In order to further demonstrate the collaboration ability of the proposed GD-ATSC algorithm, we conduct simulations under the scenario consisting of a single-direction arterial road and several branch roads. As shown in Fig. 13, the intersections are labeled from the west to the east, and the vehicle arrival rate on the arterial road is 0.5 vehicles/s. The vehicle arrival rate on the branch roads are 0.05 vehicles/s. The switch interval of the traffic light signals is 15 s. In this situation, the ideal solution

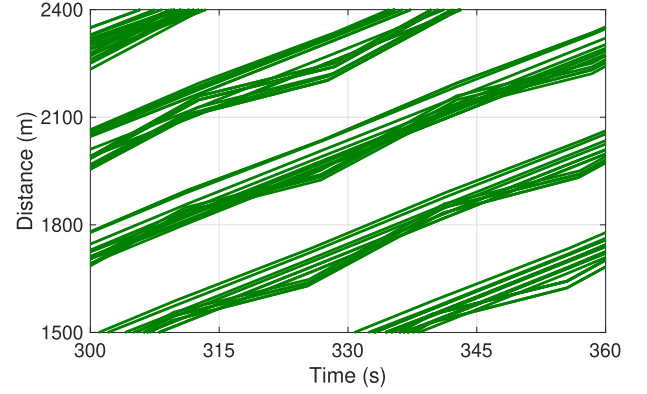


Fig. 14. Space-time diagram of vehicles' trajectories. Each line represents the relationship between one vehicle's travel distance and the simulation time.

is to set the lights to green for West-East direction most of the time, which will form a successive green light sequence along the arterial road, also known as the “Greenwave”.

As shown in Fig. 14, the green lines represent the space-time trajectory of the vehicles in the arterial. Additionally, most of the vehicles cross the network with a stable velocity, and its travel distance is linear with time, indicating that those vehicles are traveling with no stop and enjoying the successive green light. Moreover, the slope of most of the lines is nearly 10.0 m/s, which implies that most of the vehicles are driving at the maximum speed allowed on this road. Besides, the traffic light signals on the West-East direction are allocated a green light most of the time. Specifically, the number of green light phases allocated to the North-South direction and the West-East direction are 941 and 5059, respectively, which is consistent with the ideal solution to allocate green lights to the West-East direction most of the time.

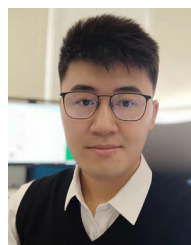
#### VI. CONCLUSION

In this paper, we proposed a decentralized ATSC framework for large-scale traffic networks with bandwidth-limited communication links between neighboring intersections. We developed a GAN-based algorithm to recover the neighboring traffic data with traffic statistics and proposed a multi-agent DRL based algorithm by adopting the value decomposition technique, which enables each intersection to independently determine its traffic light signal based on its local traffic data. With a combination of GAN and multi-agent DRL, a novel ATSC framework that is capable of achieving decentralized training and decentralized execution was developed to collaboratively control the traffic light signals. Simulation results demonstrated that the proposed GD-ATSC algorithm can significantly improve the traffic efficiency in terms of both the average accumulated waiting time and the average travel time. In particular, the proposed GD-ATSC algorithm reduced the average accumulated waiting time by 50% when compared with fixed-time scheduling under the dynamic traffic flow. Moreover, we observed the “greenwave” phenomenon that depicts the ability of the proposed algorithm in achieving collaboration among the neighboring intersections.



