Distributed Signal Control of Arterial Corridors Using Multi-Agent Deep Reinforcement Learning

Weibin Zhang, Chen Yan, Xiaofeng Li, Liangliang Fang, Yao-Jan Wu, and Jun Li, *Senior Member, IEEE*

***Abstract*— Traffic congestion at signalized intersections often leads to serious impacts on adjacent intersections on a corridor. To enhance intersections' throughput efficiency, traffic signals are commonly coordinated across intersections. Traditional signal coordination methods control the adjacent intersections by setting a fixed phase offset. However, these traditional coordination methods may have poor adaptability to dynamic traffic conditions, which can cause additional congestion. To reduce arterial traffic delays, this paper develops an adaptive coordination control method based on multi-agent reinforcement learning (MARL). Most existing MARL-based methods rely on impractical assumptions to improve their performance in complex and dynamic traffic scenarios. To overcome these assumptions, this paper proposes a fully scalable MARL algorithm for arterial traffic signal coordination based on the proximal policy optimization algorithm. We apply a parameter-sharing training protocol to mitigate the slow convergence due to nonstationarity and to reduce computational requirements. In addition, a new action setting is designed by using the lead-lag phase sequence to simultaneously improve the implementation and coordination flexibility of the method. Extensive simulation experiments and comparisons with existing methods demonstrate that the proposed method performed stably in both simulated and real-world arterial corridors. Hence, the proposed signal coordination method can alleviate traffic congestion more effectively than existing traditional and MARL-based methods.**

***Index Terms*— Arterial traffic signal control, reinforcement learning, deep reinforcement learning, multi-agent reinforcement learning, proximal policy optimization.**

I. Introduction

With global increases in urban populations and economic development, urban transportation demands are growing and traffic congestion is becoming increasingly severe. Dynamic coordination of traffic signals between multiple intersections in arterial corridors and effective diversion of arterial vehicles are potential ways to alleviate urban traffic congestion. At present, signal coordination across multiple intersections on arterial corridors is mainly achieved using traditional methods, such as Maxband [1] and Multiband [2], and green wave methods, such as graphical methods and numerical solutions. These traditional methods have difficulty in effectively coordinating complex dynamic traffic flows as traffic flows are time-varying and vehicle speeds are easily affected.

With the development of deep neural networks (DNNs) [3] and traffic detection technologies, deep reinforcement learning (DRL) has become a potential method for arterial traffic signal control (ATSC) based on real-time traffic measurements. DRL methods have greater advantages than reinforcement learning (RL) methods in solving traffic signal control (TSC) problems with high-dimensional state spaces. Current DRL methods can be categorized into two categories: value-based and policy-based algorithms. The Deep Q-Network (DQN) is a well-known value-based method that is the most commonly used DRL algorithm for single-intersection TSC. The proximal policy optimization (PPO) algorithm estimates an advantage function by generalized advantage estimation (GAE) to make a compromise between variance and bias. The PPO is simple to implement with excellent stability and reliability and is the focus of this paper [4].

The application of multi-agent reinforcement learning (MARL) to multi-intersection TSC is a recent research hotspot [5], [6]. However, many challenges remain in applying it to multiple intersections. Although DNNs have improved the scalability of RL, it is still infeasible to train a centralized policy [7] for ATSC because the joint state space and action space increase exponentially with the number of intersections. A recent study applied independent-advantage actor-critic (IA2C) to large-scale TSC and was inspired by independent Q-learning (IQL) [8] but replaces Q-learning with advantage actor-critic (A2C). IQL and IA2C are fully extensible, but learning strategies for each agent imposes computational and memory stress, and non-stationary environments will cause convergence problems. To resolve the above issues, the proposed method uses the decentralized parameter-sharing training protocol [9] for its ability to alleviate the convergence problems caused by nonstationarity and reduce training time.

An effective action definition is a prerequisite for MARL in finding effective ATSC strategies. Most studies generally use two types of action settings. One is the variable-phase sequence, which is flexible but challenging to implement in real-world ATSC [5], [10], [11], [32] because of the increasing likelihood of traffic accidents due to the uncertainty of the next phase. The other is the fixed-phase sequence [12], [13], [14];

Manuscript received 13 June 2021; revised 13 March 2022 and 7 August 2022; accepted 13 October 2022. This work was supported by the National Natural Science Foundation of China under Grant 71971116. The Associate Editor for this article was A. Hajbabaie. *(Corresponding author: Jun Li.)*

Weibin Zhang, Chen Yan, Liangliang Fang, and Jun Li are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: wbin.zhang@outlook.com; 121104022451@njust.edu.cn; chillbboy@njust.edu.cn; jun.li@njust.edu.cn).

Xiaofeng Li and Yao-Jan Wu are with the Department of Civil and Architectural Engineering and Mechanics, The University of Arizona, Tucson, AZ 85721 USA (e-mail: xfli@arizona.edu; yaojan@arizona.edu).

Digital Object Identifier 10.1109/TITS.2022.3216203

however, traditional split phasing [12] and lag-lag left-turn signal phases are not flexible enough to adapt to dynamic traffic flows. Therefore, we propose a type of flexible action setting that uses the lead-lag left-turn phase [15]. The main contributions of the proposed method are as follows.

1) A MARL-based method is proposed to improve the traffic efficiency of arterial corridors by dynamically controlling the signals at multiple intersections. In the proposed method, local policy and value regressors are based on local observations instead of impractical assumptions due to the limitations of traffic detection technology and communication latency. Additionally, the decentralized execution of this method enables real-time decision-making in real-world ATSC because there is no need for communication when making decisions. Compared with traditional methods, TSCs can effectively cooperate under different traffic demands to not only optimize the efficiency of the through movement on major corridors but also arterial roads with all traffic directions.
2) The independent PPO (IPPO) algorithm is proposed for ATSC, which extends the idea of IA2C to PPO. To make the proposed method more stable and implementable, a parameter-sharing PPO (PS-PPO) algorithm is also proposed based on PPO with the clipped surrogate objective. The use of parameter sharing can mitigate the slow convergence caused by the partial observability and nonstationarity of IPPO, and enhance training stability and efficiency.
3) Two constraints in real-world ATSC are considered when defining the *action* and *reward* components of the MARL model. Under the constraint that the signal status can only switch according to a fixed phase sequence, we propose a more effective action setting using the lead-lag left-turn phase. Moreover, the learned policy restricts queue spillback, since priority is given to lanes that overflow more easily.

The rest of this paper is organized as follows. Section II reviews related studies. The problem statement is presented in Section III. In Section IV, we propose the IPPO and PS-PPO algorithms for ATSC. Section V elaborates upon the implementation details of the proposed MARL model. We describe the simulation used to evaluate our method, and its results, in Section VI. Finally, we conclude the paper in Section VII.

II. Relevant Studies

In traditional ATSC, setting a fixed offset (i.e., the signal cycle start time difference between the intersections and a master clock) among all intersections along an arterial corridor is the most common way to achieve signal coordination. The MAXBAND model [16] used mixed-integer linear programming to optimize cycle length, speeds, offsets, and left-turn phase sequence in order to achieve maximal green-wave bandwidth. The MULTIBAND system subsequently developed by Gartner et al. [17] is a two-way green-wave control model that can adapt to different traffic flow patterns on each link of an arterial road. To make existing methods of green-wave coordination control suitable for arterial corridors under asymmetric two-way traffic conditions, Kai et al. [18] proposed an algebraic method of bidirectional green-wave coordinated control by using phase combination and velocity transformation methods. Unlike green-wave methods, max pressure [19], [20] stabilizes the queues and aims to maximize the throughput of the network. However, these existing traditional methods are not guaranteed to provide the best signal timing in the field under variable traffic conditions because they rely on an assumption of simplified traffic conditions.

Efforts have been made to apply the recently-developed MARL method to multi-intersection traffic signal coordination control. In MARL-based signal coordination control, there are usually two types of controllers: joint action learners [21], [22] and independent learners. Joint action modeling methods learn to choose the optimal joint action for different joint observations. The disadvantage of these methods is that the dimensions of the state space and action space increase exponentially with the number of intersections. To alleviate this problem, Literature [22] decomposes the global q-value into a linear combination of local Q-values based on the max plus algorithm [35]. Some other works [36], [37] further treat the joint Q-value as a weighted sum of local Q-values. Unlike joint action learning methods, each agent learns an independent policy based on local observations, and independent RL methods allow each agent to learn an independent policy based on local observations [38], [39], [40], [41], [42], [43]. In some scenarios with a simple arterial corridor, these methods can be applied to optimize signal timing plans with a maximize green wave. However, the problem of non-stationarity is exacerbated when the road environment becomes complex [33], and there are challenges in converging to a stationary policy when there is no communication or coordination mechanism between agents. Literature [32] designs a reward function based on max pressure theory to achieve coordinated control of arterial corridors, but the action set was based on a non-fixed phase sequence. When communication is reliable between agents, the traffic status of neighbors is typically added to the agent's observations [44], rather than just using observations from local intersections. Literature [45], [46] propose to exploit the graph attention network [34] to learn the dynamic interactions between hidden states of adjacent intersections. The newly proposed parameter sharing method [9] can effectively alleviate the environmental non-stationarity in cooperative multi-agent control without the need for communication between agents. Recent research has been more inclined to use variable phase sequences [11], [12] to define actions that may be unsuitable for the real world. To further improve the applicability and efficiency of the proposed method, we added a constraint where the signal status switches using a fixed phase sequence with the lead-lag left-turn phase sequence.

