Learning to Help Emergency Vehicles Arrive Faster: A Cooperative Vehicle-Road Scheduling Approach

Lige Ding®, Dong Zhao®, Member, IEEE, Zhaofeng Wang®, Guang Wang®, Chang Tan, Lei Fan, and Huadong Ma®, Fellow, IEEE

Abstract—The ever-increasing heavy traffic congestion potentially impedes the accessibility of emergency vehicles (EVs), resulting in detrimental impacts on critical services and even safety of people’s lives. Hence, it is significant to propose an efficient scheduling approach to help EVs arrive faster. Existing vehicle-centric scheduling approaches aim to recommend the optimal paths for EVs based on the current traffic status while the road-centric scheduling approaches aim to improve the traffic condition and assign a higher priority for EVs to pass an intersection. With the intuition that real-time vehicle-road information interaction and strategy coordination can bring more benefits, we propose LEVID, a LEarning-based cooperative Vehicle-roadD scheduling approach including a real-time route planning module and a collaborative traffic signal control module, which are closely related to each other and make decisions iteratively. The real-time route planning module adapts the artificial potential field method to address the real-time changes of traffic signals and avoids falling into a local optimum. The collaborative traffic signal control module leverages a graph attention reinforcement learning framework to extract the latent features of different intersections and abstract their interplay to learn cooperative policies. Extensive experiments based on multiple real-world datasets show that our approach outperforms the state-of-the-art baselines.

Index Terms—Cooperative vehicle-infrastructure system, emergency vehicles, deep reinforcement learning, route planning

1 INTRODUCTION

With the continual growth of population and vehicles in cities, we have been facing increasingly serious traffic congestion. Heavy traffic congestion not only causes extra air pollution and energy/time waste, but also potentially impedes the accessibility of Emergency Vehicles (EVs), such as ambulances, fire engines, and police cars, when facing unexpected accidents, resulting in detrimental impacts on critical services and even safety of people’s lives. In medical emergencies such as cardiac arrest, every one-minute delay causes mortality rate to increase by 1% and imposes additional $1542 medical cost in USA [1]. The building fires typically grow by 20% per minute, causing an average $4000 of additional damages [1]. Therefore, it is of great significance to design an efficient scheduling approach to help EVs arrive faster, especially in congested traffic conditions.

Different from Ordinary Vehicles (OVs), EVs may be exempted from some conventional road rules, such as driving through an intersection when the traffic light is red, or exceeding the speed limit. Nevertheless, EVs may still be obstructed by numerous OVs on roads with a heavy traffic. To address this issue, one research line resorts to vehicle-centric approaches, which aim at scheduling EVs with the best priority for EVs to pass an intersection. With the intuition that real-time vehicle-road information interaction and strategy coordination can bring more benefits, we propose LEVID, a LEarning-based cooperative Vehicle-roadD scheduling approach including a real-time route planning module and a collaborative traffic signal control module, which interact with each other and make decisions iteratively. The real-time route planning module adapts the artificial potential field method to address the real-time changes of traffic signals and avoids falling into a local optimum. The collaborative traffic signal control module leverages a graph attention reinforcement learning framework to extract the latent features of different intersections and abstract their interplay to learn cooperative policies. Extensive experiments based on multiple real-world datasets show that our approach outperforms the state-of-the-art baselines.

1I

© 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Digital Object Identifier no. 10.1109/TMC.2022.3188344
Recent years have witnessed the great advance in Coop-
erative Vehicle-Infrastructure Systems (CVIS), wherein the
sensing infrastructure (e.g., cameras, GPS) monitors traffic
conditions and vehicles’ locations in real time, and the
communication infrastructure enables vehicles and road infra-
structure to exchange real-time information [17]. It provides
a new opportunity to design a cooperative vehicle-road sched-
uling (CVRS) approach. On the other hand, the results of
existing methods on real-world traffic data show that only
relying on a single vehicle-centric or road-centric approach
cannot meet the fast-passing needs of EVs well (Section 3).
The above two factors motivate us to dynamically optimize
the route and concomitantly coordinate the traffic signals
along this route for better handling the dynamic traffic flow.
However, it is a very challenging task as the route planning
and traffic signal control have complex interactions as
follows:

- **Impact of frequently changing traffic signals on real-time route planning.** The real-time route planning needs to
consider not only dynamic traffic flows over the road
network but also frequently changing traffic signals.
Some cost-centric approaches [18], [19], [20] are able
to address dynamic traffic flows by a stochastic graph
based on historical data. However, the traffic signal
changes are so frequent (minute-level) that it is hard
to accurately predict the travel time costs of different
routes with only historical data.
- **Collaborative traffic signal control based on dynamic rout-
ing.** First, the dynamic route planning causes different
intersections to become upstream and downstream
intersections of EVs, which has different influences
on EVs according to the Kinematic-wave theory [21].
It is significant but difficult to extract the latent fea-
tures and dynamic influences of these intersections.
Second, multiple traffic lights should learn to cooper-
ate with each other to balance the traffic demands of
both EVs and OVs. The joint optimization may lead
to the scale expansion of the problem and increase the
computational complexity. Although extensive stud-
ies focus on traffic signal control for OVs, they cannot
well handle our problem [22], [23], [24].

To this end, we propose **LEVID**, a LEarning-based cooper-
ative Vehicle-road scheduling approach, consisting of a real-
time route planning module and a collaborative traffic signal
control module, which influence each other and make deci-
sions iteratively. The real-time route planning module
adopts the artificial potential field method to address the
real-time changes of traffic signals and avoid falling into a
local optimum by considering the long-term cumulative ben-
efit of a route. The traffic signal control module leverages a
graph attention reinforcement learning framework, which
models the traffic environment as a dynamic directed graph
to present the influences of dynamic routes and increase the
receptive field of each agent (traffic signal controller). By
employing the multi-head attention as relation kernel, this
framework is able to extract the latent features of different
intersections and abstract their interplay to learn cooperative
policies. Meanwhile, the asynchronous parameter-sharing
method is adopted to reduce the computational complexity.
Specifically, our contributions are three-fold as follows:

- We investigate the cooperative vehicle-road scheduling
paradigm for helping EVs arrive faster. To the best of
our knowledge, this is the first work to simulta-
neously optimize the route planning and traffic sig-
nal control in real time (Sections 3 and 4).
- We propose the **LEVID** approach, which considers the
long-term cumulative benefit of a dynamically
planned route and leverages graph attention rein-
forcement learning for better cooperation between
neighboring intersections (Section 5).
- We evaluate our **LEVID** using both synthetic and
real-world datasets from multiple cities. Experimen-
tal results demonstrate that our approach greatly
reduces the average travel time for both EVs and
OVs than the state-of-the-art baselines (Section 6).

