Hierarchical Reinforcement Learning for Dynamic Autonomous Vehicle Navigation at Intelligent Intersections

Qian Sun
qsunal@connect.ust.hk
The Division of Emerging Interdisciplinary Areas, The Hong Kong University of Science and Technology

Le Zhang*
zhangle0202@gmail.com
Business Intelligence Lab, Baidu Research

Huan Yu
huanyu@ust.hk
Thrust of Intelligent Transportation, The Hong Kong University of Science and Technology(Guangzhou)
The Department of Civil Engineering, The Hong Kong University of Science and Technology

Weijia Zhang
wzhang411@connect.hkust-gz.edu.cn
The Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology(Guangzhou)

Yu Mei
whqyqy@hotmail.com
Department of Intelligent Transportation System, Baidu Inc.

Hui Xiong*
xionghui@ust.hk
The Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology(Guangzhou)
The Department of Computer Science and Engineering, The Hong Kong University of Science and Technology

ABSTRACT

Recent years have witnessed the rapid development of the Cooperative Vehicle Infrastructure System (CVIS), where road infrastructures such as traffic lights (TL) and autonomous vehicles (AVs) can share information among each other and work collaboratively to provide safer and more comfortable transportation experience to human beings. While many efforts have been made to develop efficient and sustainable CVIS solutions, existing approaches on urban intersections heavily rely on domain knowledge and physical assumptions, preventing them from being practically applied. To this end, this paper proposes *NavTL*, a learning-based framework to jointly control traffic signal plans and autonomous vehicle rerouting in mixed traffic scenarios where human-driven vehicles and AVs co-exist. The objective is to improve travel efficiency and reduce total travel time by minimizing congestion at the intersections while guiding AVs to avoid the temporally congested roads. Specifically, we design a graph-enhanced multi-agent decentralized bi-directional hierarchical reinforcement learning framework by regarding TLs as manager agents and AVs as worker agents. At lower temporal resolution timesteps, each manager sets a goal for the workers within its controlled region. Simultaneously, managers learn to take the signal actions based on the observation from the environment as well as an intention information extracted from its workers. At higher temporal resolution timesteps, each worker makes rerouting decisions along its way to the destination based on its observation from the environment, an intention-enhanced manager state representation, and a goal from its present manager. Finally, extensive experiments on one synthetic and two real-world network-level datasets demonstrate the effectiveness of our proposed framework in terms of improving travel efficiency.

CCS CONCEPTS

• **Computing methodologies** → **Control methods**; • **Applied computing** → **Transportation**.

KEYWORDS

Dynamic Vehicle Navigation, Traffic Signal Control, Mixed Autonomy Traffic Control, Reinforcement Learning, Multi-Agent System, Intelligent Transportation System

ACM Reference Format:
Qian Sun, Le Zhang, Huan Yu, Weijia Zhang, Yu Mei, and Hui Xiong. 2023. Hierarchical Reinforcement Learning for Dynamic Autonomous Vehicle Navigation at Intelligent Intersections. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599839

1 INTRODUCTION

Recent advances in autonomous driving and Vehicle-to-Everything (V2X) communication technologies enable a new paradigm for promoting efficient and intelligent transportation solutions by controlling AVs and road infrastructures in a collaborative way. In recent years, researchers in the transportation field started to design algorithms to jointly optimize traffic signal plans and vehicle trajectories. For example, prior studies [14, 15, 38] proposed approaches such as mixed integer linear programming to optimize

*Corresponding authors.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599839

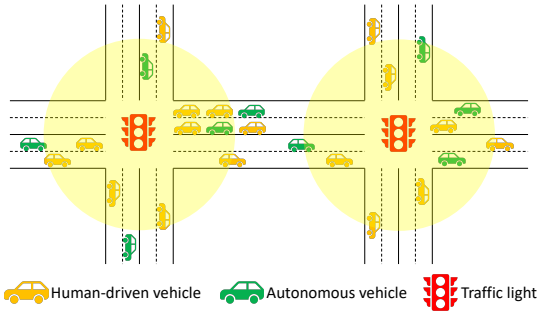


Figure 1: Mixed traffic at urban intersections.

both the signal timing scheme and the accelerations of AVs. However, such optimization-based methods depend hugely on expert knowledge and theoretical physics principles, potentially leading to bad results when implemented in the dynamically changing real-world scenarios [5, 9].

In addition to the aforementioned optimization-based methods, machine learning-based approaches have been shown to be effective in solving individual tasks in the traffic system. For example, deep reinforcement learning (DRL) has been widely used in traffic signal control [1, 18, 31], outperforming traditional heuristic methods in terms of congestion minimization by a large margin. Moreover, in the field of vehicle trajectory control, RL-based models have also achieved significant success, such as dynamic navigation [13, 27] and control of driving behaviors [25, 32, 34] in fully autonomous or mixed traffic circumstances. Nevertheless, to the best of our knowledge, none of these existing studies in machine learning-based mixed traffic control has studied the joint control of autonomous vehicle navigation and traffic signals. It is a non-trivial task, which faces the following two major challenges: (1) Traffic signal control and AV control are highly interdependent. That is, the real-time changes of traffic signals impact the motions of approaching vehicles, while the dynamic changes of autonomous vehicles are key factors affecting the decision-making of traffic signals [15, 38]. As a result, the coupled nature leads to a highly dynamic and chaotic traffic system. (2) Moreover, the objectives of the two sub-tasks are independent and inconsistent: traffic signal control works towards reducing congestion, while AV navigation aims at avoiding congestion. It is challenging to balance these two objectives under a unified framework to improve overall travel efficiency.

To this end, in this paper, we propose a multi-agent hierarchical RL framework to deal with the above challenges and collaboratively control the traffic signal plans and autonomous vehicle rerouting in hybrid urban traffic as depicted in Figure 1. The goal is to improve overall traffic efficiency by reducing congestion and minimizing the travel time of vehicles. Inspired by the feudal framework [19, 24], a classical hierarchical reinforcement learning architecture, we model our system in a two-level hierarchy, with a bi-directional propagation mechanism. Particularly, we regard the traffic signals as manager agents and AVs as worker agents. Both managers and workers interact with the same environment and information can be shared between each manager-worker pair through bi-directional message passing. In this way, collaborations between AVs and traffic signals can be effectively promoted. Specifically, at each time step, we build a timely signal-vehicle ego-network for each signal control zone in which traffic signals and AVs are interconnected. Then a dynamic heterogeneous graph neural network is adopted to aggregate the regional navigation intention for managers. Afterwards, managers combine their observations from the environment and the extracted intention to generate a comprehensive state representation, which is further propagated to workers to enhance their states. With the coordinated states, we leverage deep Q-networks to learn state-action values for both managers and workers. Notably, in the traffic signal control task, we construct a signal-signal graph based on the road network and leverage a graph convolutional network to enhance the cooperation among signals. Finally, to balance the inconsistent objectives of two sub-tasks, we leverage a goal network to compute the manager state based desired goal vectors, which are used to guide the actions of workers. We reward each worker for taking the navigation action that yields a state transition close to matching the goal.