III. Problem Statement

The goal of this paper is to minimize global traffic congestion throughout the entire arterial corridor by distributed control of all the traffic lights with better adaptability to dynamic traffic conditions and fewer computational requirements. As shown in Fig.1,we consider an arterial with N

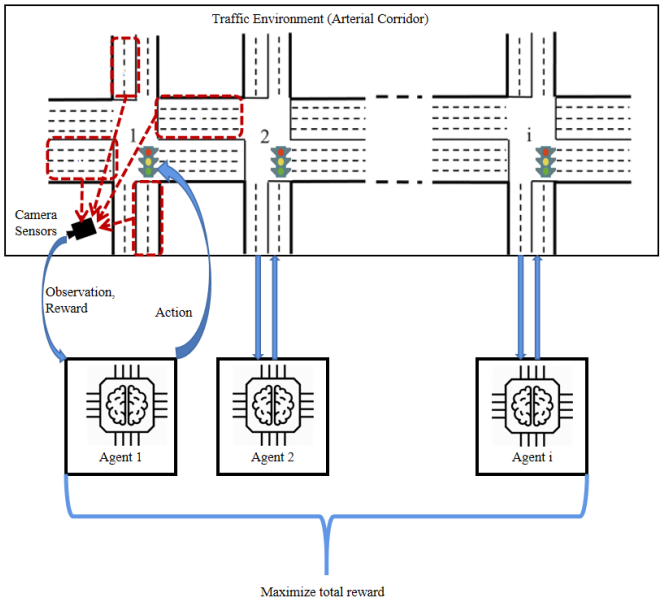


Fig. 1. Multi-agent deep reinforcement learning model for arterial traffic signal control.

intersections, $N > 1$. Each intersection has multiple crossroads consisting of incoming lanes and outgoing lanes. The vehicle clearance process on each incoming lane follows the current signal phase, which is one of the legal combinations of red and green signals for all traffic lights at an intersection. We assume that right-turning vehicles at all intersections are not controlled by signals since vehicles are allowed to turn right during a safe gap. According to real-world constraints, the phases are switched in a fixed order. When the signal in one direction turns green to red, the yellow duration is inserted for t_y seconds. In order to ensure that pedestrians have enough time to cross the street, the green duration t of each phase shall not be less than t_{min} (minimum green time), and the green time $t$cannot be greater than t_{max} (maximum green time). In this study, traffic state information from the incoming lanes of local intersections is collected via camera sensors, as shown by the red dotted line in Fig. 1. However, traffic state may be very difficult to achieve in real-time ATSC due to sensor faults, communication delays, and losses of signal controllers. Therefore, each agent can only observe part of the state of the entire arterial corridor, which is coincident with reality and enhances the authenticity of the proposed model. In the real world, adjacent signalized intersections on arterial corridors may have a short distance causing spillback. In the case of over-saturated traffic, spillback is likely to occur, and adjacent agents need to coordinate effectively to prevent overflow. In order to dynamically adjust signals at all intersections according to different traffic states under the above constraints by MARL, the average delay of all vehicles on an arterial corridor is minimized. Naturally, such an arterial traffic control problem can be modeled as a decentralized partially-observable Markov decision process [25] (Dec-POMDP) defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{Z}, P, r, O, \gamma \rangle$. Here $\mathcal{N} := \{1, \ldots, N\}$ is a finite set of N agents each of which controls all traffic lights of an intersection. $\mathcal{S}$is the set of possible global states of the entire

TABLE I
NOTATIONS

Notation	Meaning
RL	Reinforcement Learning
TSC	Traffic Signal Control
ATSC	Arterial traffic signal control
DRL	Deep Reinforcement Learning
MARL	Multi-agent Reinforcement Learning
DNNs	Deep Neural Networks
A2C	Advantage Actor-critic
PPO	Proximal Policy Optimization
IPPO	Independent Proximal Policy Optimization
Dec-POMDP	Decentralized Partially-observable Markov Decision Process
A_i	Action space of agent i
N	The number of agents or intersections
s_t	Global state at time step t
o_t^i	The observation of agent i at time step t
a_t^i	The action of agent i at time step t
r_t^i	The reward of agent i at time step t
t_y	The yellow time
t_{min}	Minimum green time
t_{max}	Maximum green time
$phase_t^i$	Current phase of intersection i at time step t
$\text{vehs}_t^i(l)$	The number of vehicles on lane l of intersection i at time step t
$waiting_t^i(l)$	Mean stop delay of vehicles on lane l of intersection i at time step t
$queue_t^i(l)$	The measured queue length along each incoming lane $l$of intersection $i$at time step t

traffic environment. At each time step t, each agent $i \in \mathcal{N}$ can only draw an individual local observation $o_t^i \in O$ (including the length of queues on incoming lanes and vehicle stop delays) from the observation kernel $\mathcal{Z}(s_t, i)$ instead of being able to observe the full state $s_t \in \mathcal{S}$. Each agent $i \in \mathcal{N}$ uses a decentralized policy $\pi_{\theta_i}\left(a_t^i | o_t^i\right)$ to select its action $a_t^i \in \mathcal{A}$(i.e., change the duration of the current signal phase) according to the local observation $o_t^i \in O$. This yields the joint action $a_t := \left\{a_t^i\right\}_{i=1}^N \in \mathcal{A}^N$. After taking the joint action, the state of the entire arterial corridor changes from s_t to s_{t+1} according to the state transition probability $P(s_{t+1}|s_t, a)$. Subsequently, the agents receive a scalar team reward $r_t = r(s_t, a_t)$ (i.e., queue length and stop delay across the arterial corridor after adjustment for phase time at all intersections). All agents work together to maximize their expected discounted return, $\mathbb{E}\left[\sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}\right]$, where $\gamma \in 0, 1)$ is a discount factor. In summary, the problem contains three constraints, namely (1) minimum and maximum green time, (2) anti-overflow, (3) local observation by sensors. Under the above constraints, the goal of each agent is to select the optimal time period from historical experience to optimize the traffic efficiency of arterial corridors. Table I shows the meaning of each notation.

IV. Deep Reinforcement Learning Algorithm for Arterial Traffic Signal Control

In this section, first, IPPO is formulated by extending the idea of IA2C to the PPO algorithm. Then, parameter sharing is applied to IPPO to mitigate the slowing of convergence due to nonstationarity. One limitation is that the observation

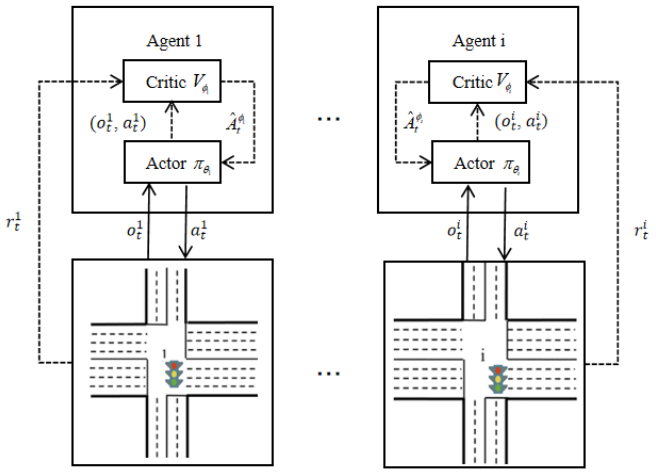


Fig. 2. Overview of the full decentralized framework of the IPPO algorithm.

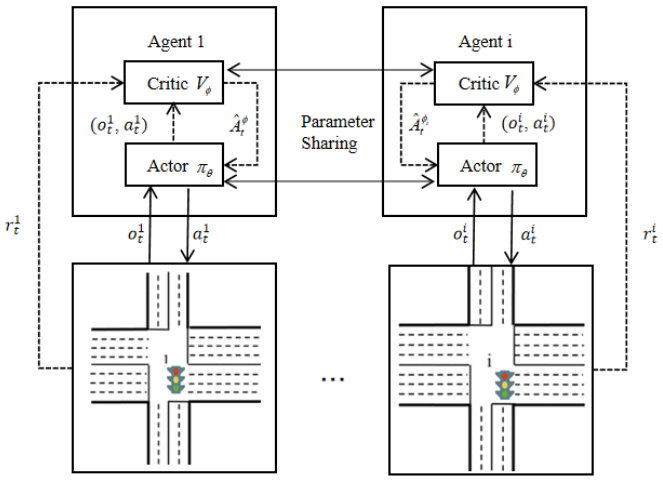


Fig. 3. Overview of the PS-PPO algorithm framework, in which the network parameters ϕ and θ are shared across critics and actors, respectively.

spaces of all agents must be the same size since there is a single neural network. However, in practice, the numbers of edges and lanes at each intersection in an arterial corridor may be different. We resolve this problem by "padding" the observations of each agent to a uniform size.