## 2 RELATED WORK

### 2.1 Vehicle-Centric Scheduling

The vehicle-centric scheduling methods aim at scheduling vehicles with the best routes that can minimize the travel cost or satisfy personalized preferences. Most of studies focus on traffic signal control for OVs, which can be broadly divided into two categories, cost-centric routing [9], [10], [11], [12], [13], [25], [26] and trajectory-based routing [5], [6], [7], [8], [11], [12], [25]. Cost-centric methods mainly focus on route planning on a dynamic stochastic graph with time-dependent, uncertain edge weights [9], [10], [11], [12]. Other cost-centric studies [13], [26] take the dependencies among time distributions of different roads into account to improve the accuracy of travel time estimation. The trajectory-based studies focus on leveraging historical trajectories for path recommendation [5], [6], [7], [8]. However, these methods may not apply to EVs as the main concerns of EVs should be time sensitivity rather than personalized preferences.

Only a few early studies focus on route planning for EVs
[2], [3], [4]. Nordin et al. [2] utilize A* algorithm to determine the path with the shortest geographic distance for dispatching an ambulance to a specific ambulance station or emergency site. Chen et al. [3] calculate the evacuation time of different paths according to the length of the evacuation path and the delay at intersections, and finally select the path with the shortest evacuation time based on the Dijkstra algorithm. Barrachina et al. [4] utilize vehicular communications to accurately estimate the traffic density in a certain area and help reduce the emergency services arrival time with evolution strategies. However, these methods just avoid congested roads in a passive way, while failing to proactively improve the traffic condition to shorten the travel time of EVs.

### 2.2 Road-Centric Scheduling

The road-centric scheduling methods aim at actively improv-
ing traffic conditions by traffic signal control technologies [22],
[23]. Extensive studies focus on traffic signal control for OVs,
and the mainstream technologies have undergone an develop-
ment from rule-based methods to learning-based methods
[24], [27], [28], [29], [30]. The conventional traffic signal control
method Maxpressure [31], [32] measures the traffic flow in real
time and changes the current phase according to the rule-
based preset scheme. Reinforcement learning (RL) based
methods attempt to address traffic signal control problem by
interacting with the environment and learning from real-time data. Some studies leverage tabular Q-learning [33], [34] and deep reinforcement learning [27] for the traffic signal control of single intersection. For the multi-intersection traffic signal control, the centralized RL method [28], [35] models the actions of all agents jointly and negotiates the traffic signal control with centralized optimization, which is computationally expensive. While the decentralized RL method makes its decision based on observation of each independent agent. Some methods [30], [36], [37], [38], [39] handle the non-stationary impacts of other agents in complicated environment with exquisite reward design, which requires more human expert experience. In contrast, other methods [24], [40], [41], [42] add neighbors’ traffic condition into observation, enabling agents to behave as a group and form coordination. However, these methods do not consider the priority of EVs.

By contrast, the existing traffic signal control studies for EVs are still limited to the rule-based methods [14], [15], [16]. Kang et al. [14] propose an EV signal coordination approach to provide “green wave” for EVs. Younes et al. [15] design a real-time dynamic traffic signal control method which can handle the presence of one or more EVs over the road networks. Rosayyan et al. [16] leverage a global navigation satellite system based on geo-fencing techniques to identify the entry of EVs and provide green signal automatically. However, these rule-based methods rarely consider the impact of scheduling strategy on OVs and cannot interact with the environment in real-time.

2.3 Cooperative Vehicle-Road Scheduling

The cooperative vehicle-road scheduling provides route planning and traffic light control for EVs simultaneously. Djehel et al. [43] design a traffic signal controller, which finds the quickest path for an EV in advance, and utilizes RFID to give priority to an EV as soon as it approaches the traffic lights on this route. Karmakar et al. [44] determine the signal lights to be green based on current traffic condition and calculate the priority levels of different EVs based on the type and the severity of an incident in case of the conflict between EVs. This work also considers the impact on the traffic in the neighboring roads surrounding the EV’s travel route. However, these methods are mainly based on pre-set fixed rules and simplified assumptions. They cannot be updated synchronously as the real-time dynamic traffic flow changes [45].

3 Motivation

In this section, we first introduce the supporting devices and technologies of a CVIS, which provides opportunities for designing effective scheduling approaches. Second, we point out the defects of the existing approaches from two separated perspectives, i.e., traffic signal control and route planning, which motivates us to design a cooperative vehicle-road scheduling approach LEVID for sufficiently leveraging the ability of a CVIS.

CVIS. Fig. 1 shows the architecture of a CVIS, which consists of EVs, road infrastructure and a control center. On the EV side, a GPS module is used to collect real-time locations of an EV; a communication module is used to interact with the road infrastructure via Vehicle-to-Roadside (V2R) and Roadside-to-Vehicle (R2V) communications according to the Dedicated Short Range Communication (DSRC) standard, and interact with the control center via the cellular communications (e.g., 4G/5G). On the road infrastructure side, traffic cameras and traffic signal controllers have been widely deployed on major roads of many cities. For example, there are over 3,000 major intersections in the urban area of Hefei, China, of which 1,338 intersections have traffic signal controllers that can be adjusted by the control center. Meanwhile, there are 14,967 traffic cameras deployed at intersections and other locations such as entrances/exits of expressways and key locations along arterial roads, as partly shown in Fig. 2. The trajectories of all the vehicles are recorded when they pass through cameras, and can be extracted by the advanced vehicle identification technologies [46]. The traffic volume can also be obtained by counting the number of vehicles passing through intersections. Finally, the control center can obtain real-time locations of EVs and traffic conditions from the road infrastructure; in turn, it can provide a driving plan to the EV and determine a traffic control strategy to the traffic signal controller.

Traffic Signal Control. GreenWave [14] is the most commonly used traffic signal control approach, which allows all the traffic lights in the route to turn green so that EVs can pass intersections continuously along the emergency corridor. The “green wave” is achieved by signal coordination setting. However, if there is a traffic jam on one road segment, the signal offset time between intersections will be changed such that EVs cannot pass intersections continuously. In other words, GreenWave cannot handle a dynamic and heavy traffic flow well. To address this issue, we are working with the traffic police department in Hefei to improve the GreenWave. More specifically, since the real-time locations of an EV are available, we can adjust the traffic signal to turn green automatically whenever the
distance between EV and intersection is less than a certain threshold. Nevertheless, it is still a non-trivial task to determine a proper threshold. If the threshold is too large, it may cause the vehicles in the opposite direction to be blocked for a long time. Conversely, if the threshold is too small, it may fail to clear the way for the EV at an congested intersection. To analyze this phenomenon, we collect the trajectory data of 5,448 vehicles from traffic cameras during 9-11 a.m. on one working day in a region of Hefei (Fig. 3). We simulate the movement of an EV through an intersection. When the threshold is set as 500 m for an intersection with a low traffic pressure, the traffic flow in the opposite direction has to wait for more than 32 seconds. When the threshold is set as 200 m for an intersection with a heavy traffic pressure, the EV is blocked by queuing vehicles at the intersection for about 47 seconds. This phenomenon motivates us to design a more effective approach from two perspectives: 1) utilize a learning-based traffic signal control strategy instead of a rule-based strategy, and 2) integrate it with a route planning strategy to further reduce the waiting time at intersections with a heavy traffic.

Route Planning. To preliminarily demonstrate the importance of route planning, we generate an EV to move along different routes by data-driven simulations. As shown in Fig. 3, given the same origin and destination, if the EV moves along Route 1 with the shortest distance, the travel time is 236.7 s; if it moves along Route 2 with the least congestion, the travel time would be 214.9 s; whereas, when we further consider the real-time status of traffic signals into account, Route 3 is the best choice with the travel time of 198.5 s, as the go-straight phase in the east-west direction is allowed and the left-turn phase is forbidden at Intersection A. It implies the significance of considering both traffic conditions and changes of traffic signals for route planning.