The major contributions of this paper are summarized as follows:

- We propose a novel multi-agent hierarchical RL framework, *NavTL*, to dynamically control the traffic signals and rerouting directions of the AVs simultaneously.
- We provide an improved version of the uni-directional feudal RL framework, which involves both top-down guidance from managers and bottom-up intention from workers. Additionally, we incorporate graph neural networks to enable vehicle-signal cooperation and signal-signal coordination.
- We conduct extensive experiments on one synthetic and two real-world datasets to demonstrate the superior performance of *NavTL* in terms of minimizing congestion and improving travel efficiency.
- To the best of our knowledge, *NavTL* is the first work that: (i) solves the task of coordinated control of traffic signals and autonomous vehicles in the mixed traffic environment, (ii) applies graph-enhanced hierarchical reinforcement learning on the task of dynamic vehicle navigation in urban intersections and tests the model with real-world road traffic data, (iii) employs hierarchical reinforcement learning involving heterogeneous agents that focus on two distinct tasks.

2 RELATED WORKS

The relevant literature can be classified into three categories, namely *traffic signal control*, *dynamic vehicle navigation*, and *hierarchical reinforcement learning*.

2.1 Traffic Signal Control

The target of intelligent traffic signal control (TSC) is to study the change of phase plans of traffic signals to reduce congestion. Traditional methods of traffic signal control include MaxPressure [23], Webster [28], SOTL [4], etc. These methods mainly control traffic signals in a heuristic manner. For example, MaxPressure [23] greedily selects the next phase corresponding to the maximum pressure, which is defined as the total difference between number of waiting vehicles in the upstream incoming lanes and that in the downstream [29]. However, traditional methods rely on strong assumptions which limit their performance. Accordingly, several studies formulate this control task as an RL problem, where each traffic signal is regarded as an agent. Generally, the state refers

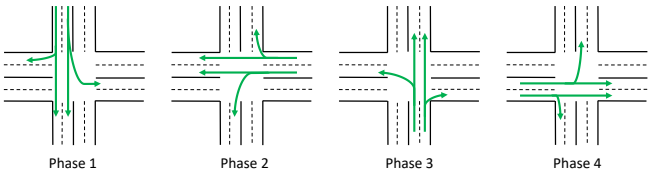


Figure 2: A four-phase cycle in a demo intersection.

to the information regarding the traffic situation at intersections, such as length of waiting vehicles in the incoming directions, vehicles' average waiting time, total number of approaching vehicles on incoming roads, and average travel speed of approaching vehicles [1, 7, 18]. The agent's action is either the next phase in the upcoming time step, assuming the phase order is not predefined, or the time to switch to the upcoming phase, assuming the order is predetermined [26, 29, 30]. Although these RL-based models have been shown to be effective in the TSC task, they have merely studied the simple homogeneous setup where only human-driven vehicle(HVs) exist in the network. None of them has considered the TSC task in mixed scenarios where both HVs and AVs co-exist.

2.2 Dynamic Vehicle Navigation

Vehicle route planning aims at finding a path for the vehicle to its destination with minimum cost. Classical pathfinding solutions such as Dijkstra's algorithm [6] and A* [10] compute the shortest path between a departure node and a destination node greedily. Though widely adopted in real-world applications, these static path-finding algorithms do not consider the changing dynamics of the roads in real-time. Recently, a few works have proposed learning-based real-time vehicle navigation models by formulating the navigation task as a sequence of rerouting decisions [8, 13, 27]. They define the task as an RL problem in which the vehicle learns to make a turn at the upcoming intersection, e.g. left-turn, right-turn, or go-straight at a four-way crossing, given its observation including its current location and destination information, as well as real-time road traffic information. However, these works make hypothetical assumptions regarding autonomous vehicles, i.e. AVs can directly obtain the traffic data of all the roads as their observation, which is unrealistic in real-world implementations. Moreover, they neglect the controllability of traffic signals in the system setup, which is a key component of urban traffic scenarios.

2.3 Hierarchical Reinforcement Learning

Hierarchical Reinforcement Learning (HRL), a special RL architecture, decomposes a complex RL problem into sub-tasks and trains the hierarchical policy in an end-to-end manner [2, 11, 20]. Existing methods in HRL can be categorized into two classes, including the options framework and the feudal hierarchy. The options framework introduces a set of sub-tasks, namely options, to the main task. An option is defined as a policy, a termination condition, and an initiation set. During the learning process, if the current state belongs to the initiation set of an option, the option will be activated until termination [22]. Differently, feudal reinforcement learning [24] introduces a two-level architecture in which the higher-level manager is trained to learn a goal to guide the trajectories of the lower-level worker, while the worker is trained to learn to satisfy the goal.

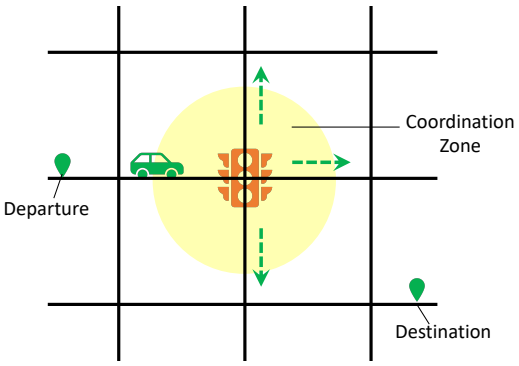


Figure 3: Illustrative demonstration of the navigation task.

A few works have applied the HRL framework in complex transportation scenarios, such as traffic signal management systems and ride-hailing platforms [12, 16, 17]. For example, [16] solves the traffic signal control task by defining regions as managers and intersections as workers, and setting the states of the region managers as abstractions of workers' states within the region. However, the managers and workers in this problem formulation are homogeneous. In other words, both manager and worker focus on the same task and have the same state and action space. To the best of our knowledge, no studies have focused on the application of feudal RL in heterogeneous tasks, where managers and workers both interact with the same environment but acquire different states, and learn to take distinct actions simultaneously and collaboratively.

3 PROBLEM FORMULATION

Our task is to improve travel efficiency at the system level through joint control of autonomous vehicles and traffic signals, aiming to minimize congestion at intersections while guiding AVs to avoid congested roads by providing intelligent rerouting choices. In other words, our CVIS contains two sub-tasks, namely traffic signal control and dynamic autonomous vehicle navigation, respectively.

For traffic signal control, we study the change of the phase plan of traffic lights. A phase cycle is defined as a set of ordered green phases, each followed by a default yellow phase. As shown in Figure 2, a green phase defines the permitted traffic flow(s) of one or more non-conflicting directions. Here, the duration of each phase T_G is fixed, and our task is to control their order.

For dynamic autonomous vehicle navigation in mixed traffic scenario, we assume all AVs are under control, while HVs are not. Given the vehicle's departure location and destination location, the real-time navigation task can be regarded as a series of decisions made at each intersection along its way to the destination. Once the vehicle enters the coordination zone of an intersection, as illustrated in Figure 3, it decides the way to turn at the current crossing. Immediately, a new route is formulated between the road it decided to turn to and the destination road with the least travel cost.