A. Multi-Agent Independent PPO Algorithm

High sample complexity and the need to carefully adjust the step size are the drawbacks of most strategy gradient algorithms [26]. Both issues are resolved by the PPO algorithm. Inspired by the IA2C algorithm and given the advantages of the PPO algorithm, we formulate the IPPO algorithm by deploying PPO algorithms independently on each agent. In IPPO, each agent needs to update its own actor and critic network. Fig. 2 shows the overall framework of the IPPO algorithm, which is a framework of decentralized training and decentralized execution. The training process is represented by the dotted lines in Fig. 2. The global information, including s_t and a_t, is not available for training and execution. In practical implementation, the input of the actor network is o_t^i, which is the partial state of states$_t$. After receiving a local observationo_t^i, each actor chooses their action a_t^i according to their own current policy $\pi_{\theta_i}\left(a_t^i|o_t^i\right)$ and obtains feedback and a new observation o_{t+1}^i from the environment. We use a truncated version of generalized advantage estimation (GAE) [27] based on independent learning because it can compromise between variance and bias. Each agent learns a local observation from critic $V_{\phi_i}\left(o_t^i\right)$ parameterized by ϕ_i, and GAE is used to estimate the advantage function for each agent i's trajectory element,

$$\hat{A}_t^{\phi_i} = \delta_t^i + (\gamma\,\lambda)\,\delta_{t+1}^i + \cdots + (\gamma\,\lambda)^{T-t+1}\,\delta_{T+1}^i, \tag{1}$$

where $\delta_t^i = r_t^i + \gamma\,V_{\phi_i}\left(o_{t+1}^i\right) - V_{\phi_i}\left(o_t^i\right)$ is the Temporal Difference (TD) error at time step t. Each agent's independent policy updates are clipped based on the objective:

$$J\left(\theta_i\right) = \mathop{\mathbb{E}}_{o,a\sim\pi_{\theta_k}}\Big[min\left(l_t^i\left(\theta_i\right)\hat{A}_t^{\phi_i},\right.$$
$$\left.clip\left(l_t^i\left(\theta_i\right),1-\varepsilon,1+\varepsilon\right)\hat{A}_t^{\phi_i}\right)\Big], \tag{2}$$

where $l_t^i\left(\theta_i\right) = \frac{\pi_{\theta_i}(a_t|o_t,i)}{\pi_{\theta_{i,old}}(a_t|o_t,i)}$ is the likelihood ratio of the new policy π_{θ_i} to the old policy $\pi_{\theta_{i,old}}$, $clip\left(l_t\left(\theta\right),1-\varepsilon,1+\varepsilon\right)$ clips $l_t\left(\theta\right)$ in the interval $[1-\varepsilon,1+\varepsilon]$ to remove incentives for the policy to change dramatically, and ε is a small hyperparameter that roughly represents the gap between the new and old policies.

The policies are updated via stochastic gradient ascent with the Adam algorithm. Similarly, the critic parameter ϕ_i of each critic is updated by minimizing the loss:

$$L\left(\phi_i\right) = \mathop{\mathbb{E}}_{o,a\sim\pi_{\theta_k}}\left[\left(V_{\phi_i}\left(o_t^i\right) - \sum\nolimits_{t'=t}^{T}\gamma^{\,t'-t}r_{t'}^i\right)^2\right]. \tag{3}$$

During training, this method requires updating an actor network parameterized by θ_i and a critic network parameterized by ϕ_i for each agent, which leads to huge computational and memory burdens in MARL tasks. Moreover, each agent adjusts its own policy dynamically and the stored experience rapidly becomes obsolete, so the environment becomes dynamic and non-stationary. In order to solve these problems, the PS-PPO algorithm is proposed.

B. Multi-Agent Parameter-Sharing PPO Algorithm

Unlike IPPO, the network parameters ϕ and θ are shared among critics and actors, respectively. It was demonstrated in [28] that for any partially observable Markov decision process (POMDP) with disjoint observation spaces, there exists a single (shared) policy $\pi_\theta^* : (\cup_{i\in\mathcal{N}}O_i)\times(\cup_{i\in\mathcal{N}}\mathcal{A}_i)\rightarrow[0,1]$, which is optimal for all agents. As shown in Fig. 3, the experiences of all agents are used to jointly train the shared policy and critic network. Although the parameters are shared between agents, different agents can select different actions because each agent receives different observations, including their respective indexes according to the "agent indication" technique [9]. In the decentralized parameter-sharing training protocol, execution is decentralized but learning is not. Each agent i takes action according to the shared policy and rollouts of individual trajectory $\tau^i = \left\{o_t^i, a_t^i, r_t^i\right\}, t\in[0,T]$, where r_t^i

is the local reward for agent i. We do not need to learn an actor and critic for each agent, so we simply update the shared policy π_θ. Then, the surrogate objective Eq. (2) becomes:

$$J(\theta) = \mathop{\mathbb{E}}_{o,a\sim\pi_{\theta_k}}\Big[min\left(l_t^i(\theta)\,\hat{A}_t^i,\; clip\left(l_t^i(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_t^i\right)\Big], \tag{4}$$

where $l_t^i(\theta) = \frac{\pi_\theta(a_t|o_t,i)}{\pi_{\theta_{\text{old}}}(a_t|o_t,i)}$, and $\hat{A}_t^i$ is the GAE estimator based on the current critic $V_{\hat{\phi}}$; i.e., $\delta_t^i = r_t^i + \gamma V_{\hat{\phi}}\left(o_{t+1}^i\right) + V_{\hat{\phi}}\left(o_t^i\right)$. The shared critic also needs to be updated by minimizing the loss function:

$$L(\phi) = \mathop{\mathbb{E}}_{o,a\sim\pi_{\theta_k}}\left[\left(V_\phi\left(o_t^i\right) - \sum\nolimits_{t'=t}^{T}\gamma^{t'-t}r_{t'}^i\right)^2\right]. \tag{5}$$

All agents execute decentralized policies by using shared parameters. This method can reduce the computational burden, speed up training, and enhance stability. One limitation of the parameter-sharing training approach is that the observation spaces of all agents must be the same size since there is a single neural network. If all intersections along an arterial corridor are homogeneous, the observation spaces of each agent are the same size. If these intersections are not homogeneous, this issue can be resolved by "padding" the observations of agents to a uniform size [28]. We can similarly "pad" the action spaces to a uniform size, and agents can ignore actions outside of their "true" action space.

The pseudocode for PS-PPO is illustrated in Algorithm 1, where L refers to the total number of iterations, K refers to the maximum number of training epochs, and B refers to the minibatch size. The policy network and critic network are initialized first. In the process of collecting training data, multiple agents interact with the environment to generate trajectories, which can be used to compute reward-to-go and advantage estimates. In order to break the correlation between the data and thus stabilize the training process, we select a random mini-batch from memory buffer D with all the agent data. These sampled data are used to calculate the gradient via Adam [29] to update the policy and the critic networks.

Algorithm 1 PS-PPO

initialize θ_{old}, $\hat{\phi}$ the parameters for policy π and critic V, memory buffer D
for iteration $= 0, 1, 2, \cdots L$ **do**
  Initialize state s_0
  for an episode $t = 1, 2, \cdots, T$ **do**
    Each agent i executes action according to a shared policy $\pi_{\theta_{old}}(a_t|o_t, i)$, then obtain r_t^i, o_{t+1}^i
  end for
  Collect trajectories for each agent i: $\tau^i = \left\{o_t^i, a_t^i, r_t^i\right\}, t \in [0, \cdots, T]$
  Compute rewards-to-go $\hat{R}_t^i = \sum_{t'=t}^{T}\gamma^{t'-t}r_{t'}^i$ on τ^i for each agent
  Compute advantage estimates $\hat{A}_t^i$ on τ^i for each agent using GAE according to Eq. (1)
  Store data $\left\{o_t^i, a_t^i, \hat{R}_t^i, \hat{A}_t^i\right\}\Big|_{t=1}^{T}$ for each agent $i \in \mathcal{N}$ into D
  for epoch $k = 1, 2, \cdots, K$ **do**
    Shuffle and renumber the data's order
    for $j = 0, 1, \cdots, \frac{T}{B} - 1$ **do**
      Select B group of data D_j: $\mathrm{D}_j = \left\{\left[o_t^i, a_t^i, \hat{R}_t^i, \hat{A}_t^i\right]\Big|_{i=1}^{N}\right\}\Big|_{t=1+Bj}^{B(j+1)}$
      Adam updates the policy according to Eq. (4), i.e., $\theta = \arg\max_\theta J(\theta)$
      Adam updates the critic according to Eq. (5), i.e., $\phi = \arg\min_\phi L(\phi)$
    end for
  end for
  Update $\theta_{old} \leftarrow \theta$ and $\hat{\phi} \leftarrow \phi$
  Empty D
end for

V. PS-PPO for Arterial Traffic Signal Control

A. States and Observations

Observation is defined for each agent, which includes the current phase, $phase_t^i$, the number of vehicles that were "seen" by camera sensors, $vehs_t^i$, the mean stop delay of vehicles still within the detection area, $waiting_t^i$, and the agent index i. So, the observation is defined as follows:

$$o_t^i = \left\{phase_t^i, \text{vehs}_t^i(l), waiting_t^i(l), i\right\} \tag{6}$$

where l is each incoming lane at intersection i. In SUMO simulation software, a *laneAreaDetector* is used to collect the observation information on an area along one or multiple lanes. The global state is defined as the collection of observations at all intersections:

$$s_t = \left\{o_t^1, o_t^2, \cdots, o_t^N\right\} \in \mathcal{S}, \tag{7}$$

where $\mathcal{S}$ is the state space. The observations of all agents need to be padded to the same size.