4 PROBLEM FORMULATION

This section introduces the key definitions and Table 1 shows the mainly used notations.

Definition 1 (Road Network). The road network is defined as a directed graph \( G = (V, E) \), where \( V = \{v_1, v_2, \ldots, v_t\} \) is the set of nodes (i.e., intersections) and \( E \) is the set of edges (i.e., road segments). An edge \( e_{i,j} \in E \) represents a directed road segment from intersection \( v_i \) to intersection \( v_j \).

Definition 2 (Route). A route \( R \) connects the origin location \( v_o \) and the destination location \( v_d \) with an ordered sequence of intersections, i.e., \( R : v_o \rightarrow v_1 \rightarrow \cdots \rightarrow v_l \rightarrow \cdots \rightarrow v_d \), where each pair of consecutive locations corresponds to a road segment \( e_{i,i+1} \).

TABLE 1

| Notation | Description |
|----------|-------------|
| \( V = \{v_1, v_2, \ldots, v_t\} \) | The set of intersections |
| \( v_o \) | Origin location |
| \( v_d \) | Destination location |
| \( R : v_o \rightarrow v_1 \rightarrow \cdots \rightarrow v_l \rightarrow \cdots \rightarrow v_d \) | The route of one vehicle |
| \( l_i^{in} \) | The incoming lane at \( v_i \) |
| \( l_i^{out} \) | The outgoing lane at \( v_i \) |
| \( L^{in}_l \) | The set of incoming lanes at \( v_i \) |
| \( L^{out}_l \) | The set of outgoing lanes at \( v_i \) |
| \( g(l_i^{in}, l_i^{out}) \) | A traffic movement |
| \( ph_i \) | A movement signal |
| \( T_e \) | A phase of traffic lights |
| \( T_v \) | Travel time on road segment \( e \) |
| \( T_o \) | Waiting time at intersection \( v \) |
| \( S_v = \{d_1, \ldots, d_l\} \) | Observation of intersection \( v \) |
| \( S_t \) | Traffic condition at time \( t \) |
right-turn signals are omitted as the traffics on the right-turn lanes are usually allowed in the real world. Furthermore, there are eight mutually exclusive phases, each of which is a combination of two traffic movements. In this example, the phase #4 is activated, indicating that the traffics on the left-turn lanes from south and north are allowed to turn left.

**Definition 5 (Traffic Observation and Overall Condition).** The observation of intersection \( v_i \) at time \( t \) is denoted by \( o_i \). It includes the current phase \( ph_i(t) \), the number of OVs \( x_i^{in}(t) \) on each incoming lane \( l_i^{in} \), the number of OVs \( x_i^{out}(t) \) on each outgoing lane \( l_i^{out} \), and the number of EVs on each incoming and outgoing lane, which are denoted as \( x_i^{in}(t) \) and \( x_i^{out}(t) \). We utilize \( S_i = \{ o_i, \ldots, o_j \} \) to denote the overall condition at time \( t \).

**Definition 6 (Travel time).** Given a route \( R \) of an EV, its travel time \( T \) consists of the driving time \( T_e \) on each road segment \( e \in R \) and the waiting time \( T_v \) at each intersection \( v \in R \) to wait for the existing queued OVs to pass through the intersection. Note that, although an EV may be exempted from some conventional road rules, such as driving through an intersection when the traffic light is red, or exceeding the speed limit, it may still be obstructed by OVs on roads with a heavy traffic. We denote the travel time of one EV by

\[
T(R) = \sum_{e \in R} T_e + \sum_{v \in R} T_v. \tag{1}
\]

**Problem State.** Given the origin location \( v_o \) and the destination location \( v_d \) of an EV and the overall dynamic traffic condition \( S_t \) at each time step \( t \), a real-time route planning strategy \( \varphi_1 \) is utilized to determine a driving route \( R \). Meanwhile, given the observation \( o_i \) of each intersection \( v_i \) at each time step \( t \), a collaborative traffic signal control strategy \( \varphi_2 \) is utilized to choose a control action (i.e., which phase to set). The objective of this work is expressed as follows:

\[
\varphi^* = \arg \min_{\varphi_1, \varphi_2} T(R|v_o, v_d, S_t). \tag{2}
\]

5 **DESIGN OF LEVID**

5.1 Framework

As shown in Fig. 5, LEVID contains a real-time route planning module and a collaborative traffic signal control module, which influence each other and make decisions iteratively.

- **Real-time route planning module** adapts the artificial potential field method [47] by modeling the estimated travel time as the repulsion and the trend of an EV moving towards the destination as the gravity. Furthermore, it introduces the long-term repulsion to handle the changing traffic lights and avoid falling into a local optimum. At every time interval \( \Delta \), the route with the maximum long-term cumulative benefit is selected according to the current traffic signal phases and traffic condition near the EV. Meanwhile, the length of any candidate route is limited to reduce search depth and computational complexity.

- **Traffic signal control module** models the traffic environment as a dynamic directed graph and adjusts the relational distance between intersections according to the dynamically updated route and upstream/downstream relationships. The receptive field of each agent contains its top-K relevant neighboring intersections to differentiate valuable local information from global information. The observed features of the top-K relevant intersections are transformed into hidden features with Multi-Layer Perceptron (MLP). Then the multi-head attention is employed as relation kernel to extract the latent features of different intersections and abstract their interplay to learn cooperative policies. Finally, the long-term impacts of different traffic signal phases are evaluated by the centralized critic model whose parameters are shared by all distributed actors (traffic signal controllers).

5.2 Real-Time Route Planning Module

The detailed calculation process of gravity and long-term repulsion is introduced in the following part.
Gravity. The gravity indicates the trend of an EV moving towards the destination. The greater the gravity, the faster the EV can reach the destination. Suppose an EV is arriving at the current intersection \(v_c\) and will go to the final destination intersection \(v_d\). First, we define the road network distance \(\text{dis}\) as the length of the shortest path between two points in the road network. Then the gravity of \(v_i\)'s neighbor \(v_j\) to the EV is calculated as

\[
F_g(v_c, v_i) = \frac{(\text{dis}(v_c, v_d) - \text{dis}(v_i, v_d))}{S_{c,i}},
\]

(3)

where \(\text{dis}(v_c, v_d)\) denotes the road network distance between \(v_c\) and \(v_d\), \(\text{dis}(v_i, v_d)\) denotes the road network distance between \(v_i\) and \(v_d\), and \(S_{c,i}\) is the real-time average traffic speed on the road \(e_{c,i}\). Please note that \(\text{dis}(v_c, v_i)\) is not always equal to \(\text{dis}(v_i, v_d) - \text{dis}(v_i, v_d)\). The road network distance \(\text{dis}(v_i, v_j)\) is always positive, while \(\text{dis}(v_i, v_d) - \text{dis}(v_i, v_d)\) can be either positive or negative, because the intersection \(v_i\) that is chosen to drive to may be farther from the destination \(v_d\).