4 METHODOLOGY

In this section, we present the technical details of our NavTL framework. We begin by explaining the multi-agent RL setup and the deep Q-learning algorithm. We then introduce the key components

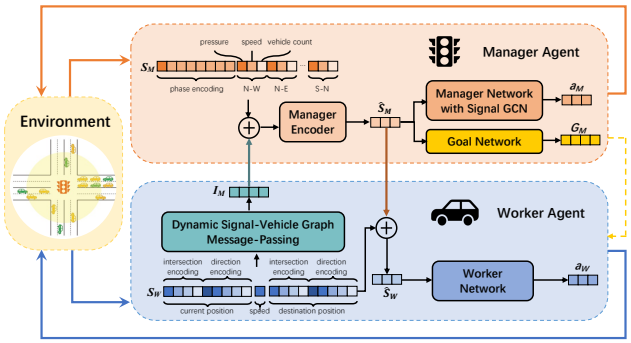


Figure 4: An overview of the NavTL framework.

that facilitate signal-vehicle cooperation in our framework, including the regional navigation intention extraction module, intention-enhanced message passing, and hierarchical navigation guidance. Finally, we discuss inter-signal communications for signal control.

4.1 Multi-Agent RL Setup

Assuming each traffic signal and autonomous vehicle is controlled by an individual agent, we treat the CVIS joint control task as a Partially-Observable Markov Decision Process(POMDP) problem, where each agent observes part of the total system condition. Specifically, the problem is characterized by the following components $\langle S_l, O_l, A_l, r_l, \pi_l, P_l, \gamma_l, S_v, O_v, A_v, r_v, \pi_v, P_v, \gamma_v \rangle$:

- **Traffic Signal State**. S_l represents the global state space for all traffic signal agents. Each agent partially observes $o \in O_l$, which includes its current phase encoding (a one-hot vector indicating the current green phase index in the phase cycle), pressure, total number of vehicles and average travel speed in 12 total incoming-outgoing flow directions (*N-S*, *N-W*, *N-E*, *S-N*, *S-W*, *S-E*, *W-N*, *W-S*, *W-E*, *E-N*, *E-W*, *E-S*). Here we define a possible traffic flow e.g. *N-W*, as the flow from the incoming direction North to the outgoing direction West at the corresponding intersection.
- **Traffic Light Action**. A_l denotes the action space for traffic signal agents, which is a possible green phase index to be selected during the next $t = T_G$ period.
- **Traffic Light Reward**. we select the negative of total queue length for the reward function r_l, which is defined as the total number of waiting vehicles on all incoming roads at the intersection, following [30].
- **Autonomous Vehicle State**. S_v stands for the global state space for all vehicle agents, each of which observes $o \in O_v$. Here, $O_v = [V, X_{cur}, D_{cur}, X_{dest}, D_{dest}]$, in which V is the vehicle's speed; X_{cur} is a one-hot encoding of length N_M (total number of traffic signals), representing the index of the vehicle's upcoming intersection; D_{cur} is a one-hot direction encoding of length 4 which stands for its current approaching direction to the upcoming intersection i.e., N/S/E/W; similarly, X_{dest} and D_{dest} are encodings that represent the destination intersection and the direction of the destination lane to that intersection .
- **Autonomous Vehicle Action**. A_v is the action space for the vehicle agents, an action is a possible turning direction at the upcoming crossing. In a regular intersection without turnarounds, the number of possible actions is 3, including turn-left, turn-right, and go-straight.
- **Autonomous Vehicle Reward**. r_v is the reward function for vehicle agents, we select the accumulated travel time since the last action time as the reward, following [13].
- **Policy**. π_l and π_v are the policy functions for the signal agents and vehicle agents respectively. The policy $O \times A \rightarrow \pi$ guides the agents to select the optimal actions given different state observations.
- **State Transition Function**. P_l and P_v are the state transition functions for the signal agents and vehicle agents respectively. At each timestep, given the global state s_t and the joint actions $\hat{a}_t$ produced by the agents, i.e., $a_1 \times a_2 \times ... \times a_n$, the system produces the next state s_{t+1} following the transition probability $P(s_{t+1}|s_t, \hat{a}_t)$.
- **Discount Factor**. γ_l and γ_v are the discount factors in calculating accumulated returns, $R = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$, for the signal and vehicle agents, respectively.

4.2 Deep Q-Learning

We adopt a classical RL algorithm in our framework, namely Deep Q-Networks (DQN) [21], which is an extension of the traditional Q-learning algorithm [22]. Q-learning is a value-based and model-free RL model that learns the state-action value function i.e., $Q(s, a)$, which specifies the value of the action to perform in a given state according to the optimal policy π^* [22]. With the raising power of deep learning technology, $Q(s, a)$ can be approximated by neural networks, so-called deep Q-learning. In this paper, we employ two neural networks, namely a current network Q and a target network $\tilde{Q}$, such that Q is directly trained to learn the state-action values as the agent interacts with the environment, while $\tilde{Q}$ is asynchronously updated with the most recent weights of Q every T steps during the learning process. In such way, learning stability can be improved. The parameter θ of Q is optimized by minimizing the Temporal-Difference (TD) error, i.e., the difference between the TD target and the predicted Q value, where the TD target is the sum of the immediate reward r_t and the discounted optimal Q value at the next state S_{t+1} computed using $\tilde{Q}$. The TD loss is defined below:

$$L_\theta = E[r_t + \gamma \max_a \tilde{Q}(S_{t+1}, a|\tilde{\theta}) - Q(S_t, a_t|\theta)]^2, \quad (1)$$

where a_t is the action at timestamp t, and γ is the discount factor.

4.3 Signal-Vehicle Cooperation for Navigation

In our problem setup, traffic signals operate at an upper level, controlling the macroscopic vehicle flows, while AVs operate at a lower level, controlling their own routing directions. This particular setup naturally aligns with the structure of the feudal RL architecture, which constructs a two-level hierarchy in which the higher-level manager updates at lower temporal resolution e.g. every few timesteps, while the worker operates at higher temporal abstraction i.e., every timestep [19, 24]. In traditional feudal RL, the manager does not interact directly with the environment, its state is an abstraction of the worker's state. And the manager learns a goal vector G as its action, which is not directly executed in the

environment but fed to the worker instead. The worker then takes G, a desirable state that it should reach, into consideration when calculating its reward. Indeed, the original feudal RL is designed to solve homogeneous tasks. In other words, this framework works well when managers and workers focus on a homogeneous task since in such cases managers' observations and workers' observations only differ in scale. Differently, our managers and workers are heterogeneous agents focusing on two different tasks, with distinct observation and action spaces. Therefore, we develop an improved version of the feudal RL framework for the coordinated control task by introducing the bi-directional message-passing mechanism, which enables the integration of the bottom-up intention information from workers to managers as well as the top-down guidance information from managers to workers. In this way, the collaboration between managers and workers is much closer. Figure 4 presents the overview of our framework, and we will further discuss details of the framework in the following subsections.