B. Actions

Although the use of variable-phase sequences in ATSC can make the policy more flexible, it can cause frequent changes in the phase sequence and increase the possibility of traffic conflicts. To improve traffic safety, the proposed method utilizes a fixed-phase sequence. In MARL-based ATSC, the most commonly adopted phase sequence is established based on traditional four-phase sequences comprising split phasing or lag-lag left-turn phase sequence, as illustrated in Fig. 4. Under such signal control, it is legal for vehicles to turn right when safe to do so, regardless of the traffic signal. In comparison with the traditional split phasing or lag-lag left-turn signal phases, the lead-lag left-turn phase has advantages in providing maximum bandwidth and improving arterial traffic mobility [15]. Therefore, we propose an optimized sequence based on the dual-ring schema with a lead-lag left-turn phase to further increase arterial mobility. Additionally, the protected-only left-turn signal is applied in the proposed signal control method because most arterial corridors in China have a large number of pedestrian crossings, and this signal can significantly improve pedestrian safety [30].

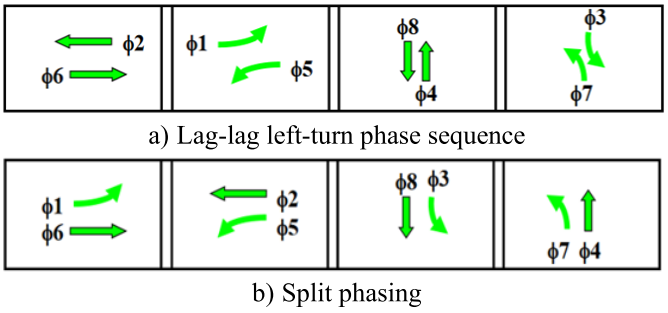


Fig. 4. Schematic diagram of two types of traditional four-phase sequences.

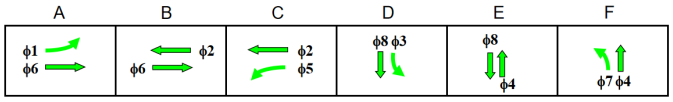


Fig. 5. Schematic diagram of a lead-lag phase sequence based on a dual-ring phase.

In the proposed model, $\phi 2$ and $\phi 6$ are the arterial coordination control phases. Then, an optimized six-phase sequence can be set, as shown in Fig. 5. The duration of each phase is variable but it cycles in the fixed order of A-B-C-D-E-F. Compared with traditional four-phase sequences, the optimized six-phase sequence can adapt to a variety of traffic flows since it increases the flexibility to adjust the release time in each direction. Therefore, each agent has the same action set:

$$A_i = \{0, 1, 2, 3, 4, 5\}. \tag{8}$$

For example, the traffic signal phase at intersection i can only be switched to the next phase if $a_t^i = 0$, and other options must extend the duration of the current phase for a_t^i seconds. In addition, considering pedestrian crossing times and drivers' maximum tolerance times, the minimal green duration t_{min} is used (15 seconds) and the maximal value t_{max} is 60 seconds. To reduce the dilemma area, the yellow-light time t_y is set to 4 seconds.

C. Reward

The main objective of our model is to improve the traffic efficiency of arterial corridors. A common metric used to indicate traffic efficiency is the total vehicle travel time. However, using the total travel time as feedback to the model may lead to delayed reward, which is unreasonable. Therefore, we propose a reward function consisting of the sum of the queue lengths of each incoming lane and the cumulative stop delay at all intersections. In particular, the possibility of queue spillback increases as the vehicle density increases on a lane, so priority should be given to lanes that are about to overflow or have overflowed. This proposed reward function can minimize the cumulative stop delay of all vehicles and avoid traffic congestion at intersections adjacent to one with spillback.

Vehicles with a speed $< v_{min}$ are considered to be stopped. Let $queue_t^i(l)$ denote the measured queue length along each incoming lane $l$of intersection $i$at time step t; i.e., the total number of vehicles with speeds $< v_{min}$. Define the vehicle density of a lane as $\rho_t^i(l) = \frac{queue_t^i(l)}{queue_{max}}$, where $queue(l)_{max}$ is the maximum permissible number of vehicles in lane l. Then, the reward of each agent i can be defined as:

$$r_t^i = -\sum_{ij\in\varepsilon, l\in L_{ij}} \left(\rho_t^i(l) \cdot queue_t^i(l) + wait_t^i(l)\right), \tag{9}$$

where $wait_t^i(l)$ is the average cumulative stop delay of all vehicles that are still inside the observed area of lane l.

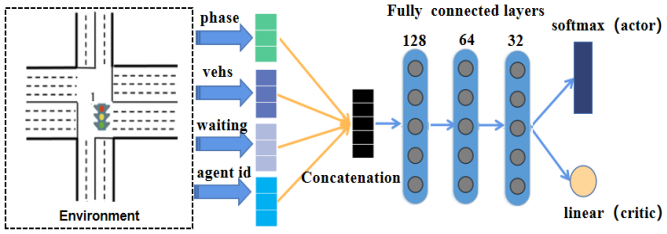


Fig. 6. Proposed DNN structure of the PS-PPO used in ATSC.

D. DNN Settings

Fig. 6 illustrates the whole network structure of each agent. The inputs of the actor network consist of the traffic observations of the local agent. We first concatenate the four features into a vector and then feed them into three MLP layers with (128, 64, 32) units and the tanh activations. The output layer for the actor is to compute the action probability with a softmax activation function. For the critic, a linear function is used to compute the value. Good convergence of DNN requires proper normalization of the collected state data. Therefore, all states are normalized to the range of [0, 1] to prevent gradient explosion, and each gradient is capped at 40. To make mini-batch updating more stable, we normalize the rewards to the range of $[-1, 0]$.

VI. Simulation and Performance

In this section, the Simulation of Urban Mobility (SUMO) [31] simulator is used to implement and evaluate our ATSC algorithm in two traffic environments: a 5×1 synthetic arterial corridor and a real-world 8-intersection arterial corridor in Hangzhou city, under time-variant traffic flows. SUMO supports real-time traffic simulation in large networks. The TraCI APIs provided in SUMO enable agents to obtain traffic-state information at intersections and send commands to traffic signals to change the duration of each phase.

A. Dataset Description

1) Synthetic Arterial Corridor: The first dataset is of a synthetic arterial corridor consisting of five homogeneous intersections (5×1), as shown in Fig. 7. Each edge has four lanes. Vehicles can only turn right in the right-most lane, go straight in the middle two lanes, and turn left in the left-most lane. The lengths of the rim edges and adjacent intersection spacing are 600 meters. In order to make a fair comparison of different methods, it is critical to simulate a realistic environment. Instead of generating constant traffic

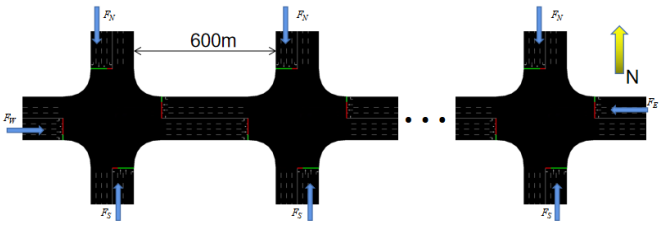


Fig. 7. Synthetic arterial corridor with five intersections. The middle two intersections are omitted due to limited space.