When the EV chooses to take a long detour to avoid a congested road segment, resulting in additional travel distance, the latter value is calculated as a negative value to indicate a possible penalty for this route planning.

Immediate Repulsion. The immediate repulsion represents the estimated travel time of a candidate route. It contains the driving time of an EV on the road segments and the waiting time at the intersections along the route. Suppose an EV is arriving at the current intersection \(v_c\). Then the immediate repulsion of \(v_i\)'s neighbor \(v_j\) to the EV contains time \(t_r(v_c, v_i)\) to wait at the intersection \(v_i\) and driving time \(t_r(v_j, v_i)\) on the road segment \(e_{c,i}\), calculated as follows:

\[
F_r(v_j, v_i) = t_r(v_c, v_i) + t_w(v_i, v_j)
= \frac{\text{dis}(v_i, v_d) - \text{len}(e_{c,i})}{S_{c,i}} + \frac{\text{len}(e_{c,i})}{S'},
\]

(4)

where \(\text{len}(e_{c,i})\) denotes the length of the queue about to drive from intersection \(v_c\) to intersection \(v_i\). \(S_{c,i}\) is the real-time average traffic speed on the road \(e_{c,i}\) and \(S'\) denotes the maximum speed for vehicles passing through an intersection allowed by law.

Long-Term Repulsion. Long-term repulsion helps approximate the long-term cumulative benefit of one route. Some routes with less immediate repulsion may guide vehicles to move to a congested road segment due to short-term shortcomings. Therefore, we expand the search depth and calculate the long-term repulsion \(F'_{r}(v_i, v_j)\) along different routes with a discounted factor \(\lambda\) as follows:

\[
F'_{r}(v_i, v_j) = F_r(v_c, v_i) + \lambda \min_{v_j \in N_i} F'_{r}(v_i, v_j),
\]

(5)

where \(N_i\) denotes the set of \(v_i\)'s neighbors, and this iterative calculation will stop when the search depth \(num\) reaches the maximum search depth limit \(\text{Dep}\). The repulsion is approximated based on the current traffic condition and it may have changed when an EV travels to the relevant road segment far away from the current location. The greater the distance between intersections, the larger the error of the estimated long-term impact. Therefore, a smaller discounted factor will be assigned to a farther intersection. We limit the depth of search space and calculate the long-term discount repulsion according to Eq. (5).

Fig. 6 illustrates the detailed process of route planning. The orange circle denotes the current location of one EV. There are three candidate routes, i.e., \(R_1\)-red, \(R_2\)-green, \(R_3\)-yellow. Based on a specific traffic condition, the gravity of intersection \(v_i\) towards EV is \(-3\) as this intersection will lead the EV to move away from the destination. The green numbers show the immediate repulsion of each road segment in route \(R_2\). Taking route \(R_3\) as an example, we introduce the discount factor. As the distance increases, the discount factor decreases exponentially. We limit the depth of search space to 4 and set the discount factor \(\lambda\) as 0.8 in this example to show the detailed process. The complete algorithm is shown in Algorithm 1.

Algorithm 1. The Real-Time Route Planning Module

1. Input: Current location \(v_c\) of EV, destination \(v_d\), the state of each road segment.
2. Initialize the discount factor \(\lambda\), and the depth of search space \(\text{Dep}\).
3. for \(v_i \in N_c\), do
   4. /*Calculate \(v_i\)'s gravity towards EV*/
      \[F_g(v_i, v_j) = \frac{(\text{dis}(v_i, v_d) - \text{dis}(v_i, v_d))}{S_{c,i}};\]
   5. /*Calculate \(v_i\)'s immediate repulsion*/
      \[F_r(v_i, v_j) = t_r(v_i, v_j) + t_w(v_i, v_j);\]
   6. /*Calculate \(v_i\)'s long-term repulsion*/
      \[F'_{r}(v_i, v_j) = F_r(v_i, v_j) - \lambda \min_{v_j \in N_i} F'_{r}(v_i, v_j);\]
   7. /*Calculate long-term cumulative benefit*/
      \[B(v_i, v_j) = F_g(v_i, v_j) - F'_{r}(v_i, v_j);\]
4. end
5. Output: The route with the maximum long-term cumulative benefit \(B(v_i, v_j)\)

5.3 Traffic Signal Control Module

The control of traffic signals can be formulated as decentralized partially observable markov decision process, where each agent chooses its phase action based on local observation \(o'\) at each time interval \(\Delta t\).

5.3.1 Agent Design

The state, action and reward for an agent which controls the signal of one intersection are as follows:

State (Observation). The State is equal to the overall traffic condition \(S_t = \{o'_1, \ldots, o'_T\}\). Correspondingly, the observation
of one agent in multi-agent RL equals to the observation $o_t^j$ of the intersection $v_i$.

**Action.** At time $t$, each agent chooses different legal available phase set according to the structure of road network and traffic demand. In our problem, we consider four phases (WE-Straight, NS-Straight, WE-Left and NS-Left) for an intersection.

**Reward.** The traffic light control method should consider both OVs and EVs. Therefore, we design the reward with an evaluation mechanism which considers these two types of traffic demands. We utilize the pressures to help OVs go through intersections more smoothly. As for EVs, they need to pass as soon as possible. Thus we leverage the queue length to measure the benefit of one action. The pressure [29], [32] of a movement for OVs is defined as the difference of OV density between the entering lanes and the exiting lanes. The pressure $P_o(i)$ of intersection $v_i$ for OVs is the sum of absolute pressures over all traffic movements, which can be defined as

$$P_o(i) = \sum_{l_p \in \text{in}} \left| x_o(l_p) - x_o(l_p) \right|,$$

where $x_o(l)$ is the number of OVs on an entering lane $l$ and $x_o(l')$ is the number of OVs on an exiting lane $l'$. What's more, considering the traffic priority of different types of vehicles, we utilize their proportions in the traffic flow to assign the weights in the reward function. We define the reward $r_i$ as

$$r_i = -\frac{L_e(i)}{\eta} \frac{P_o(i)}{1 - \eta},$$

where $L_e(i)$ is the number of EVs on the entering lanes of intersection $v_i$ and $\eta$ is the proportion of EVs in all vehicles at this intersection.

### 5.3.2 Dynamic Directed Graph

The dynamic directed graph helps capture the dynamic impacts of neighboring intersections due to real-time route planning. We construct the road network as a graph in which the weight of each edge is calculated as the corresponding real-time road network distance $\text{dis}$ between two intersections. Then, we get the top-$K$ relevant neighboring intersections $K_i$ of intersection $v_i$ based on the dynamic relational distance. Dynamic relational distance helps the current intersection pay more attention to the traffic flow at the upstream intersection when an EV will come from the upstream of the current intersection. Specifically, according to the planned route, the relational distance between upstream intersections and current intersection is set smaller by assigning a relational factor $\delta$ to these intersections. For an edge $e_{i,i+1}$ in the route of EV $R_i$, the relational distance $\text{dis}_r(i, i+1)$ from intersection $v_i$ to $v_{i+1}$ is calculated as

$$\text{dis}_r(i, i+1) = \text{dis}(i, i+1) \cdot \delta,$$

where $\text{dis}(i, i+1)$ is the road network distance from intersection $v_i$ to $v_{i+1}$ and $\delta$ is the relational factor.