4.3.1 Regional Navigation Intention Extraction. The prevailing methods for controlling traffic signals primarily rely on the present conditions at the intersections rather than taking potential future information into account. In our cooperative system, we can obtain the driving intention of AVs, such as the current approaching direction and desired destination, which intuitively provide foresight to traffic signals for better control. To capture the intention of AVs for traffic signals, we construct a heterogeneous signal-vehicle graph at each timestamp, which is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which the set of nodes is $\mathcal{V} = \{\mathcal{V}_M, \mathcal{V}_W\}$ where $\mathcal{V}_M$ and $\mathcal{V}_W$ represent all traffic light(manager) nodes and AV(worker) nodes respectively, and $\mathcal{E}$ includes edges between each manager-worker pair, as shown by the dotted lines in Figure 5. Since the coordination zone of each signal node is independent of each other, it is intuitive to further decompose $\mathcal{G}$ into ego-graphs for each manager i, denoted as $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, in which $\mathcal{V}_i = \{i, \mathcal{V}_W^i\}$, where $\mathcal{V}_W^i$ represents all AV nodes in the control zone of manager i.

Considering graph neural networks (GNNs) have achieved major success in learning information from topological structures [35, 36], we perform a message-passing neural network within the ego-graph of manager i to obtain the spatially aggregated states X_i from connected workers' states S_j as follows:

$$X_i = \frac{1}{|\mathcal{V}_i|} \sum_{j \in \mathcal{V}_i} W_i S_j, \tag{2}$$

where W_i is the learnable transformation matrix. More importantly, since the signal-vehicle relationship evolves over time while AVs traverse through the intersections, we model the dynamic intention by further combining X_i obtained at the current timestep t with the historical intention representations learned at the past k timesteps to obtain a temporally aggregated intention as follows:

$$I_i^t = f(I_i^{t-k}, I_i^{t-k-1}, ..., I_i^{t-1}, X_i), \tag{3}$$

where $f(\cdot)$ is flexible, it can be a linear function or other sequential models, such as RNN [3, 33]. For simplicity, we use the linear function during implementation. In such way, we are able to obtain the dynamic intention representation of workers at time t, which will be delivered to managers via the bottom-up message propagation.

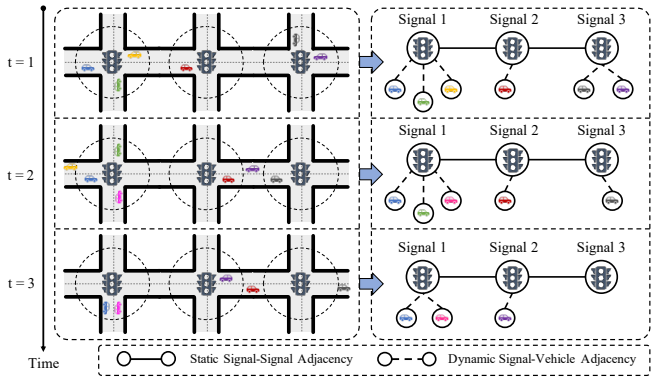


Figure 5: Illustration of the dynamic Signal-Vehicle graph and the static Signal-Signal graph in the traffic network.

4.3.2 Bi-directional Message Passing. As mentioned before, it is crucial for workers to notify managers their travel intention. Hence, after obtaining the intention representation, we propagate it in a bottom-up manner from workers to managers to enhance collaboration. Specifically, we first concatenate the current state S_i^t of each manager i and the intention I_i^t, and then feed it to the Manager Encoder which is a Multi-layer Perceptron (MLP), to obtain an intention-aware manager state representation $\hat{S}_i^t$:

$$\hat{S}_i^t = MLP(S_i^t \oplus I_i^t). \tag{4}$$

Intuitively, this enhanced representation can potentially help workers gain insights into the congestion level of roads connected to the upcoming intersection, helping them to make better-rerouting decisions. Hence, we propagate it back to workers in a top-down manner to compute the manager-enhanced worker state $\hat{S}_j^t$:

$$\hat{S}_j^t = S_j^t \oplus \hat{S}_i^t, \ \ where\ j \in V_i^t. \tag{5}$$

Finally, $\hat{S}_j^t$ will be fed into workers' Q-network Q_v with parameter θ_v containing three MLP layers for learning the worker's Q-value.

By this bi-directional propagation scheme, we not only bring the bottom-up intention representation from workers to managers, but also incorporate the top-down intention-enhanced manager state to workers. As a result, the challenging problem of jointly modeling the mutually dependent TSC and AV control tasks can be solved.

4.3.3 Hierarchical Navigation Guidance. Another substantial challenge we face is that the objectives of our heterogeneous agents do not align with each other. To deal with this problem, we adopt a neural network to learn the goal vector G_i^t based on the intention-aware manager states:

$$G_i^t = MLP(\hat{S}_i^t). \tag{6}$$

The goal G_i^t indicates the desired next state for the worker from the manager perspective and is used to guide worker's action. Specifically, we first calculate the ideal next state for worker as $S_j^t + G_i^t$, then a constraint is adopted to make the worker's real next state S_j^{t+1} align with the ideal state, defined as follows:

$$r_j^{int,t} = d_{cos}(S_j^t, S_j^{t-1} + G_i^{t-1}), \tag{7}$$

where d_{cos} stands for the cosine similarity, and $r_j^{int,t}$ is regarded as an intrinsic reward for guidance.

The final worker reward function is a linear combination of the extrinsic reward obtained from the environment r_j^{ext} and the intrinsic reward, as shown in Equation 8, where α is a parameter representing the factor of internal reward contributing to the overall worker's reward.

$$r_j^t = \alpha r_j^{int,t} + r_j^{ext,t}. \tag{8}$$

During training, we adopt the loss function in Equation 1 to optimize the parameter θ_W in the workers' Q-network, by substituting the state with $\hat{S}_j^t$ which is obtained from Equation 5 and replacing the reward with r_j^t from Equation 8.

With the introduction of the goal network and the intrinsic reward to our framework, we are able to unify the two independent sub-tasks such that they work towards a shared objective.

4.4 Inter-Signal Communication for Traffic Signal Control

In terms of the traffic signal control sub-task, since traffic flows between neighboring intersections are inter-connected, neighboring intersections' information is essential in the decision making of traffic signals. Inspired from past works such as [30] and [26], we first build a signal-signal graph based on the road network $\mathcal{G}_M = (\mathcal{V}_M, \mathcal{E}_M)$, where $\mathcal{V}_M$ represents traffic lights, and $\mathcal{E}_M$ represents the set of edges i.e., the connections between neighboring intersections in the road network, as illustrated by the solid lines in Figure 5. Then we leverage Graph Convolutional Network (GCN) [36] to improve the communications among signals by aggregating information from neighboring signals to obtain the spatially enhanced representation for each manager agent. Specifically, we define the signal-signal adjacency matrix as $A_M = \{a_{ik}\}_{i,k \in \mathcal{V}_M}$, where $a_{ik} = a_{ki} = 1$ if signals i and k are connected, and $a_{ik} = a_{ki} = 0$ otherwise. The propagation rule of a GCN layer is defined as:

$$H = \sigma\left(D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}XW\right) + X, \tag{9}$$

where $\tilde{A} = A+I$ is the adjacency matrix with self-loops, and D stands for the degree matrix, i.e., $D_{ii} = \sum_k \tilde{A}_{ik}$, W stands for the weight matrix of the layer. In our framework, we adopt two connected GCN layers with activation σ and a dropout. We also involve a skip connection operation by adding the input feature X to the GCN propagation output to improve model stability.

