This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

FELight: Fairness-Aware Traffic Signal Control via Sample-Efficient Reinforcement Learning

Xinqi Du, Ziyue Li, Cheng Long, Yongheng Xing, Philip S. Yu, *Life Fellow, IEEE,* and Hechang Chen

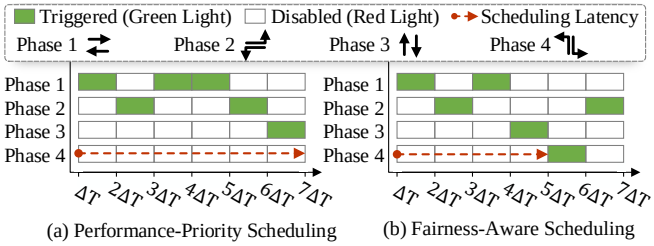
Abstract—Traffic congestion is becoming an increasingly prominent problem, and intelligent traffic signal control methods can effectively alleviate it. Recently, there has been a growing trend of applying reinforcement learning to traffic signal control for adaptive signal scheduling. However, most existing methods focus on improving traffic performance while neglecting the issue of scheduling fairness, resulting in long waiting time for some vehicles. Some works attempt to address fairness issues but often sacrifice transport performance. Furthermore, existing methods overlook the challenge of sample efficiency, especially when dealing with diversity-limited traffic data. Therefore, we propose a Fairness-aware and sample-Efficient traffic signal control method called FELight. Specifically, we first design a novel fairness metric and integrate it into decision process to penalize cases with high latency by setting a threshold for activating the fairness mechanism. Theoretical comparison with other fairness works proves why and when our fairness could bring advantages. Moreover, counterfactual data augmentation is employed to enrich interaction data, enhancing the sample efficiency of FELight. Self-supervised state representation is introduced to extract informative features from raw states, further improving sample efficiency. Experiments on real traffic datasets demonstrate that FELight provides relatively fairer traffic signal control without compromising performance compared to state-of-the-art approaches. Our codes are available at https://github.com/dxnbbsw/FELight.

Index Terms—Traffic signal control, reinforcement learning, fairness-aware scheduling, sample efficiency, multi-intersections.

1 Introduction

Traffic signal control (TSC) plays a crucial role in mitigating increasingly prominent traffic congestion by coordinating traffic flows at intersections [1]. There has been a proliferation of research on traffic signal control, broadly classified into two categories: traditional methods [2], [3] and reinforcement learning (RL)-based methods [4]–[6].

Specifically, traditional methods mainly rely on predefined traffic rules (e.g., FixedTime [2], SCATS [7], SCOOT [8],) or expert-defined actuated rules (e.g., MaxPressure [3] and SOTL [9]). However, these methods heavily depend on handcrafted rules and numerous assumptions without accounting for traffic dynamics. Recently, given that traffic signal control can be formulated as a sequential decision-making problem, RL-based methods have emerged as an effective approach [10], [11]. The standard is either to formulate one agent to control all the traffic lights, such as MPLight [12], or each agent to control each light, thus in a multi-agent RL manner, such as CoLight [13] and IntelliLight [4]. The state and reward are usually traffic-related indexes such as flow, queue length, and delay/wait time, and the action is usually to choose which traffic signal phases (combinatorial of two non-conflicting traffic signal lights, Figure 1) to be released in the next time stamp. The recent years also observed new techniques being incorporated into RL-based TSC, such as graph attention networks (GATs) to coordinate multi-lights [13], meta reinforcement learning to increase the model's transferability in new unseen scenarios [14], [15].



(a) Performance-Priority Scheduling (b) Fairness-Aware Scheduling

Fig. 1: Illustration of scheduling fairness in traffic signal control. Assume that traffic signal phases 1 and 2 are with heavy traffic loads, and phases 3 and 4 with light loads. (a) Performance-priority scheduling frequently triggers the phase with heavy loads. (b) Fairness-aware scheduling avoids prolonged scheduling delays. The red arrows indicate the time interval of the suspended traffic signal phase.

Compared to traditional methods [2] based on predefined rules, RL-based TSC methods are capable of dynamically scheduling traffic signal phases and thus have proved their superiority in terms of achieving higher traffic efficiency. Despite the superiority of RL-based methods, most RL-based methods still suffer from the inherent problems of (1) *fairness* and (2) *sample efficiency*.

Fairness in TSC generally refers to ensuring equitable treatment and allocation of resources among different road users at intersections. In this paper, we define fairness as the equitable waiting time of vehicles before they are released. An unfair signal scheduling could lead to extremely long waiting times (comparing Figure 1a and 1b) for some

Xinqi Du and Hechang Chen are with the School of Artificial Intelligence and the Engineering Research Center of Knowledge-Driven Human–Machine Intelligence, Jilin University, China. Xinqi Du is also with the School of Computer Science and Engineering, Nanyang Technological University.
Ziyue Li is with the Information System Department, WiSo Faculty, University of Cologne, Germany.
Cheng Long is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.
Yongheng Xing is with the College of Computer Science and Technology, Jilin University, Changchun 130012, China.
Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607 USA.

0000–0000/00$00.00 © 2021 IEEE
Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

individual vehicles. The potential unfairness of existing works [3], [12], [13] originates from the fundamental RL's reward-oriented mechanism. Specifically, they give a higher releasing priority to the more jammed directions since this would naturally gain more rewards, which leaves the vehicles in less jammed directions over-waiting. As shown in Figure 1a, traffic signal phase 1 and phase 2 with heavy loads are frequently scheduled, while phase 4 is delayed for a long term, which causes unfairness. Some methods [16]–[18] provide fair scheduling (Figure 1b) by activating fairness mechanisms once an unfair case occurs, which often sacrifices overall performance. Thus, the real challenge lies in how to balance both performance and fairness.