TABLE II
TRAFFIC VOLUME IN THE SYNTHETIC ARTERIAL CORRIDOR

Flow direction	1–300	301–600	601–900	901–1200	1201–1500	1501–1800	(sec)
F_W	1000	1400	1600	2000	1600	1000	(veh/h)
F_E	600	800	1000	1200	800	600	(veh/h)
F_N	400	600	800	600	500	300	(veh/h)
F_S	300	500	800	1000	800	600	(veh/h)

Fig. 8. Real-word arterial corridor with eight intersections in Hangzhou.

flows, four groups of time-varying traffic flows were simulated. F_N represents the traffic flow from each northern incoming edge, with other directions denotes by subscripts S, W, and E. For each incoming edge, the number of vehicles generated is as shown in Table II. The turning ratio is fixed at 60% going straight, 10% turning left, and 30% turning right.

2) Real-World Arterial Corridor: We also conducted experiments on a real-world arterial corridor in Hangzhou, China, with eight heterogeneous intersections (see Fig. 8, which was imported from OpenStreetMap). Wenyi Road is located in the main urban area of Hangzhou, and traffic jams often occur, especially during the morning and evening peak periods. Therefore, we selected traffic data from the morning peak (8:00–8:30) to validate the proposed method and compare it with existing methods. Traffic flow data was derived from camera data, including the camera ID, time, and vehicle information. Due to the low quality of real-world data, we used the number of vehicles visible in each lane in the video as the traffic flow for simplification. The traffic statistics for the morning peak period are shown in Table III. The turning ratio was set at fixed values similar to the real-world data, with 30% turning right, 10% turning left, and 60% going straight.

TABLE III
STATISTICS OF A REAL-WORD ARTERIAL CORRIDOR

Time period	Arrival rate (veh/300 s)			
	Mean	Std	Max	Min
Morning peak (8:00–8:30)	530.44	48.65	612	456

B. Experimental Settings

1) Traffic Parameters: In the simulation, we set the speed limit to 16.7 m/s. The maximum acceleration and deceleration of the vehicles were 2.6 m/s^2 and 4.5 m/s^2, respectively. The length of all vehicles was unified to 5 meters, and the distance between two vehicles was at least 3.5 meters. The IDM following model provided in SUMO was used to ensure that vehicles could drive safely on the road. v_{min} was set to 0.1 m/s.

2) Compared Methods: The performance of the proposed model was compared with those of two categories of baseline methods: traditional methods and RL methods. In order to make a fair comparison, the traditional methods also apply the lead-lag left-turn phase sequence. The state, action, and reward definitions of all RL baseline methods were the same, as defined in Section V.

a) Conventional methods:

- GreenWave (GW) [23]: This is the most common traditional method used to achieve arterial coordination control. It can theoretically alleviate congestion on arterial corridors. It first uses Webster's theory to calculate the cycle length of each intersection and then determines the optimal public cycle length shared by all intersections. The offset between intersections is equal to the ratio of the distance between adjacent intersections to the free-flow speed.
- MaxPressure (MP) [19]: The MP controller is a network-level adaptive control method that has advantages over other traditional methods. Each intersection calculates a pressure based on the queues in adjacent links, then selects the stage with the highest pressure.
- Longest-queue-first (LQF) [47]: This method gives priority to the direction with the longest queue at each intersection. At an intersection, the queue length of the lane determines the traffic release order.
- Maximal-weight-matching (MWM) [48]: This method can enable a switch to deliver throughput by using a quantitative differentiation based on queue occupancy or waiting time. The shortest queue length can obtain the maximum weight as a reward.

b) RL methods:

- Independent Deep Q-learning (IDQN) [32]: Traffic signals in a multi-intersection system are controlled by decentralized RL agents. Specifically, each agent updates

their own DQN networks independently without the exchange of information.

- Multi-agent advantage actor critic (MA2C) [11]: This approach extends IQL ideas to A2C algorithms and allows adjacent intersections to share traffic information through limited communication, thus stabilizing training.

3) Model Parameters: For DNN optimization, we use ADAptive Moment estimation (Adam) [29] as the gradient optimizer with a learning rate $\alpha = 5 \times 10^{-4}$. In each iteration, we collect 3600 samples as two episodes and 1800 steps as one episode. After training, 10 episodes are simulated to evaluate the policies. For IPPO and PS-PPO, we set the clipping parameter $\varepsilon = 0.3$, discounting factor $\gamma = 0.99$, generalized advantage estimate parameter$\lambda = 0.97$, and the minibatch size $B = 128$.

4) Evaluation Metrics: The performance of different methods is evaluated by the following metrics.

- Reward: Average reward over all evaluation episodes. The reward function is defined in Eq. (8), which is always negative. The bigger the reward, the better the performance of the method.
- Average queue length (veh): average queue length over time, where the queue length at time step t is the average number of stop vehicles on all incoming lanes. The shorter the average queue length, the fewer cars waiting on all lanes.
- Average travel time (sec): This is calculated by dividing the cumulative travel time of all vehicles by the number of vehicles in an episode.
- Average vehicle speed (m/s): This is calculated by dividing the cumulative average speed of all vehicles over one horizon by the number of vehicles in an episode. A higher average speed means smoother traffic.
- Throughput efficiency: This refers to the ratio of the number of vehicles that have arrived to the number of vehicles that have departed on the horizon. A higher throughput efficiency means higher traffic efficiency.
- Bandwidth efficiency: The bandwidth efficiency of a direction is the percent of green duration used for progression. While bandwidth generally increases with an increase in cycle length, efficiency may increase, decrease, or remain constant.

In summary, a higher reward, average vehicle speed, and throughput efficiency indicate a better performance. A shorter average travel time and queue length indicate that the traffic is less jammed.

C. Performance Comparison

1) Convergence Comparison With RL Baseline: Fig. 9 shows the training curves of the four MARL algorithms applied to a synthetic arterial corridor. Since the reward function is always negative, maximizing the reward means minimizing the cumulative queue length and waiting time. In general, the ideal result of the training curve is convergence as the agent learns from historical experience. As shown in Fig. 9, the training curves of all RL baseline algorithms show an upward trend and then converge. Among them,

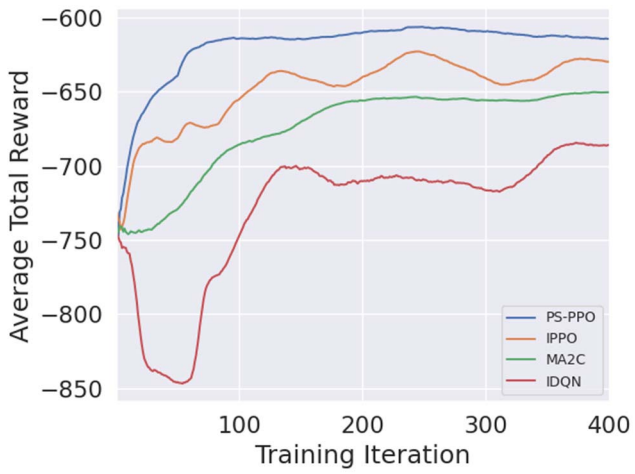


Fig. 9. Training curves of each MARL algorithm applied to the synthetic arterial corridor.

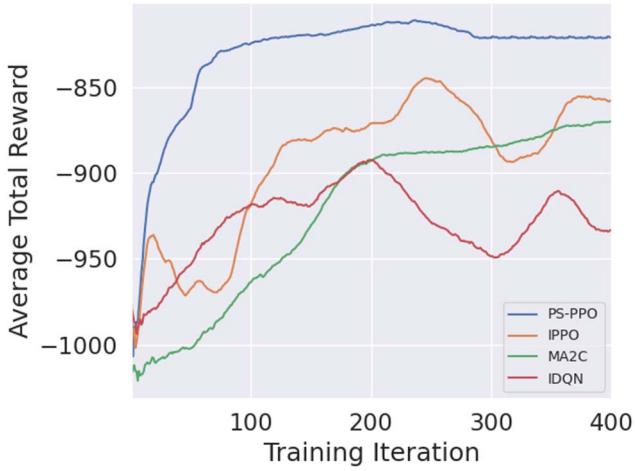


Fig. 10. Training curves of each MARL algorithm applied to the Hangzhou arterial corridor.

the training curves of IDQN and IPPO have relatively large fluctuations, but the performance of IDQN is inferior. This is because the IDQN and IPPO algorithms are similar in that each agent independently updates its own policy, which makes the environment non-stationary. Moreover, nonstationarity may cause slower and less stable training, especially in the IDQN algorithm, based on experience replay. As expected, MA2C performs better than IDQN because the neighborhood policy information is considered by each local agent, which can reduce the impact of partial observability on convergence [11]. On the other hand, PS-PPO has the best performance because it shows the fastest convergence speed, the smoothest convergence curve, and the highest reward.