With the GCN module, we are able to compute the spatially aware representations for the traffic signals, then the learnt representation H_i for signal i is fed into the MLP with parameter θ_M to learn the manager's Q-value. Lastly, we provide the loss function for optimizing managers' network in Equation 10 where $\hat{S}_M^t$ is the intention-aware manager state representation in Equation 4.

$$L = E[r_t + \gamma \max_a \tilde{Q}_M(\hat{S}_M^{t+1}, A_M, a|\tilde{\theta}_M, \tilde{W}) - Q(\hat{S}_M^t, A_M, a_M^t|\theta_M, W)]^2. \tag{10}$$

4.5 PseudoCode

The pseudocode of our framework is provided in Algorithm 1.

Algorithm 1: Pseudocode for NavTL

for *episode in 1:N* **do**
  Initialize parameters and reset the environment.
  Define a signal-signal graph $\mathcal{G}_M$ with adjacency $\mathcal{A}_M$.
  for *t in 1:T* **do**
    Construct heterogeneous signal-vehicle graph $\mathcal{G}^t$.
    Compute intention I_M^t according to Equation 3.
    Obtain enhanced $\hat{S}_M^t$ according to Equation 4.
    Obtain enhanced $\hat{S}_W^t$ based on Equation 5.
    Compute the goal vector G_M^t following Equation 6.
    With probability ϵ randomly select action a_W^t for workers; otherwise $a_W^t = max_a(Q_W(\hat{S}_W^t, a; \theta_W))$.
    Execute actions a_W^t and observe S_W^{t+1} and r_W^t.
    Store the transition $\langle S_W^t, I_M^t, a_W^t, r_W^t, S_W^{t+1}\rangle$ in B_W.
    if $len(B_W) > threshold$ **then**
      Sample a batch $\langle S_w, I_m, a_w, r_w, S'_w\rangle$ from B_W.
      Update θ_W based on the TD loss in Equation 1.
    end
    Let $S_W^t = S_W^{t+1}$.
    if $t \% T_G = 0$ **then**
      With probability ϵ randomly select action a_M^t for managers; otherwise $a_M^t = max_a(Q_M(S_M^t, I_M^t, A_M, a; \theta_M))$.
      Execute actions a_M^t and observe S_M^{t+1} and r_M^t.
      Store the transition $\langle S_M^t, I_M^t, a_M^t, r_M^t, S_M^{t+1}\rangle$ in B_M.
      if $len(B_M) > threshold$ **then**
        Sample a batch $\langle S_M, I_M, a_M, r_M, S'_M\rangle$ from B_M.
        Update θ_M based on the TD loss in Equation 10.
      end
      Let $S_M^t = S_M^{t+1}$.
    end
    if $t \% T_{update} = 0$ **then**
      $\tilde{\theta}_W \leftarrow \theta_W$, $\tilde{\theta}_M \leftarrow \theta_M$
    end
  end
end

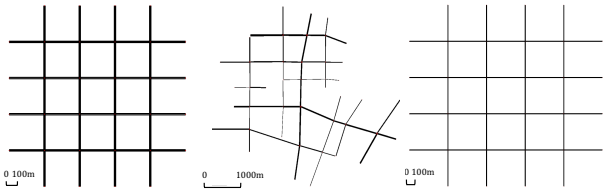
5 EXPERIMENTS

In this section, we introduce the experimental details conducted on three datasets for validating the proposed NavTL framework. Specifically, we first explain the experimental setup including the simulation environment, datasets used, evaluation metrics, compared methods, and model parameter settings, and then we present and discuss the experimental results.

5.1 Simulation

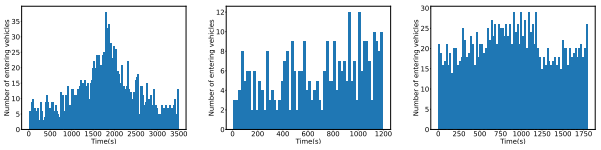
We utilize SUMO[1] (Simulation of Urban MObility), a commonly adopted microscopic and space-continuous multi-modal traffic simulator, to simulate traffic dynamics including movements of vehicles

[1] https://www.eclipse.org/sumo/



(a) **Grid-4x4** (b) **Baoding-19** (c) **Hangzhou-4x4**

Figure 6: The road networks of three datasets.



(a) **Grid-4x4** (b) **Baoding-19** (c) **Hangzhou-4x4**

Figure 7: Distribution of vehicle entrance time.

and phase changes of traffic signals. The road networks data and trip information of human-driven vehicles (HVs) are directly obtained from a road network file and a route file, while the trips of autonomous vehicles are randomly sampled from valid routes using the total number of vehicles entered the network during the last timestep and the penetration rate PR, which is defined as follows:

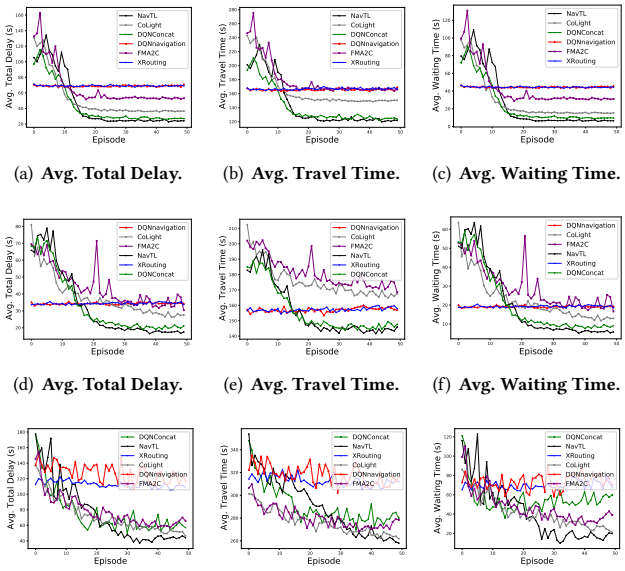
$$PR = \frac{N_{AV}}{N_{AV} + N_{HV}}, \tag{11}$$

where N_{AV} and N_{HV} denote the number of AVs and HVs.