Fig. 7 illustrates the top-$K$ relevant neighboring intersections and the corresponding relational distance to the current intersection based on different routes of the EV. We set the road network distance between adjacent intersections as 1, the discount factor $\delta = 0.5$ and $K = 6$. Please note that the current intersection itself is also included in the top-$K$ neighbors.

### 5.3.3 Multi-Head Attention Relation Kernel

The $m$-dimensional observation data $o_i$ of intersection $v_i$ are transformed into the $n$-dimensional hidden features $H_i$ via an MLP:

$$H_i = \sigma(o_i W_o + b_o),$$

where $W_o$ and $b_o$ are the weight matrix and bias vector respectively. Then we embed the representation of the current intersection $v_i$ and neighbor $v_j$ from the previous layer to get different types of importance score $e_{ij}$ of one neighbor. Specifically, we utilize the multi-head attention mechanism where attention functions with different linear projections are performed in parallel to jointly attend to a neighbor from different representation subspaces with the following operations:

$$e_{ij}^h = (H_i W_{oh}) \cdot (H_j W_{jh})^T,$$

where $h \in \{1, 2, \ldots, H\}$ is the index of different representation subspaces and $W_{oh}$ is the weight matrix to be trained in this subspace. This operation helps find the similarity and correlation of the features of two intersections in different subspaces, and provides the importance score under this subspace. $e_{ij}^h$ is the importance score of neighbor $v_j$ to current intersection $v_i$ in the subspace $h$. Please note that $e_{ij}$ is usually different from $e_{ij}^h$ due to dynamically updated route planning. We retrieve the general attention score between neighbors and the current intersection by normalizing the importance score of different neighbors in the same subspace

$$a_{ij}^h = \text{softmax}(e_{ij}^h) = \frac{\exp(e_{ij}^h / \mu)}{\sum_{j \in \mathcal{K}_i} \exp(e_{ij}^h / \mu)},$$

where $\mu$ is the temperature factor and $\mathcal{K}_i$ is the top-$K$ relevant neighboring intersections of intersection $v_i$. Finally we get $h_{m_i}$, the overall influence of neighbors to the current intersection in different subspaces by combining the hidden feature representations $H_i$ of all the top-$K$ relevant neighbors with their respective general attention scores $a_{ij}^h$:

$$h_{m_i} = \sigma \left( W_\gamma \left( \frac{1}{H} \sum_{h=1}^H \sum_{j \in \mathcal{K}_i} a_{ij}^h (H_j W_{oh}) + b_q \right) \right).$$

where $W_\gamma$ and $b_q$ are the weight matrix and bias vector in the final FC layer respectively. We utilize $W_{oh}$, the weight matrix.
associated with \( H_j \), to process the features of each neighbor intersection \( v_j \in K_i \). After that, we calculate their influence on current intersection \( v_i \) in each subspace \( h \) according to the attention scores \( a_{ij}^h \) obtained above. The averaging operation of multi-head attention is one of the most feasible ways to conclude the neighborhood cooperation.

5.3.4 Centralized Critic Model

The key idea of RL is to utilize Bellman equation to approximate the long-term discounted cumulative benefit of an action, which is significant for the transportation system with strong spatio-temporal correlations. The long-term impact \( \mathcal{R} \) of a signal control action is defined as follows:

\[
\mathcal{R} = \sum_{t=0}^{T} y^t r(o_t^i, a_t^i),
\]

where \( r(o_t^i, a_t^i) \) is the immediate reward of action \( a_t^i \) based on the observation \( o_t^i \) at intersection \( v_i \). Based on the processed real-time observation information \( h_{m_i} \), we leverage the deep RL to estimate the expected reward of the given state-action pair \((o_t^i, a_t^i)\) as \( Q(o_t^i, a_t^i|\theta) \), which can be calculated as

\[
Q(o_t^i, a_t^i) = h_{m_i}W_p + b_p,
\]

where \( W_p \in \mathbb{R}^{\exp} \) and \( b_p \) are the training parameters, \( p \) is the number of phases (action space) and \( \theta \) represents all the trainable variables in our centralized critic model. The phase action with the maximum long-term reward will be chosen. We optimize our control policy by minimizing the loss function as follows:

\[
L(\theta) = \sum_{t=1}^{T} \sum_{i=1}^{I} (Q(o_t^i, a_t^i|\theta) - y_t)^2,
\]

where \( T \) is the number of time steps, \( I \) is the number of intersections and \( y_t \) is the target Q value defined as

\[
y_t = r_t^i + \gamma \max_{a_{t+1}} Q(o_{t+1}^i, a_{t+1}^i).
\]

5.4 LEVID Algorithm

We present the specific process of the LEVID and how modules interact with each other in Algorithm 2. First LEVID dynamically chooses the optimal route with the maximum long-term cumulative benefit \( B(v_t, v_j) \) for one EV using Algorithm 1. Second, it updates the dynamic directed graph based on the route of EV, according to Eq. (8). Then the multi-head attention relation kernel captures the features of different neighboring intersections. Finally the optimal action with the maximum \( Q \) value is selected by Eq. (14).

5.5 Complexity Analysis

In this subsection, we analyze the scalability of LEVID_Dy, namely the RL part of LEVID. Specifically, we suppose our model gets the \( m \)-dimensional input data and each layer has \( n \) neurons; the scale of traffic signal phase space is \( p \). The time and space complexities are analyzed based on the following assumptions: (a) all the distributed actors leverage the centralized critic model to predict the long-term discounted reward of a traffic signal action; (b) each target intersection gets the top-\( K \) relevant neighbors based on a breadth first search with a total search number of \( 2K \), as excessive search range may cause unnecessary computational consumption; (c) the multi-head attentions are computed independently with the same time consumption as that of single-head attention, and the embedding process of either source or target intersection can be executed simultaneously; (d) all the actors can execute the prediction process independently. Then the time complexity in each component is: (a) top-\( K \) search: \( O(2K + K\log K) \); (b) MLP: \( O(mn) \); (c) Graph Attentional layers: \( O : (n^2 + n^2) \); (d) Q-value Prediction layer: \( O(np) \). And the total time complexity is \( O(n(m + 2n + p) + K) \), which is approximately equal to \( O(n^2) \).