On the real-world arterial corridor of Hangzhou, the training results of the four algorithms are shown in Fig. 10. The performance gaps between them are even greater than those on the synthetic arterial corridor because the Hangzhou arterial corridor is more complicated. Among them, the training curves of IDQN and IPPO have difficulty in converging. On the

TABLE IV
PERFORMANCE COMPARISON OF ALL METHODS ON SYNTHETIC AND ARTERIAL CORRIDORS

Metric	5-intersection synthetic arterial corridor							
	LQF	MWM	GW	MP	IDQN	MA2C	IPPO	PS-PPO
Reward	-786.32	-777.41	-774.02	-696.55	-686.16	-650.45	-629.34	**-614.24**
avg. travel time [s]	264.57	259.33	258.01	232.18	228.82	217.10	209.99	**204.42**
avg. vehicle speed [m/s]	6.25	6.83	6.82	7.51	7.65	7.99	8.13	**8.55**
avg. queue length [veh]	3.80	3.73	3.73	3.36	2.99	2.61	2.57	**2.22**
Throughput efficiency	0.84	0.85	0.85	0.86	0.86	0.87	0.87	**0.89**
Bandwidth efficiency	0.39	0.41	0.44	0.42	0.43	0.43	0.43	**0.45**

TABLE V
PERFORMANCE COMPARISON OF ALL METHODS ON REAL (HANGZHOU) ARTERIAL CORRIDORS

Metric	8-intersection Hangzhou arterial corridor							
	LQF	MWM	GW	MP	IDQN	MA2C	IPPO	PS-PPO
Reward	-1068.72	-999.12	-997.03	-976.89	-933.68	-870.27	-857.58	**-821.14**
avg. travel time [s]	315.61	295.75	292.35	275.21	270.75	258.03	254.89	**235.30**
avg. vehicle speed [m/s]	6.50	6.90	6.90	7.48	7.53	7.68	8.03	**8.36**
avg. queue length [veh]	3.32	3.12	3.06	2.80	2.64	2.03	1.94	**1.73**
Throughput efficiency	0.71	0.79	0.79	0.84	0.85	0.87	0.89	**0.91**
Bandwidth efficiency	0.34	0.37	0.43	0.43	0.43	0.43	0.44	**0.44**

contrary, the improved PS-PPO algorithm can still achieve the fastest convergence and the highest reward, which is stable at around −821. Compared with the other three MARL methods, the PS-PPO algorithm can achieve the best performance on both synthetic and real road networks. The reasons that PS-PPO performs better than other reinforcement learning models can be summarized as follows: 1) PS-PPO algorithm only needs to learn one strategy, which can significantly reduce the computation and memory burden, thus speeding up the convergence speed, which will be analyzed in detail in section VI(F). 2) More centralized learning process can alleviate non-stationarity in MARL and allow faster convergence [28]. Parameter sharing is the most centralized MARL method, so the theory also explains the experimental performance variation of the MADRL method: "complete" parameter sharing is better than completely independent single-agent learning (the most decentralized case). 3) Maximizing the clipped objective in PS-PPO can achieve better performance than maximizing the conventional objective in MA3C.

2) Evaluation Results: Table IV and V lists the key performance metrics of different methods, including traditional and MARL-based methods. PS-PPO outperforms other control methods in almost all metrics. In the synthetic arterial corridor, in comparison to MaxPressure, the PS-PPO algorithm provides a 33.90% shorter average queue length, 11.96% lower travel time, 13.85% higher average speed, and 3.49% higher throughput efficiency. The average performance gap between the PS-PPO algorithm and MaxPressure is even larger in the real-world scenario than in the synthetic scenario. This larger performance difference is consistent with the inherent flaw of MaxPressure, which is that it cannot learn from environmental feedback. These evaluation results fully demonstrate that PS-PPO is better than the latest traditional and MARL-based methods in alleviating congestion on arterial corridors.

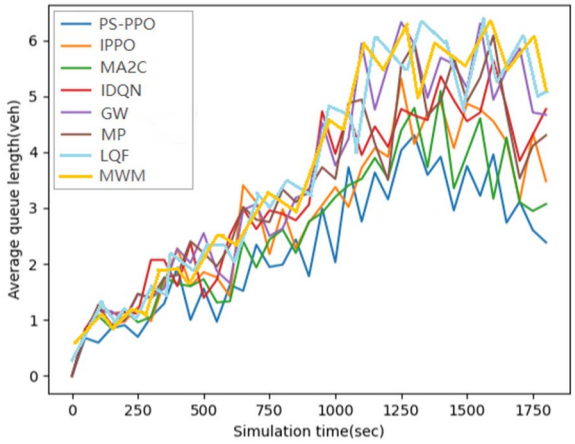


Fig. 11. Average queue length at each timestep of different methods applied to the synthetic arterial corridor.

Fig. 11 plots the average queue length in the synthetic arterial corridor at each simulation step. It is obvious that PS-PPO has the best ability to dissolve queues. In a given simulation period (1800 seconds), the flow increases from 1–1200 seconds and decreases afterward. Compared with other methods, PS-PPO can achieve faster queue dissolution and, thus, lower congestion, which indicates that it can learn more stable and sustainable policies.

D. Impact of Phase Scheme

To demonstrate the effectiveness of using an action definition with a lead-lag left-turn phase sequence, we compare

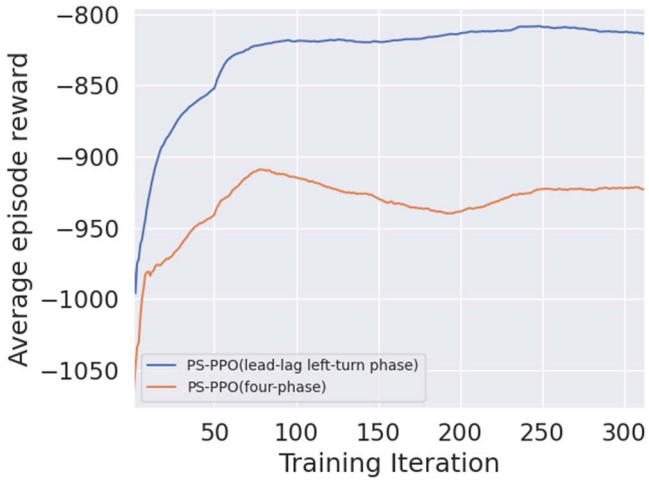


Fig. 12. Training curves of the PS-PPO algorithm using action settings of lead-lag left-turn phase or the traditional four-phase scheme on the Hangzhou arterial corridor.

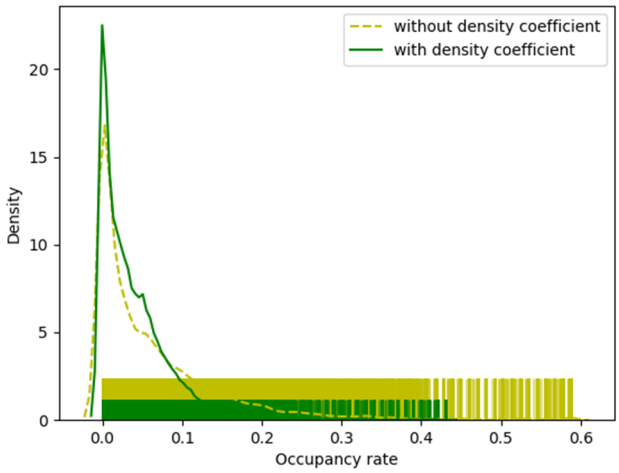


Fig. 13. Density distributions of occupancy rates in incoming lanes at all intersections in the Hangzhou arterial corridor according to PS-PPO algorithms with different reward functions.

the reward results with a model that uses the four phase sequence as shown in Fig. 4(b). Except for the action setting, the agent state definition and reward function are the same as in the proposed model. As shown in Fig. 12, the training curves of these two models with different action settings that are applied to the real-world arterial corridor show similar trends. However, using the lead-lag left-turn phase sequence provides a higher reward than the split phasing scheme. It can obtain a higher reward after the first iteration and at the end of the training. Compared with the split phasing scheme, the lead-lag phase sequence has greater flexibility in dealing with phase sequences and phase durations. This result indicates that the MARL-based arterial coordinated control system can greatly improve performance by pre-optimizing the phasing scheme when using a fixed-phase sequence in the action setting.

E. Impact of Reward Function

To prove that the defined reward function can effectively prevent the occurrence of spillback in arterial corridors, we also compared our model with a variant that uses a reward function without the vehicle density coefficient $\rho_t^i(l)$:

$$r_t^i = -\sum\nolimits_{ij\in\varepsilon, l\in L_{ij}} \left(queue_t^i(l) + wait_t^i(l)\right) \tag{10}$$

We tested the performance of this variant in the Hangzhou arterial corridor scenario and obtained the space occupancy rates on the incoming lanes at all intersections at each timestep. Fig. 13 plots the density distributions of the space occupancy rate of the variant and our method. As mentioned in section VI.B, the length of all vehicles was 5 meters and the distance between two vehicles was at least 3.5 meters. Therefore, the maximum space occupancy rate was 0.588. Fig. 13 shows that after removing the vehicle density coefficient, the space occupancy rate increased to 0.588, and the density was greater than that of our method (which includes the vehicle density coefficient) when space occupancy rates > 0.079. This result shows that our reward function can avoid spillback because the occupancy rate is always < 0.588, and the proposed reward design is effective.