The microscopic behaviors of all vehicles in our simulation are automatically modeled by SUMO via its default built-in car-following model, i.e., the Intelligent Driver Model (IDM), which adaptively controls the vehicles' accelerations based on their velocities and distances to their leading vehicles. For AVs, once we obtain their rerouting actions from the RL model, we immediately recompute the fastest route between the new road that it shall turn to and its predetermined destination road. The computation of the fastest new route is automatically completed through SUMO using the Dijkstra's algorithm [6]. The simulator is able to estimate travel times on different roads based on current traffic conditions. Given the estimated travel times as edge weights and intersections as nodes, the Dijkstra's algorithm finds the shortest path greedily.

5.2 Datasets

We conduct experiments on one synthetic dataset and two real-world datasets. The open-source synthetic dataset, namely *Grid-4x4* [1, 18], is shown in Figure 6(a), where all roads are 300m in length. It contains 16 intersections in total, each controlled by a traffic signal. We extract 1-hour route data for model training and evaluation, and a total of 1,473 vehicles enter the road network during the period. The real-world data *Baoding-19* is obtained from Baoding, Hebei province of China. As shown in Figure 6(b), the *Baoding-19* dataset contains a total of 19 traffic signals, and the road



(a) **Avg. Total Delay.** (b) **Avg. Travel Time.** (c) **Avg. Waiting Time.**

(d) **Avg. Total Delay.** (e) **Avg. Travel Time.** (f) **Avg. Waiting Time.**

(g) **Avg Total Delay.** (h) **Avg. Travel Time.** (i) **Avg. Waiting Time.**

Figure 8: Model performance on Grid-4x4 *(a-c)*, Baoding-19 *(d-f)*, and Hangzhou-4x4 *(g-i)* data.

lengths range from 168.24m to 1234.60m. We take 20-min route data from the route file, there are 615 vehicles entering the road network in total. Another public real-world data, *Hangzhou-4x4* [18], as demonstrated in Figure 6(c), also contains 16 traffic signals, each controlling an intersection. All W-E roads have lengths 800m and all N-S roads have lengths 600m. We extract 30-min route data from the route file, during which 1,660 vehicles enter the network. In this paper, we set the coordination zones to circles centered at the traffic signals with a radius of 150m for all datasets. The entering vehicle distributions of all datasets are shown in Figure 7. It can be found that these datasets represent various traffic scenarios, such as highly congested circumstances (shown by the most congested interval in *Hangzhou-4x4*), increasing travel demand (like the first 2000s of *Grid-4x4*), decreasing travel demand (as the last 1600s of *Grid-4x4*), and sparse and irregular demand (*Baoding-19*).

5.3 Evaluation Metrics

Commonly used evaluation metrics for the traffic signal control task include vehicles' average waiting time, average total number of stops, total throughput, total delay duration, etc [1, 18, 31]. For the navigation task, evaluation metrics such as average travel time and average speed of the controlled vehicles are often adopted to evaluate the model performance [13, 27]. In this paper, we select five metrics to evaluate performance, including average total delay (defined as $t_{real} - t_{expected}$, where t_{real} is the actual trip duration of a vehicle and $t_{expected}$ is the ideal trip time if the vehicle travels at its maximum speed in the traffic network without any traffic restrictions, e.g., red lights at intersections), average waiting time and average travel time of all vehicles (both HVs and AVs), and average travel time and average waiting time of AVs only.

5.4 Compared Methods

We compare the NavTL framework with some representative baselines in the traffic signal control task and the dynamic navigation task, as well as a model on the joint control task, including:

- Colight [30]: a multi-agent DQN model for the traffic signal control task which involves a multi-head Graph Attention Network to aggregate neighborhood observations.
- FMA2C [16]: a multi-agent A2C model for the traffic signal control task which employs the feudal architecture by assigning region managers and intersection workers. For all datasets we manually assign four managers, each controlling a region containing its nearby intersections.
- DQNNavigation [13]: a multi-agent independent DQN model for dynamic navigation across multiple intersections.
- XRouting [27]: an explainable multi-agent PPO model for the navigation task which adapts attention and a transformer module to learn the road-vehicle attributes dynamically.
- DQNConcat: a joint control baseline combining DQN-based traffic signal control and DQN-based vehicle navigation, where the state space of the TL agents and the AV agents follow the state space definitions in [30] and [13] respectively, and each AV agent makes its decision based on an enhanced state which is a concatenation of its own observation and the observation of the upcoming TL.

5.5 Parameter Settings

In our experiments, we set $T_G = 7$ seconds for the green phases, each followed by a default yellow phase with 3 seconds. In terms of NavTL, parameters are shared among all managers and all workers for training efficiency. In the RL setup, we set *threshold* to 100, and *target update frequency* to 10, $\gamma = 0.95$, and $lr = 0.001$. The hidden layer of all MLPs contain 64 neurons, and the GCN layer has hidden shape 128. For the temporally enhanced intention extraction, we set $k = 1$. In terms of internal reward, we set $\alpha = 1$. We adopt the *RMSprop* optimizer [37] for both managers and workers, and we use *ReLU* as the activation functions in our model. For all experiments we train the models for 50 epochs and evaluate over 25 epochs.

5.6 Performance Evaluation

To evaluate the effectiveness of NavTL, we compare its performance with those of the baselines on three datasets, where the penetration rate is set to 30%. The overall performance over different evaluation metrics are shown in Table 1, and the training performances evaluated with different metrics are provided in Figure 8. We provide the training reward curves of managers and workers in Figure 9.

According to the results, we can conclude that our model outperforms either the TSC baselines or the dynamic navigation baselines. On one hand, compared with RL-based TSC methods, NavTL converges faster to the lowest level in terms of total delay, total travel time, and total waiting time. Our model reduces the total delay time by 25.98% on the *Grid-4x4* data and 26.08% on the *Baoding-19* data compared to Colight, and our model achieves 15.75% improvement in terms of the total waiting time on the *Hangzhou-4x4* data compared to FMA2C. On the other hand, comparing with the dynamic navigation baselines, our model can reduce AVs' total waiting time by 53.41% on the *Grid-4x4* data compared with XRouting and 13.97% on the *Hangzhou-4x4* data compared with DQNnavigation, which indicates our model can not only reduce congestion from the macroscopic perspective, but also guide AVs to avoid the temporally congested regions and spend the least time in waiting. More importantly, the effectiveness of the bi-directional framework can be illustrated by the superior performance of NavTL over DQNConcat for nearly all evaluation metrics. For example, our model outperforms DQNConcat by 14.02% and 16.41% on the *Baoding-19* and *Hangzhou-4x4* data respectively in terms of total delay duration. Combined with the learning curves for managers, we can validate that the design of the intention-aware bi-directional propagation mechanism and the goal guidance scheme successfully improves training performance of the intelligent traffic signals, as they are able to receive routing requests from AVs and make more collaborative decisions to guide traffic flows accordingly. Last but not least, the robustness of our model can be shown by the superior performance on all three datasets, which have different vehicle entrance distributions as illustrated in Figure 7. Thus, our model can handle various traffic scenarios effectively.