Algorithm 2. The LEVID Algorithm

1 Input: The current location \( v_i \) and the destination \( v_d \) of each EV, the state of each road segment, and the overall traffic condition \( S_t \).
2 for \( \Delta t \in T \) do
3 for each EV \( v \) do
4 Choose route \( R \) with the maximum long-term cumulative benefit \( B(v_i, v_j) \) using Algorithm 1;
5 end
6 /*Update the dynamic directed graph*/
7 for edge \( e_{i,j+1} \in \bar{R} \) do
8 \( dis(i, i+1) = dis(i, i+1) \cdot \delta; \)
9 end
10 for intersection \( v_i \in V \) do
11 Calculate the overall influence on the current intersection \( v_j \), according to Eq. (12);
12 Choose the phase action with the maximum \( Q(o_t^i, a_t^i) \) according to Eq. (14);
13 end
14 end

As for space complexity, the size of weight matrix and bias vectors in each component are as follows: (a) top-\( K \) search: \( 2K \) (b) Observation Embedding layer: \( mn + n \); (c) Graph Attention layers: \( 3n^2 + (n^2 + n) \) \( = n(4n + 1) \); (d) Q-value Prediction layer: \( np + p \). Then the total number of parameters to store is \( O(n(4n + m + p + 2) + 2K + p) \). Normally, the size of the hidden layer \( n \) is far greater than that of the data dimension \( m \) and state space \( p \). Therefore, the space complexity of LEVID_Dy is approximately equal to \( O(n^2) \). For a method with \( N \) separate RL models (without parameter sharing) to control traffic signals in \( N \) intersections, the space complexity is approximately equal to \( O(n^2 \cdot N) \).

6 Performance Evaluation

We conduct experiments with an open-source traffic simulator called CityFlow [48]. After the traffic trajectory data with specific route and start time are fed into the simulator, each OV moves towards its destination according to the environmental setting and the phase of traffic lights. The simulator provides states to a traffic signal control strategy and performs traffic signal actions from the control strategy. Meanwhile, we add a route planning module to the simulator, which controls an EV towards its destination along a dynamic route.
6.1 Setting

Both synthetic and real-world datasets are utilized to evaluate the effectiveness and efficiency of different approaches. One synthetic dataset is used to generate uniform traffic flows to test the performance of various approaches in steady traffic conditions. Four real-world traffic flow datasets are collected from four cities for evaluations on realistic and dynamic traffic conditions, and the road networks are imported to the simulator from OpenStreetMap.\(^1\) We randomly select some vehicles in the traffic flows as EVs whose routes could be dynamically changed according to a routing plan, and the rest of vehicles still follow their original routes. Indeed, EVs usually account for a very low proportion of the overall traffic flows in the real world. Nevertheless, if the proportion of EVs is set too small, there will be only a small amount of transition experiences of EVs interacting with the environment, which will cause sparse rewards in the training of an RL model. Therefore, we set the proportion of EVs as 1\% for model training, which can not only generate necessary interaction experiences, but also simulate conflicting situations of multiple EVs at the same intersection. While the proportion of EVs is set according to the real-world conditions for model testing (around 0.37 ~ 1\%). Table 2 lists the statistics of different datasets. Fig. 8 further shows the spatio-temporal distribution of traffic flows. The detailed descriptions on how we set or preprocess these datasets are as follows:

- **D_{Synthetic}**: Following the setting of [42], this dataset contains a 6 × 6 grid network where each intersection has 4 directions (West → East, East → West, South → North, North → South) and 3 lanes (300 meters long and 3 meters wide) for each direction. In the traffic flow, vehicles come uniformly with 300 vehicles/lane/hour in the East → West direction and 90 vehicles/lane/hour in the South → North direction.

- **D_{Hefei}**: There are 11 intersections in one region of Baohu district, Hefei, China. The traffic flow data are collected by roadside surveillance cameras during 9-11 a.m. on the working days of April 2021. The cameras record the time, location and vehicle ID. We set the traffic volume as the number of vehicles passing through these intersections for experiments.

- **D_{Jinan}**: There are 12 intersections in Dongfeng Sub-district, Jinan, China. The traffic flow data are collected by cameras in the similar way to D_{Hefei}.

- **D_{Hangzhou}**: There are 16 intersections in Guadan Sub-district, Hangzhou, China. The traffic flow data are collected by cameras in the similar way to D_{Hefei}.

- **D_{NewYork}**: There are 192 intersections in the Upper East Side of Manhattan. The traffic flow data are collected based on the taxi trip data containing the origin and destination geo-locations of each trip. The geo-locations are mapped to intersections and the corresponding shortest path between them is obtained. The trips falling within the selected areas are chosen for experiments.

6.2 Compared Methods

We compare our LEVID approach with various baselines and variants of LEVID. We summarize these approaches from two aspects in Table 3. On the one hand, considering the key technologies (e.g., vehicle-centric or road-centric, whether to use the RL method or not), they can be classified as: conventional traffic signal control approaches, RL-based traffic signal control approaches, route planning approaches and cooperative vehicle-road scheduling approaches. On the other hand, they can also be classified according to whether an approach is specially designed for EVs or just for OVs. All RL-based approaches are learned without any pre-trained parameters for fair comparison. The evaluation metric is the average travel time of all the EVs or OVs between origin and destination (in seconds).

**Baselines:**

- **FixedTime** [49]: It’s the most commonly used traffic signal control method with preset offsets in the real world. It utilizes a pre-determined schedule plan considering the cycle length and phase time to handle the traffic flow.

- **MaxPressure** [31]: It’s the most popular network-level traffic signal control approach in the transportation field, which greedily selects the phase with the maximum pressure.

- **GreenWave** [14]: It allows all the traffic lights in the route to turn green so that EVs can pass intersections continuously along the emergency corridor. All the intersections share the same green phase length for each movement.

- **Dijkstra** [3]: It’s a vehicle-centric scheduling approach, which builds a dynamic road network model for vehicles evacuation based on the Dijkstra algorithm.

- **AAF** [43]: It’s an advanced adaptive and fuzzy approach to reduce emergency services response time. It selects the fastest path for an EV in advance and gives priority to the EV as soon as it approaches the traffic lights on the preset route. Note that, it is just a loosely coupled cooperation, as the route planning and traffic signal control modules are sequentially conducted, while our LEVID has a tightly coupled cooperation as the two modules are simultaneously conducted.

- **Individual RL** [27]: It’s the individual deep RL approach without considering the information of neighbors. Each intersection is controlled by one heterogeneous agent which updates its own network independently.

---

1. https://www.openstreetmap.org
OneModel [50]: It designs the state and reward of the agent in the same way with Individual RL. Each agent only considers the state of the roads connecting the controlled intersections and all agents share the same centralized critic model.

CoLight [42]: It’s an RL-based traffic signal control approach utilizing graph attention networks to automatically extract traffic features of adjacent intersections for facilitating communication.

Variants of LEVID:

- LEVID_UnDy: It removes the real-time route planning module from the LEVID. Meanwhile, its traffic signal control module removes the design of dynamic directed graph, and chooses the top-\(K\) relevant neighbors based on the fixed geographic distance. This variant can show the improvements brought by the design of our state and reward function.

- LEVID_Dy: It removes the real-time route planning module from the LEVID and retains the traffic signal control module, which selects the top-\(K\) relevant neighbors based on a dynamic directed graph.

- LEVID_APF: It utilizes the artificial potential field method which only considers the gravity and immediate repulsion, instead of the real-time path planning module of LEVID.