## REFERENCES

- [1] R. Zhong, R. Xu, A. Sumalee, S. Ou, and Z. Chen, "Pricing environmental externality in traffic networks mixed with fuel vehicles and electric vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5535–5554, Sep. 2021, doi: [10.1109/TITS.2020.2987832](https://doi.org/10.1109/TITS.2020.2987832).
- [2] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1243–1253, Feb. 2019.
- [3] K. Chu, A. Y. S. Lam, and V. O. K. Li, "Traffic signal control using end-to-end off-policy deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: [10.1109/TITS.2021.3067057](https://doi.org/10.1109/TITS.2021.3067057).
- [4] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of Things is the backbone," *IEEE Consum. Electron. Mag.*, vol. 5, no. 3, pp. 60–70, Jul. 2016.
- [5] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-things-based smart cities: Recent advances and challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, Sep. 2017.
- [6] P. W. Hunt, D. I. Robertson, R. D. Bretherton, and M. Royle, "The SCOOT on-line traffic signal optimisation technique," *Traffic Eng. Control*, vol. 23, no. 4, Apr. 1982.
- [7] M. Chang, J. M. Carroll, and J. S. Alberto, "Timing traffic signal change intervals based on driver behavior," *Transp. Res. Rec.*, vol. 1027, pp. 20–30, 1985.
- [8] S. Chiu and S. Chand, "Adaptive traffic signal control using fuzzy logic," in *Proc. IEEE Regional Conf. Aerosp. Control Syst.*, Westlake Village, USA, 1993.
- [9] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," Jan. 2020, [arXiv:1904.08117](https://arxiv.org/abs/1904.08117).
- [10] S. Lee, S. Wong, and P. Varaiya, "Group-based hierarchical adaptive traffic-signal control part II: Implementation," *Transp. Res. B-Methodological*, vol. 104, pp. 376–397, 2017.
- [11] Y. Du, W. ShangGuan, and L. Chai, "A coupled vehicle-signal control method at signalized intersections in mixed traffic environment," *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2089–2100, Mar. 2021.
- [12] D. Liu, W. Yu, S. Baldi, J. Cao, and W. Huang, "A switching-based adaptive dynamic programming method to optimal traffic signaling," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4160–4170, Nov. 2020.
- [13] L. Li, W. Huang, A. H. F. Chow, and H. K. Lo, "Two-stage stochastic program for dynamic coordinated traffic control under demand uncertainty," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: [10.1109/TITS.2021.3118843](https://doi.org/10.1109/TITS.2021.3118843).
- [14] H. Wang, M. Zhu, W. Hong, C. Wang, G. Tao, and Y. Wang, "Optimizing signal timing control for large urban traffic networks using an adaptive linear quadratic regulator control strategy," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: [10.1109/TITS.2020.3010725](https://doi.org/10.1109/TITS.2020.3010725).
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
- [16] B. Abdulhai, P. Rob, and J. K. Grigoris, "Reinforcement learning for true adaptive traffic signal control," *J. Transp. Eng.*, vol. 129, no. 3, pp. 278–285, Apr. 2003.
- [17] P. LA and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 412–421, Jun. 2011.
- [18] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," in *Proc. NIPS Deep Learn. Workshop*, Lake Tahoe, USA, pp. 1–9, Dec. 2013.
- [19] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [20] M. Wang, L. Wu, J. Li, and L. He, "Traffic signal control with reinforcement learning based on region-aware cooperative strategy," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: [10.1109/TITS.2021.3062072](https://doi.org/10.1109/TITS.2021.3062072).
- [21] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, London, U.K., 2018, pp. 2496–2505.
- [22] C. Wan and M. Hwang, "Value-based deep reinforcement learning for adaptive isolated intersection signal control," *IET Intell. Transp. Syst.*, vol. 12, no. 9, pp. 1005–1010, Nov. 2018.
- [23] H. Pang and W. Gao, "Deep deterministic policy gradient for traffic signal control of single intersection," in *Proc. Chin. Control Decis. Conf.*, Nanchang, China, 2019, pp. 5861–5866.
- [24] H. Wei *et al.*, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, Anchorage, USA, 2019, pp. 1290–1298.
- [25] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, Apr. 2018.
- [26] I. Althamary, C.-W. Huang, and P. Lin, "A survey on multi-agent reinforcement learning methods for vehicular networks," in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, Tangier, Morocco, 2019, pp. 1154–1159.
- [27] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, "Evolutionary dynamics of multi-agent learning: A survey," *J. Artif. Intell. Res.*, vol. 53, pp. 659–697, Aug. 2015.
- [28] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2687–2700, Jun. 2019.
- [29] H. Wei *et al.*, "Colight: Learning network-level cooperation for traffic signal control," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, Beijing, China, 2019, pp. 1913–1922.
- [30] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2019.
- [31] D. F. Borges, J. P. R. R. Leite, E. M. Moreira, and O. A. S. Carpinteiro, "Traffic light control using hierarchical reinforcement learning and options framework," *IEEE Access*, vol. 9, pp. 99155–99165, 2021.
- [32] T. Wu *et al.*, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8243–8256, Aug. 2020.
- [33] P. Zhou, X. Chen, Z. Liu, T. Braud, P. Hui, and J. Kangasharju, "DRLE: Decentralized reinforcement learning at the edge for traffic light control in the IoV," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2262–2273, Apr. 2021.
- [34] W. Liu, G. Qin, Y. He, and F. Jiang, "Distributed cooperative reinforcement learning-based traffic signal control that integrates v2x networks' dynamic clustering," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 8667–8681, Oct. 2017.
- [35] J. V. S. Busch, V. Latzko, M. Reisslein, and F. H. P. Fitzek, "Optimised traffic light management through reinforcement learning: Traffic state agnostic agent vs. holistic agent with current V2I traffic state knowledge," *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 201–216, 2020, doi: [10.1109/OJITS.2020.3027518](https://doi.org/10.1109/OJITS.2020.3027518).
- [36] F. A. Oliehoek, M. T. Spaan, and N. Vlassis, "Optimal and approximate q-value functions for decentralized pomdps," *J. Artif. Intell. Res.*, vol. 32, pp. 289–353, May 2008.
- [37] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial-temporal correlation," *Physica A*, vol. 446, pp. 54–63, Mar. 2016.
- [38] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Montreal, Canada, pp. 2672–2680, 2014.
- [39] M. He, X. Luo, Z. Wang, F. Yang, H. Qian, and C. Hua, "Global traffic state recovery via local observations with generative adversarial networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 3767–3771.
- [40] P. Sunehag *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. Int. Conf. Auton. Agents MultiAgent Syst.*, Stockholm, Sweden, pp. 2085–2087, 2018.
- [41] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Vienna, Austria, 2018, pp. 4295–4304.
- [42] H. Zhang *et al.*, "Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *Proc. World Wide Web Conf.*, San Francisco, USA, May 2019, pp. 3620–3624.
- [43] S. Krauß, "Microscopic modeling of traffic flow: Investigation of collisionfree vehicle dynamics," Ph.D. dissertation, Universität zu Köln, 1998.
- [44] X. Chen, Z. He, and L. Sun, "A bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transp. Res. C*, vol. 98, pp. 73–84, Jan. 2019.
- [45] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Design of reinforcement learning parameters for seamless application of adaptive traffic signal control," *J. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 227–245, May 2014.