5.7 Ablation Study

We further evaluate the effectiveness of individual components of NavTL by investigating the following variants:

- w/o-tsc: In this model, traffic lights operate with a fixed plan, while AVs are controlled by RL.
- w/o-nav: In this model, AVs follow their default path, while traffic signals are controlled by RL.
- w/o-hrl: removing the hierarchical RL architecture. Here, both traffic signals and AVs are controlled by RL, but they perform as independent tasks without collaboration.
- w/o-gnn: removing graph neural networks. In this model, the inter-signal and signal-vehicle communications are disabled.

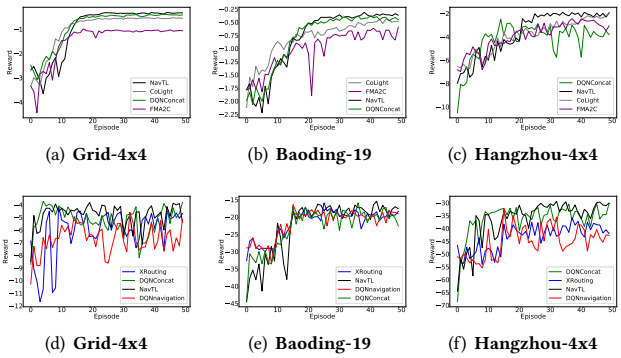
Due to space limitation, we provide the results of the ablation study on the *Grid-4x4 30min* data in Table 2. From the results we can conclude that both the traffic signal control module and the navigation module are the most important components of our model. Moreover, the feudal architecture along with the graph neural networks are also effective, since they build the connections among manager agents and worker agents, promoting signal-signal and signal-vehicle coordinations.

5.8 Performance at Different Penetration Rates

In addition, we conduct a study on the impact of different AV penetration rates on the improvements in traffic efficiency. The results on the first 30 minutes of the *Grid-4x4* data with three evaluation metrics are shown in Figure 10, from which we can conclude that our model performance improves as PR increases from 10% to 50%, with the most substantial difference between 10% and 30%. This is reasonable since when the number of AVs is small, the workers' network might not be trained with sufficient experiences. As PR increases, more vehicles are able to make rerouting decisions, contributing to the total travel efficiency.

Table 1: The Overall Performance of Different Models on Three Datasets.

Data	Method	Total Delay (s)	Travel Time (s)	AV Travel Time (s)	Waiting Time (s)	AV Waiting Time (s)
Grid-4x4	Colight	36.61 (+25.98%)	149.80 (+16.67%)	161.20 (+70.26%)	15.33 (+37.70%)	19.44 (+92.70%)
	FMA2C	53.08 (+48.94%)	165.96 (+24.78%)	176.33 (+72.81%)	31.22 (+69.41%)	35.66 (+96.02%)
	XRouting	69.07 (+60.76%)	165.83 (+24.72%)	48.41 (+0.97%)	44.68 (+78.63%)	3.03 (+53.41%)
	DQN navigation	68.92 (+60.68%)	167.03 (+25.26%)	57.57 (+16.73%)	44.25 (+78.42%)	4.09 (+65.28%)
	DQNConcat	30.51 (+11.18%)	125.43 (+0.48%)	68.50 (+30.01%)	9.75 (+2.05%)	1.52 (+6.58%)
	NavTL	**27.10**	**124.83**	**47.94**	**9.55**	**1.42**
Baoding-19	Colight	27.72 (+26.08%)	167.10 (+12.12%)	179.29 (+53.33%)	13.17 (+31.66%)	15.89 (+70.80%)
	FMA2C	34.03 (+39.79%)	172.43 (+14.84%)	186.69 (+55.18%)	15.89 (+43.36%)	23.21 (+80.01%)
	XRouting	34.42 (+40.47%)	157.99 (+7.06%)	**81.13** (-3.13%)	19.22 (+53.17%)	6.44 (+27.95%)
	DQN navigation	34.73 (+41.00%)	158.54 (+7.38%)	82.59 (-1.31%)	19.53 (+53.92%)	6.62 (+29.91%)
	DQNConcat	23.83 (+14.02%)	148.38 (+1.04%)	85.55 (+2.02%)	**8.58** (-4.90%)	5.00 (+7.20%)
	NavTL	**20.49**	**146.84**	83.67	9.00	**4.64**
Hangzhou-4x4	Colight	59.28 (+7.91%)	273.34 (+2.39%)	298.21 (+33.77%)	51.98 (+6.85%)	81.31 (+60.61%)
	FMA2C	63.14 (+13.54%)	279.52 (+4.55%)	301.01 (+34.39%)	57.47 (+15.75%)	88.75 (+63.91%)
	XRouting	78.29 (+30.27%)	304.41 (+12.36%)	251.79 (+21.56%)	68.75 (+29.57%)	36.79 (+12.94%)
	DQN navigation	74.66 (+26.88%)	307.90 (+13.35%)	256.80 (+23.09%)	64.21 (+24.59%)	37.23 (+13.97%)
	DQNConcat	65.31 (+16.41%)	291.02 (+8.32%)	223.27 (+11.43%)	53.26 (+9.09%)	34.90 (+8.22%)
	NavTL	**54.59**	**266.80**	**197.50**	**48.42**	**32.03**



(a) **Grid-4x4** (b) **Baoding-19** (c) **Hangzhou-4x4**

(d) **Grid-4x4** (e) **Baoding-19** (f) **Hangzhou-4x4**

Figure 9: Training curves for managers *(a-c)* and workers *(d-f)*.

Table 2: Ablation Studies.

	NavTL	w/o-nav	w/o-tsc	w/o-hrl	w/o-gnn
Total Delay (s)	**23.00**	36.44	68.48	37.76	28.33
Travel Time (s)	**112.89**	149.84	165.24	141.65	117.54
AV Travel Time (s)	**45.89**	161.24	48.83	97.35	50.54
Waiting Time (s)	**7.56**	15.06	44.09	17.00	11.91
AV Waiting Time (s)	**1.32**	19.14	3.10	11.46	3.59

5.9 Case Study

Moreover, we conduct a case study at $PR = 50\%$ and provide a demonstration[2] tracking an AV's trajectory. In the video, the AV under control originally plans to turn left, waiting on the left-turning lane, but decides to go straight instead due to the congestion in the left outgoing road. This demo highlights the effectiveness of our framework in mitigating severe traffic congestion at intersections by redirecting autonomous vehicles (AVs) to less congested roads.

[2]https://youtu.be/iJo544PGgYs

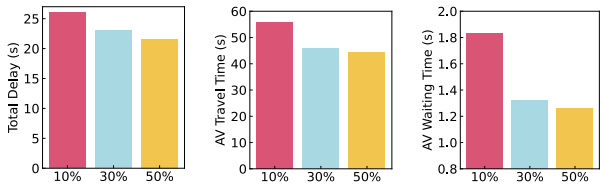


Figure 10: Performance of total delay *(left)*, AV average travel time *(middle)*, and AV average waiting time *(right)* at penetration rates 10%, 30%, and 50%, respectively.