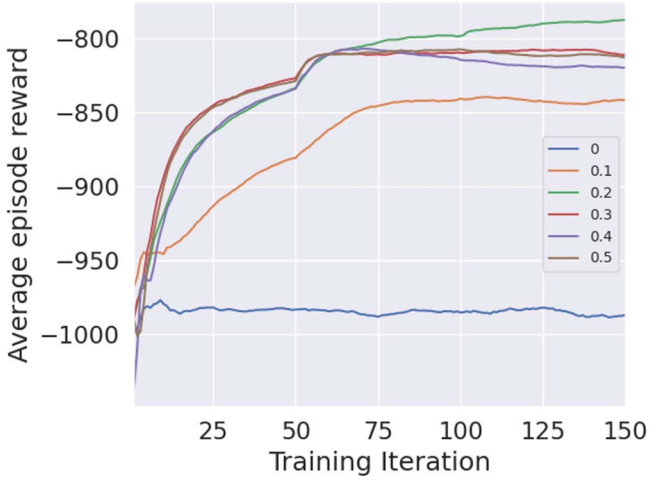


Fig. 14. Training curves of PS-PPO models with different clipping parameter ε on Hangzhou real-world arterial corridor.

F. Comparison of Surrogate Objectives in PS-PPO

We compare the performance of the surrogate objectives with several different clipping parameter ε, which is evaluated on a 8-intersection real-world arterial corridor. When $\varepsilon = 0$, the surrogate objective Eq. (4) becomes:

$$J(\theta) = \mathop{\mathbb{E}}_{o,a\sim\pi_{\theta_k}} [l_t^i(\theta)\,\hat{A}_t^i]$$

Fig. 14 shows the comparison results. Note that the reward score achieved is the lowest for the setting without clipping, which is much smaller than the other settings with clipping. Unexpectedly, when $\varepsilon = 0.2$, the PS-PPO algorithm can obtain higher reward score on the Hangzhou arterial corridor than the original setting with $\varepsilon = 0.3$.

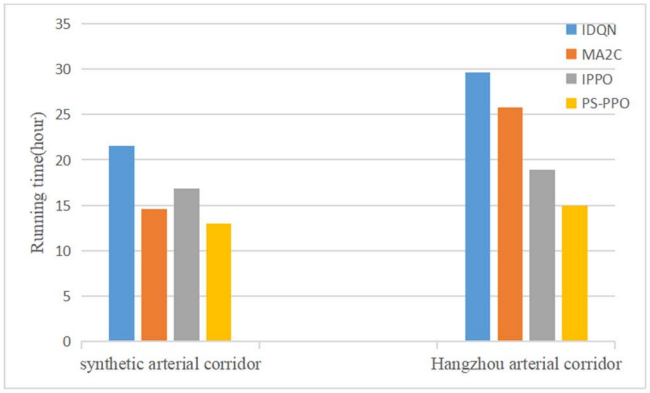


Fig. 15. Training time of four MARL methods for 300 iterations on both Synthetic and real-world arterial corridor.

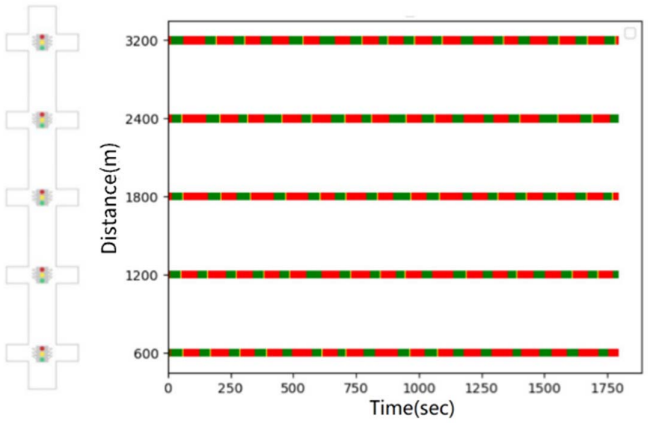
G Scalability Comparison: Whether PS-PPO is more scalable than other RL-based methods can be analyzed from the following two aspects:

1) Effectiveness: As shown in the evaluation results in Table III, and the convergence curves in Fig. 8 and Fig. 10, the performance of the PS-PPO algorithm consistently outperforms other RL methods on different scales from the 5-intersection synthetic arterial corridor to the 8-intersection real-world arterial corridor.

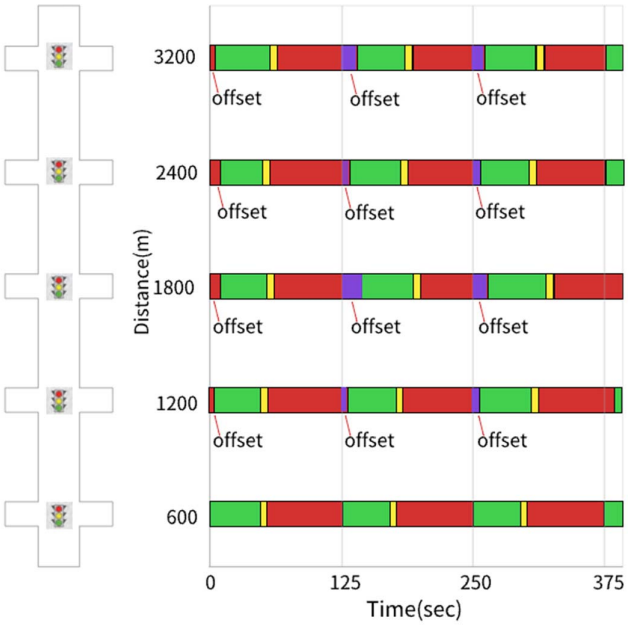
2) Training Time: We assume that the dimension of the state vector is k, and the number of trunk ports is N. Then the parameter quantity of PS-PPO is $124k+124*68+68*32+32*6$, in which everything except k is constant. Therefore, the space complexity and time complexity of PS-PPO are approximately equal to$O(k)$. However, the complexity of IDQN, MA2C and IPPO (without using parameter sharing) is approximately equal to $O(k \cdot N)$, which is infeasible when the number of intersections N is very large. We compare the training time of PS-PPO (total time for 300 iterations) with the corresponding running times of the other 3 RL methods on arterials with different numbers of intersections. For a fair comparison, all methods are evaluated individually on the same computer. As shown in Fig. 15, PS-PPO takes much less training time than IDQN, IPPO, and MA2C, which is consistent with the complexity analysis above. Therefore, PS-PPO can greatly reduce the computational requirements.

G. Policy Learned by RL Agents

Fig.16 shows the uni-directional phase plan learned by 5 agents that control each intersection on the synthetic arterial corridor. The x-axis represents time and the y-axis represents distance (the reference point is the westernmost end of the arterial corridor). It can be observed that the duration of the green light at each intersection is constantly changing, which causes the offset of adjacent intersections to be constantly changing as well. Fig. 17 is a time-space diagram showing the trajectories of all vehicles traveling straight from west to east. The x-axis represents time, the y-axis represents the position of the vehicle, and the color of the line represents vehicle speed. The black color in the picture indicates that the vehicles are waiting in a queue at an intersection. Few



(a) Uni-directional phase plan for 5 intersections



(b) Enlarged view of the first three cycles of (a)

Fig. 16. Uni-directional phase plan learned by agents for the synthetic arterial corridor.

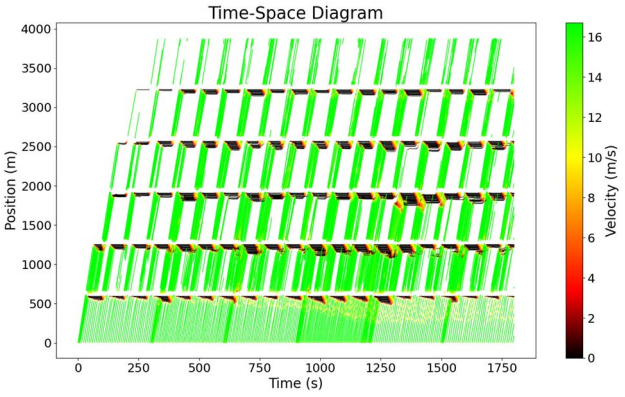


Fig. 17. Time-space diagram to illustrate the coordination strategy learned in the synthetic arterial corridor.

vehicles need to wait for the next green light phase to pass through intersections, and the green light time is rarely wasted. This result shows that PS-PPO can learn the optimal phase

split. The first three intersections on the west side have green waves that can prevent some vehicles from being stopped by the red light, which means that agents can learn a coordination policy.