### 6.3 Overall Performance Comparison

Table 4 compares the average travel time of both OVs and EVs achieved by LEVID and various baselines/variants on the five datasets.

| Classification of Various Approaches | OVs          | EVs          |
|--------------------------------------|--------------|--------------|
| Conventional Traffic Signal Control | FixedTime,   | GreenWave    |
|                                       | MaxPressure  |              |
| RL-based Traffic Signal Control      | Individual RL| LEVID_UnDy,  |
|                                       | OneModel,    | LEVID_Dy     |
| Route Planning                      | /            | Dijkstra     |
| CVRS (loosely coupled)              | /            | AAF          |
| CVRS (tightly coupled)              | /            | LEVID_APF,   |
|                                     |              | LEVID        |

### 6.3.1 Advantages of LEVID Over Conventional Traffic Signal Control Approaches

From Table 4, we observe that FixedTime has similar performance to MaxPressure on \(D_{\text{Synthetic}}\) with a uniformly simulated traffic flow, while MaxPressure performs much better than FixedTime on the four real-world datasets, indicating that MaxPressure has a stronger ability of handling dynamic traffic flows. However, both FixedTime and MaxPressure do not consider the priorities of EVs, resulting in that OVs and EVs have similar performance. By contrast, GreenWave is specially designed for EVs and greatly reduce the average travel time of EVs on all the datasets (at most 71.71% and 45.07% reductions compared with FixedTime and MaxPressure, respectively). Nevertheless, GreenWave increases the average travel time of OVs (at most 258.0s longer than FixedTime), and there is a large performance gap between OVs and EVs (at most 1885.5s difference). By contrast, our LEVID and its three variants not only greatly reduce the average travel time of EVs on all the datasets (at most 74.71% and 50.94% reductions compared with FixedTime and MaxPressure, respectively, by LEVID), but also shorten the average travel time of OVs in most cases (at most 807.3s difference by LEVID). The results demonstrate the obvious advantages of our LEVID from two aspects: 1) utilize a learning-based traffic signal control strategy instead of a rule-based strategy for handling dynamic traffic flows, and 2) integrate it with a route planning strategy to further reduce the waiting time at intersections with a heavy traffic.

### 6.3.2 Advantages of LEVID over RL-Based Traffic Signal Control Approaches

From Table 4, we observe that Individual RL performs better than other two RL-based baselines, OneModel and CoLight, and even performs best for OVs on four small-scale datasets, \(D_{\text{Synthetic}}, D_{\text{HeFei}}, D_{\text{Jinan}}\) and \(D_{\text{Hangzhou}}\). It is because Individual RL trains an exclusive agent for each intersection, which can evaluate the intersection state more accurately and reduce the performance loss. However, Individual RL cannot be applied to a large-scale dataset (e.g., \(D_{\text{NewYork}}\)) due to the low computational efficiency. On the contrary, OneModel utilizes a shared centralized critic network, which may ignore the differences between individuals, resulting in an inevitable performance loss. Compared with OneModel, CoLight reduces the average travel time of EVs by 9.1%, 55.7%, 59.5%, 6.4% and 26.3% on \(D_{\text{Synthetic}}, D_{\text{HeFei}}, D_{\text{Jinan}}, D_{\text{Hangzhou}}\) and \(D_{\text{NewYork}}\), respectively, because it considers...
the state of neighboring intersections and leverages the GAT to model the interactions between neighboring intersections. Compared with CoLight, our LEVID and its three variants achieve consistent and obvious performance improvements for EVs, and achieve similar performance for OVs. More specifically, LEVID_UnDy reduces the average travel time of EVs by 14.76%, 8.91%, 12.94%, 32.72% and 19.24% than CoLight on the five datasets, respectively, which demonstrates the importance of the reward design considering both EVs and OVs simultaneously. LEVID_Dy further reduces the average travel time of EVs by 1.70%, 10.76%, 7.36%, 9.51% and 14.82% than LEVID_UnDy on the five datasets, respectively, which demonstrates the importance of utilizing a dynamic directed graph. Moreover, our proposed approaches have a larger advantage with the increase of the road network scale.

### 6.3.3 Advantages of Integrating a Real-Time Route Planning Module

From Table 4, we observe that Dijkstra reduces the average travel time of EVs by 0.29%, 10.72%, 5.86%, 17.25% and 42.92% than OVs in the same environment on the five datasets, respectively, which demonstrates the importance of route planning. Nevertheless, this performance improvement is far less than that by several CVRS, because a vehicle-centric approach just avoids congested roads in a passive way while failing to proactively improve traffic conditions. By contrast, AAF reduces the average travel time of EVs by 18.53%, 61.61%, 70.97%, 49.37% and 72.36% than OVs in the same environment on the five datasets, respectively. Compared with GreenWave, AAF greatly reduces the average travel time of OVs due to the merit of integrating a route planning module. However, AAF cannot achieve consistent advantages for EVs especially on small-scale datasets, demonstrating its limited ability of handling frequently changing traffic flow in the way of planning routes in advance. Compared with AAF, LEVID_APF reduces the average travel time of OVs and EVs by at most 58.19% and 6.81%, respectively, on the five datasets, which demonstrates the importance of the tightly coupled cooperation between route planning and traffic signal control. In spite of this, we observe that LEVID_APF performs worse than AAF for EVs on $D_{NewYork}$ because LEVID_APF does not consider the limitation of the immediate repulsion in a large-scale road network. By contrast, LEVID reduces the average travel time of EVs by 1.96%, 12.54%, 11.48%, 11.94% and 18.26% than LEVID_APF on the five datasets, respectively, which demonstrates the importance of considering the long-term repulsion.

### 6.4 Convergence Comparison

In Fig. 9, we present the average travel time of EVs evaluated at each episode to the corresponding learning curves for the five RL-based traffic signal control approaches. The results show that our method has better performance in both time to threshold (learning time to achieve a pre-specified performance level) and asymptotic performance (final learned performance). The convergence curve also presents the influence of dynamic traffic flows on the convergence. The convergence curves on the synthetic dataset are smoother while the dynamic real-world datasets bring some fluctuations to the convergence curve of most RL-based approaches. The training time (total time for 100 episode training) of all RL-based traffic signal control approaches are also presented. For fair comparison, each model is trained individually. As shown in Fig. 9f, the time consumption of LEVID_Dy is much less than that of Individual RL and all the approaches with centralized model are efficient. This is consistent with the complexity analysis of the centralized model. In addition, the actual time consumption of Individual RL on $D_{NewYork}$ is not provided, as each episode takes more than 100 hours when all models are trained centrally on one server.