6 CONCLUSION AND FUTURE WORK

In this paper, we proposed *NavTL*, a graph-enhanced bi-directional hierarchical reinforcement learning framework to collaboratively control traffic signal phases and navigation directions of autonomous vehicles. The results on three datasets demonstrate the effectiveness of our framework in terms of minimizing overall congestion level and improving travel efficiency. This is the first work that studies this joint control task in mixed traffic scenarios using learning-based method, and it is a stepping stone for the implementation of Cooperative Vehicle Infrastructure System in the real world. Potential future research directions include using RL to jointly control the navigation directions as well as other driving behaviors such as accelerations and lane-changes of AVs at intelligent intersections.

7 ACKNOWLEDGMENTS

This research was supported in part by grants from the National Natural Science Foundation of China (Grant No. 61960206008).

REFERENCES

[1] James Ault and Guni Sharon. 2021. Reinforcement learning benchmarks for traffic signal control. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

[2] Andrew G Barto and Sridhar Mahadevan. 2003. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems* 13, 1-2 (2003), 41–77.

[3] Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. 2022. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[4] Seung-Bae Cools, Carlos Gershenson, and Bart D'Hooghe. [n. d.]. *Self-Organizing Traffic Lights: A Realistic Simulation*. Springer London, London, 45–55. https://doi.org/10.1007/978-1-4471-5113-5_3

[5] Xuan Di and Rongye Shi. 2021. A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. *Transportation research part C: emerging technologies* 125 (2021), 103008.

[6] E. W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1, 1 (1959), 269–271. https://doi.org/10.1007/BF01386390

[7] Wade Genders and Saiedeh Razavi. 2019. An open-source framework for adaptive traffic signal control. *arXiv preprint arXiv:1909.00395* (2019).

[8] Yuanzhe Geng, Erwu Liu, Rui Wang, Yiming Liu, Weixiong Rao, Shaojun Feng, Zhao Dong, Zhiren Fu, and Yanfen Chen. 2021. Deep reinforcement learning based dynamic route planning for minimizing travel time. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 1–6.

[9] Jiaying Guo, Long Cheng, and Shen Wang. 2022. CoTV: Cooperative Control for Traffic Light Signals and Connected Autonomous Vehicles using Deep Reinforcement Learning. *arXiv preprint arXiv:2201.13143* (2022).

[10] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE transactions on systems science and cybernetics* 4, 2 (Jul 1968), 100–107. https://doi.org/10.1109/TSSC.1968.300136

[11] Matthias Hutsebaut-Buysse, Kevin Mets, and Steven Latré. 2022. Hierarchical reinforcement learning: A survey and open research challenges. *Machine Learning and Knowledge Extraction* 4, 1 (2022), 172–221.

[12] Jiarui Jin, Ming Zhou, Weinan Zhang, Minne Li, Zilong Guo, Zhiwei Qin, Yan Jiao, Xiaocheng Tang, Chenxi Wang, Jun Wang, et al. 2019. Coride: joint order dispatching and fleet management for multi-scale ride-hailing platforms. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1983–1992.

[13] Songsang Koh, Bo Zhou, Hui Fang, Po Yang, Zaili Yang, Qiang Yang, Lin Guan, and Zhigang Ji. 2020. Real-time deep reinforcement learning based vehicle navigation. *Applied soft computing* 96 (Nov 2020), 106694. https://doi.org/10.1016/j.asoc.2020.106694

[14] Meiqi Liu, J Zhao, SP Hoogendoorn, and M Wang. 2022. An optimal control approach of integrating traffic signals and cooperative vehicle trajectories at intersections. *Transportmetrica B: transport dynamics* 10, 1 (2022), 971–987.

[15] Meiqi Liu, Jing Zhao, Serge Hoogendoorn, and Meng Wang. 2022. A single-layer approach for joint optimization of traffic signals and cooperative vehicle trajectories at isolated intersections. *Transportation research part C: emerging technologies* 134 (2022), 103459.

[16] Jinming Ma and Feng Wu. 2020. Feudal multi-agent deep reinforcement learning for traffic signal control. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 816–824.

[17] Jinming Ma and Feng Wu. 2022. Feudal Multi-Agent Reinforcement Learning with Adaptive Network Partition for Traffic Signal Control. (May 27, 2022). https://doi.org/10.48550/arxiv.2205.13836

[18] Hao Mei, Xiaoliang Lei, Longchao Da, Bin Shi, and Hua Wei. 2022. LibSignal: An Open Library for Traffic Signal Control. *arXiv preprint arXiv:2211.10649* (2022).

[19] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. 2018. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems* 31 (2018).

[20] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. 2021. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–35.

[21] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[22] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112, 1-2 (1999), 181–211.

[23] Pravin Varaiya. 2013. Max pressure control of a network of signalized intersections. *Transportation research. Part C, Emerging technologies* 36 (Nov 2013), 177–195. https://doi.org/10.1016/j.trc.2013.08.014

[24] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*. PMLR, 3540–3549.

[25] Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M Bayen. 2018. Benchmarks for reinforcement learning in mixed-autonomy traffic. In *Conference on robot learning*. PMLR, 399–409.

[26] Yanan Wang, Tong Xu, Xin Niu, Chang Tan, Enhong Chen, and Hui Xiong. 2022. STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control. , 2228-2242 pages. https://doi.org/10.1109/TMC.2020.3033782

[27] Zheng Wang and Shen Wang. Jan 01, 2022. XRouting: Explainable Vehicle Rerouting for Urban Road Congestion Avoidance using Deep Reinforcement Learning. The Institute of Electrical and Electronics Engineers, Inc. (IEEE), Piscataway. https://doi.org/10.1109/ISC255366.2022.9922404

[28] F. V. Webster. 1958. *Traffic Signal Settings*.

[29] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1290–1298.

[30] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1913–1922.

[31] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2021. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter* 22, 2 (2021), 12–18.

[32] Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. 2017. Flow: Architecture and benchmarking for reinforcement learning in traffic control. *arXiv preprint arXiv:1710.05465* 10 (2017).

[33] Han Wu, Kun Zhang, Guangyi Lv, Qi Liu, Runlong Yu, Weihao Zhao, Enhong Chen, and Jianhui Ma. 2019. Deep technology tracing for high-tech companies. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1396–1401.

[34] Zhongxia Yan and Cathy Wu. Sep 19, 2021. Reinforcement Learning for Mixed Autonomy Intersections. IEEE, Piscataway, 2089–2094. https://doi.org/10.1109/ITSC48978.2021.9565000

[35] Le Zhang, Ding Zhou, Hengshu Zhu, Tong Xu, Rui Zha, Enhong Chen, and Hui Xiong. 2021. Attentive heterogeneous graph embedding for job mobility prediction. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2192–2201.

[36] Weijia Zhang, Hao Liu, Jindong Han, Yong Ge, and Hui Xiong. 2022. Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2471–2481.

[37] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. 2019. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11127–11135.

[38] Yangang Zou, Fangfang Zheng, Zhichen Fan, and Youhua Tang. 2022. Integrated control of traffic signal and automated vehicles for mixed traffic: Platoon-based bi-level optimization approach. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2107–2113.