VII. Conclusion

This paper focuses on using the PS-PPO algorithm for arterial traffic signal control under a partially observable scenario. MARL converges slowly under normal circumstances due to nonstationarity, and parameter sharing during learning can solve this issue in ATSC. Extensive experiments on synthetic and real-world arterial corridors show that the proposed method is effective in alleviating arterial traffic congestion and is superior to other state-of-the-art traditional and MARL-based approaches. Specifically, we designed a more effective action setting using the lead-lag left-turn phase sequence, which is more suitable for real-world applications and greatly improves the flexibility of signal control strategies for agent learning. Moreover, we designed a more comprehensive reward function to prevent overflows between adjacent intersections within a short distance. Therefore, this paper has practical significance for the intelligent control of traffic signals on arterial corridors.

The proposed method has some limitations that should be addressed before the real-world deployment. First, the phase sequence was designed manually, while automatic decision-making by the agent is more desirable. Additionally, pedestrian and vehicle classification should be considered to make the proposed method more humanized. Our future work will focus on tackling these limitations by modifying the proposed method.

References

[1] J. D. C. Little, "The synchronization of traffic signals by mixed-integer linear programming," *Oper. Res.*, vol. 14, no. 4, pp. 568–594, 1966.

[2] N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou, "A multi-band approach to arterial traffic signal optimization," *Transp. Res. B, Methodol.*, vol. 25, no. 1, pp. 55–74, 1991.

[3] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[5] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1140–1150, Sep. 2013.

[6] M. Aslani, M. S. Mesgari, and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 732–752, Dec. 2017.

[7] L. A. Prashanth 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.

[8] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, pp. 330–337.

[9] J. K. Gupta, M. Egorov, and M. Kochenderferl, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 66–83.

[10] K. J. Prabuchandran, A. N. H. Kumar, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 2529–2534.

[11] 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.

[12] X. Liang, X. Du, G. Wang, and Z. Han, "Deep reinforcement learning for traffic light control in vehicular networks," 2018, *arXiv:1803.11115*.

[13] H. Wei, G. Zheng, H. Yao, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[14] M. Aslani, M. S. Mesgari, and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 732–752, Dec. 2017.

[15] Z. Tian, V. Mangal, and H. Liu, "Effectiveness of lead–lag phasing on progression bandwidth," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2080, no. 1, pp. 22–27, Jan. 2008.

[16] J. D. C. Little, M. D. Kelson, and N. H. Gartner, "MAXBAND: A versatile program for setting signals on arteries and triangular networks," *Transp. Res. Rec.*, vol. 795, pp. 40–46, Jan. 1981.

[17] N. H. Gartner et al., "MULTIBAND—A variable-bandwidth arterial progression scheme," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1287, pp. 212–222, Jan. 1990.

[18] L. U. Kai et al., "Algebraic method of bidirectional green wave coordinated control of the end of green time," *China J. Highway Transp.*, vol. 32, no. 11, p. 202, 2019.

[19] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013.

[20] P. Varaiya, *The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections*. New York, NY, USA: Springer, 2013.

[21] L. Kuyer, W. Shimon, B. Bakker, and N. Vlassis, "Multiagent reinforcement learning for urban traffic control using coordination graphs," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2008, pp. 656–671.

[22] E. Van der Pol and F. A. Oliehoek, "Coordinated deep reinforcement learners for traffic light control," in *Proc. Learn. Inference Control Multi-Agent Syst.*, 2016, pp. 1–8.

[23] W. R. McShane and R. P. Roess, *Traffic Engineering*, 4th ed. New York, NY, USA: Pearson, 2010.

[24] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 128–135, 2010.

[25] F. A. Oliehoek and C. Amato, *A Concise Introduction to Decentralized POMDPs*. Cham, Switzerland: Springer, 2016.

[26] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. PMLR*, 2015, pp. 1889–1897.

[27] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[28] J. K. Terry, N. Grammel, S. Son, and B. Black, "Parameter sharing for heterogeneous agents in multi-agent reinforcement learning," 2020, *arXiv:2005.13625*.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[30] X. Li et al., "Impacts of changing from permissive/protected left-turn to protected-only phasing: Case study in the city of Tucson, Arizona," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2674, no. 4, p. 626, 2019.

[31] M. Behrisch et al., "SUMO-simulation of urban mobility: An overview," in *Proc. SIMUL 3rd Int. Conf. Adv. Syst. Simulation*, 2011, pp. 1–6.

[32] H. Wei et al., "PressLight: Learning max pressure control to coordinate traffic signals in arterial network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1290–1298.

[33] A. Nowé, P. Vrancx, and Y. M. D. Hauwere, "Game theory and multi-agent reinforcement learning," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 441–470.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[35] J. R. Kok and N. Vlassis, "Using the max-plus algorithm for multiagent decision making in coordination graphs," in *Robot Soccer World Cup*. Berlin, Germany: Springer, 2005, pp. 1–12.

[36] Z. Zhang, J. Yang, and H. Zha, "Integrating independent and centralized multi-agent reinforcement learning for traffic signal network optimization," 2019, *arXiv:1909.10651*.

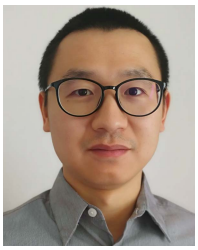
[37] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2687–2700, Jun. 2020.

[38] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*. Cham, Switzerland: Birkhäuser, 2016, pp. 47–66.

[39] N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, *arXiv:1703.09035*.
[40] G. Zheng et al., "Diagnosing reinforcement learning for traffic signal control," 2019, *arXiv:1905.04716*.
[41] X.-Y. Liu, Z. Ding, S. Borst, and A. Walid, "Deep reinforcement learning for intelligent transportation systems," 2018, *arXiv:1812.00979*.
[42] J. A. Calvo and I. Dusparic, "Heterogeneous multi-agent deep reinforcement learning for traffic lights control," in *Proc. AICS*, 2018, pp. 1–117.
[43] Y. Gong, M. Abdel-Aty, Q. Cai, and M. S. Rahman, "Decentralized network level adaptive signal control by multi-agent deep reinforcement learning," *Transp. Res. Interdiscipl. Perspect.*, vol. 1, Jun. 2019, Art. no. 100020.
[44] M. Xu, J. Wu, L. Huang, R. Zhou, T. Wang, and D. Hu, "Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning," *J. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1–10, Jan. 2020.
[45] H. Wei et al., "Colight: Learning network-level cooperation for traffic signal control," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, 2019, pp. 1913–1922.
[46] Y. Wang, T. Xu, X. Niu, C. Tan, E. Chen, and H. Xiong, "STMARL: A spatio–temporal multi-agent reinforcement learning approach for cooperative traffic light control," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2228–2242, Jun. 2022.
[47] S. Mehdian, Z. Zhou, and N. Bambos, "Longest-queue-first scheduling with intermittent sampling," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5, doi: 10.1109/PIMRC.2017.8292291.
[48] C.-B. Lin and R. Rojas-Cessa, "Maximal weight matching scheme with frame occupancy-based for input-queued packet switches," in *Proc. 17th IEEE Workshop Local Metrop. Area Netw. (LANMAN)*, May 2010, pp. 1–5, doi: 10.1109/LANMAN.2010.5507154.

Weibin Zhang received the Ph.D. degree in automation from Xi'an Jiaotong University, China, in 2008. He worked as a Research Associate with the Department of Civil and Environmental Engineering, University of Washington, Seattle, WA, USA, from 2014 to 2017. He is currently a Professor at the Nanjing University of Science and Technology. His research interests include intelligent transportation systems, data-driven transportation modeling, connected vehicle, and marine safety.

Chen Yan received the B.E. degree from the Nanjing University of Science and Technology, in 2021, where he is currently pursuing the master's degree. His research interests include traffic signal control and reinforcement learning.



Xiaofeng Li received the Ph.D. degree in civil engineering from The University of Arizona in 2021. He is a Research Assistant Professor at the Center of Applied Transportation Sciences (CATS), The University of Arizona. His major research interest is to leverage existing data sources, apply data-driven approaches to solve transportation problems in traffic monitoring, traffic signal control, public transport, shared micro-mobility, and smart cities.

Liangliang Fang received the B.E. and master's degrees from the Nanjing University of Science and Technology, in 2019 and 2022, respectively. His research interests include communication, traffic signal control, and reinforcement learning.

Yao-Jan Wu is an Associate Professor of transportation engineering at the Civil and Architectural Engineering and Mechanics Department and the Executive (Founding) Director at the Center for Applied Transportation Sciences (CATS), The University of Arizona (UA). He is currently a Faculty Advisor at the UA Institute of Transportation Engineers (ITE) Student Chapter. He has served as the Principal Investigator (PI) or Co-PI for over 40 national/international research projects. He has more than 100 refereed publications, including more than 60 journal publications. He has presented his research findings more than 100 times at national and international conferences and invited speaker events. His research interests highlight a strong connection between information technology (IT) and traditional transportation research.

Jun Li (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently a Professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a Visiting Professor at Princeton University, from 2018 to 2019. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultra-dense wireless networks, mobile edge computing, network privacy and security, and the Industrial Internet of Things.