### 6.5 Case Study

The time-space diagrams are utilized to show the trajectory of one EV and the corresponding traffic signal control plan on dataset $D_{NewYork}$. In Fig. 10, the left part shows the real-world network structure and the right part denotes the specific driving process of the EV. The x-axis is the time and y-axis denotes the distance. There are one gray line denoting the trajectory of the EV and 11 bands with green-yellow-red colors denoting the changing phases of 11 intersections in this trajectory. This EV turns right at the 9th intersection, where the right turn signal is always green. Fig. 10 illustrates

### Table 4

Comparisons of Average Travel Time of Both OVs and EVs on the Five Datasets

| Model          | $D_{Synthetic}$ | $D_{Hefei}$ | $D_{Jinan}$ | $D_{Hangzhou}$ | $D_{NewYork}$ |
|----------------|-----------------|-------------|-------------|-----------------|---------------|
|                | OVs             | EVs         | OVs         | EVs             | OVs           | EVs           |
| FixedTime      | 209.7           | 209.1       | 1175.9      | 1097.6          | 867.7         | 869.3         | 654.6         | 645.5          | 2239.1        | 1399.4        |
| MaxPressure    | 194.5           | 192.4       | 660.2       | 642.45          | 387.4         | 394.7         | 514.2         | 523.4          | 1666.1        | 1113.5        |
| GreenWave      | 203.8           | 169.6       | 1384.9      | 529.2           | 832.2         | 245.9         | 796.8         | 336.4          | 2497.1        | 611.6         |
| Dijkstra       | 210.4           | 209.8       | 1247.9      | 1114.7          | 895.3         | 841.1         | 690.9         | 572.4          | 2020.5        | 1152.7        |
| AAF            | 205.2           | 167.2       | 1417.8      | 544.3           | 829.3         | 241.7         | 676.7         | 343.1          | 2107.5        | 581.3         |
| Individual RL | 189.36          | 188.5       | 569.3       | 570.6           | 343.1         | 347.3         | 404.3         | 403.7          | 2072.8        | 347.4         |
| OneModel       | 211.8           | 211.2       | 1411.3      | 1459.2          | 724.9         | 725.8         | 570.8         | 520.5          | 1979.1        | 1219.8        |
| CoLight        | 192.5           | 188.4       | 625.7       | 603.3           | 293.3         | 291.6         | 534.4         | 504.82         | 1459.5        | 906.6         |
| LEVID_UnDy     | 197.4           | 160.6       | 667.2       | 549.6           | 354.6         | 254.1         | 567.9         | 339.9          | 1596.3        | 732.8         |
| LEVID_Dy       | 193.1           | 157.8       | 674.3       | 490.5           | 352.5         | 235.6         | 586.7         | 307.7          | 1574.5        | 624.2         |
| LEVID_APF      | 198.6           | 158.5       | 664.0       | 507.5           | 347.4         | 248.4         | 556.6         | 330.5          | 1434.7        | 667.9         |
| LEVID          | 195.4           | 155.4       | 654.0       | 443.9           | 341.8         | 220.2         | 571.7         | 291.1          | 1431.8        | 546.5         |
that the EV takes 300s to pass through 11 intersections. We observe that the duration and frequency of the green phase at each intersection are different. This shows that each traffic light dynamically adjusts the phase according to the specific traffic conditions at the intersection. Meanwhile, the gray track line shows that LEVID can automatically form a green wave for the EV to help it pass quickly.

**Fig. 9.** Convergence speed and running time of LEVID_Dy (green continuous curves) and other 4 RL-based traffic signal control approaches (dashed curves) during training. In most cases, LEVID_Dy starts with the best performance (Jumpstart), reaches to the pre-defined performance the fastest (Time to Threshold), and ends with the optimal policy (Asymptotic). Curves are smoothed with a moving average of 5 points. Note that, the convergence curve of Individual RL is not provided on $D_{NewYork}$, as it cannot be applied to a large-scale dataset due to the low computational efficiency.

![Convergence speed](image)

7 Conclusion

In this paper, we consider a cooperative vehicle-infrastructure system to help EVs arrive faster. Based on the key insight that real-time vehicle-road interaction and strategy coordination can bring more benefits, we propose LEVID, a learning-based cooperative vehicle-road scheduling approach. LEVID contains a real-time route planning module and a collaborative traffic signal control module, which influences each other and makes decisions iteratively. The first module adapts the artificial potential field method to handle the real-time changes of traffic signals and jump out of the local optimum. The second module utilizes the multi-agent reinforcement learning framework to handle the traffic features that are hard to be combined linearly based on human experience and predefined rules. It further leverages graph attention networks based on a dynamic directed graph to model the interactions between intersections. Extensive experiments based on multiple real-world datasets demonstrate that our approach outperforms the state-of-the-art baselines.

![Space-time diagram](image)
Lige Ding received the BS degree from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2017, where he is currently working toward the PhD degree. His research interests include mobile crowd sensing, urban computing, and reinforcement learning.

Dong Zhao (Member, IEEE) received the BS degree from the Department of Computer Science and Technology, Henan University, Kaifeng, China, in 2008, and the PhD degree from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He is currently a professor with the Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications. He was a visiting PhD student with the Illinois Institute of Technology, Chicago, Illinois, from 2012 to 2013, and was a visiting scholar with Rutgers University, New Brunswick, New Jersey, from 2019 to 2020. His research interests include Internet of Things, mobile crowd sensing, urban computing, and data science, and he has published more than 60 papers and two books on these fields. He was awarded the China Computer Federation (CCF) Outstanding Doctoral Dissertation Award, in 2015, the ACM Beijing Doctoral Dissertation Award, in 2015, and the Natural Science Award of the Ministry of Education, China, in 2017.

Zhaoefeng Wang received the BS degree from International School, Beijing University of Posts and Telecommunications, Beijing, China, in 2021. He is currently working toward the MS degree with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include reinforcement learning and urban computing.

Guang Wang received the PhD degree in computer science from Rutgers University, New Brunswick, New Jersey. He is currently a postdoctoral research associate with the MIT Institute for Data, Systems, and Society, Schwarzman College of Computing. He is interested in mobile computing, cyber-physical systems, Big Data analytics, and machine learning. Currently, his research is focusing on human mobility, ubiquitous computing, mobile computing, autonomous vehicle, and fairness, etc. His technical contributions have led to more than 35 peer-reviewed publications in premium conferences and journals, e.g., MobiCom, IMWUT/UbiComp, RTSS, KDD, ICDE, WWW, AAAI, and TIST.

Chang Tan is currently a general manager of iFLYTEK Intelligent Transportation Business Department. His general area of research interest includes data mining, with a focus on solving urban traffic problems using Big Data.

Lei Fan is currently working in iFLYTEK, mainly responsible for data mining and urban Big Data analysis.

Huadong Ma (Fellow, IEEE) received the BS degree in mathematics from Henan Normal University, Xinxiang, China, in 1984, the MS degree in computer science from the Shenyang Institute of Computing Technology, Chinese Academy of Science, China, in 1990, and the PhD degree in computer science from the Institute of Computing Technology, Beijing, Chinese Academy of Science, in 1995. He is currently a professor with the School of Computer Science, Beijing University of Posts and Telecommunications (BUPT), China. From 1999 to 2000, he held a visiting position with the University of Michigan, Ann Arbor, Michigan. His current research interests include Internet of Things and sensor networks, multimedia computing, and he has published more than 300 papers in prestigious journals (such as ACM/IEEE Transactions) or conferences (such as ACM SIGCOMM, ACM MobiCom/ MM, IEEE INFOCOM) and five books. He received the first class prize of the Natural Science Award of the Ministry of Education, China, in 2017. He received the 2019 Prize Paper Award of IEEE Transactions on Multimedia, 2018 Best Paper Award from IEEE Multimedia, Best Paper Award in IEEE ICPADS2010, and Best Student Paper Award in IEEE ICME2016 for his coauthored papers. He received the National Funds for Distinguished Young Scientists, in 2009. He was/is an editorial board member of the IEEE Transactions on Multimedia, IEEE Internet of Things Journal, ACM Transactions on Internet of Things, and Multimedia Tools and Applications. He serves as a chair of the ACM SIGMOBILE China.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.