**Zixin Wang** received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His research interests include Internet of Vehicles, over-the-air computation, and federated learning.



**Hanyu Zhu** received the B.Eng. degree from Nanchang University, Nanchang, China, in 2016, and the Ph.D. degree from the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, in 2021. He is currently an Engineer with Purple Mountain Laboratories, Nanjing, China. His research interests include intelligent traffic system and Internet of Vehicles.

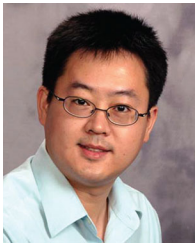


**Mingcheng He** received the B.S. and M.Eng. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2013 and 2017, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include space-air-ground integration networks and machine learning in wireless networks.



**Yong Zhou** received the B.Sc. and M.Eng. degrees from Shandong University, Jinan, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2015. From November 2015 to January 2018, he was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. He is currently an Assistant Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His research interests

include 6G communications, edge intelligence, and Internet of Things.



**Xiliang Luo** (Senior Member, IEEE) received the B.Sc. degree in physics from Peking University, Beijing, China, in 2001, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 2003 and 2006, respectively.

After finishing his Ph.D. studies, he joined Qualcomm Research and was involved in the system designs, analyses, and standardization of 4G LTE. He was the Designer of various enhancements to Qualcomm's LTE solutions and led the designs of heterogeneous networks from initial concepts to successful modem development.

Since 2014, he has been with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. He has authored or coauthored more than 100 research papers in top journals and conferences. He is the Co-Inventor of more than 70 U.S. and international patents and majority of those have been adopted into current 4G and 5G wireless communication standards. His general research interests include signal processing, communications, and machine learning. Particularly, he is interested in researches combining information theory and machine learning theory that can shape and guide the designs of next generation data and information processing networks. In 2017, he was the recipient the Excellent Paper Award from the IEEE ICUFN.



**Ning Zhang** received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2015. He is currently an Associate Professor and a Canada Research Chair (Tier 2) with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada. After that, he was a Postdoc Research Fellow with the University of Waterloo and University of Toronto, Toronto, ON, Canada, respectively. His research interests include connected vehicles, mobile edge computing, wireless networking, and machine learning. He is a Highly Cited Researcher and has 20 ESI highly cited papers. He is an Associate Editor of the IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and IEEE SYSTEMS JOURNAL, and a Guest Editor of several international journals, such as the IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is/was also a TPC Chair for IEEE VTC 2021 and IEEE SAGC 2020, the General Chair for IEEE SAGC 2021, a Track Chair for several international conferences and workshops. He was the recipient of eight best paper awards from conferences and journals, such as IEEE Globecom and IEEE ICC. He was also the recipient of the IEEE TCSVC Rising Star Award for outstanding contributions to research and practice of mobile edge computing and Internet of things service.