Another unsolved problem ignored by most RL-based TSC methods is *sample efficiency*. Existing RL-based TSC methods majorly rely on simulators, yet there are only a few simulation environments available (including SUMO [19], CityFlow [20], VISSIM [21]) and less than a dozen real-world datasets available (including New York [12], Atlanta [22], Hangzhou [13], etc.), whose rollout periods last only 3600 seconds in reality [22]. The amount of data is small, which results in restricted data diversity. An intuitive solution is to simply create more real-world datasets, e.g., Jiang [15] creates two new datasets based on two Chinese cities. However, the cost is huge [15] due to the following reasons: 1) collecting traffic data requires the deployment of sensors on a large scale with extra overhead; 2) the process of collecting and cleaning traffic data is time-consuming. Therefore, a sample-efficient RL model is essential and economical since it would learn effectively and achieve good performance using fewer interactions or samples with the environment.

Motivated by these discussions, we propose a **F**airness-aware and sample-**E**fficient traffic signal control method called **FELight**, which takes fairness scheduling into account with guaranteed performance, and combines counterfactual data augmentation and self-supervised state representation learning to alleviate the aforementioned issues. Firstly, to balance fairness and performance, a fairness metric F_i is devised to evaluate the fairness of the traffic signal phase schedule. Meanwhile, F_i is incorporated into the state and reward calculation as a regularization, which **only** penalizes the unfair cases with higher latency, unlike existing methods whose fairness penalty is always active. Secondly, to encourage sample efficiency, we make advancements in two aspects: 1) extracting the underlying structural information of the data to generate more diverse data for training; 2) improving the utilization of current interaction data. Specifically, counterfactual data augmentation is employed in which the structural causal model captures the dynamics of traffic signal control to generate more interaction data. In this way, it enriches the diversity of the interactive data and avoids endless exploration. In addition, self-supervised state representation is utilized to learn a better state embedding, facilitating the downstream Q-value estimation. The main contributions of this work are summarized as follows:

1) We propose a fairness-aware and sample-efficient traffic signal control method called FELight, which schedules the traffic signal phases fairly without compromising performance.
2) A fairness metric is newly devised to measure how long a phase is delayed in scheduling, which is used to penalize cases with unfair scheduling. To the best of our knowledge, it is the first effort in the literature to consider fairness and performance in traffic signal control jointly. Moreover, we also theoretically compare all the existing fairness designs and prove why and when our fairness could bring advantages.
3) FELight deploys counterfactual data augmentation to generate more data, enriching the interaction data diversity. Self-supervised state representation is also employed, introducing extra training loss to learn state embedding efficiently. Both of them are conducive to boosting the sample efficiency of FELight.
4) We conduct extensive experiments on real-world traffic datasets and compare FELight with traditional and RL-based traffic signal control methods. The experimental results demonstrate that FELight provides a fair and efficient traffic signal control strategy.

The structure of this paper is organized as follows. Section 2 reviews various traffic signal control methods and techniques for sample-efficient reinforcement learning. The common terminologies of traffic signal control are summarized in Section 3. The details of FELight are shown in Section 4. Section 5 describes the conducted experiments and provides a theoretical analysis of the results. Finally, Section 6 draws the conclusion and discusses potential future work.

2 Related Work

In this section, we review the existing works for traffic signal control and sample-efficient reinforcement learning.

2.1 Traffic Signal Control

Traffic signal control is a significant but sophisticated problem in transportation, aiming to maximize traffic network throughput. A substantive of literature has been proposed for traffic signal control, categorized into two groups: traditional approaches and RL-based methods.

2.1.1 Traditional Traffic Signal Control

Traditionally, traffic signal control obeys the predefined rules designed by transportation domain experts. For example, FixedTime [2] cyclically schedules the phases following a predefined sequence, regardless of traffic loads (heavy or not). Its main strength is that it is simple to implement. However, it lacks the flexibility to adapt to dynamic traffic systems and is prone to traffic congestion. Furthermore, some actuated control methods are proposed to improve flexibility, such as MaxPressure [3] and SOTL [9], which trigger the phases based on predefined rules. MaxPressure incorporates the concept of traffic pressure to control traffic signals. SOTL changes the phase according to the waiting vehicle count. However, these approaches require expert knowledge and are challenging for complex and dynamic traffic signal control tasks.

2.1.2 RL-based Traffic Signal Control

Recently, given that traffic signal control is a sequential decision process, reinforcement learning has been used for

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

TABLE 1: Comparison of reward designs among different w/ fairness traffic signal control methods.

Method	Equation ($q(l)$ is the queue length)	Description (ω and α are trade-off hyperparameter)	Drawback
FIT [16]	$r=-\sum_l q_t(l)/(\bar{L}_l(t)+\epsilon)$ $\bar{\Psi}_l(t)=(1-1/\omega)\bar{\Psi}_l(t-1)+\Psi_l(t)/\omega$	$\Psi_l(t)$ is current throughput for lane l; $\bar{\Psi}_l(t)$ is the average throughput, iteratively updated; ϵ is a negligible positive number to ensure the denominator is not zero.	It is sensitive to lane's throughput, especially for those lanes with low throughput.
DFC [18]	$r=-\sum_{n\in N_t}(1+\alpha(2d_n(t)-1))$	N_t is the number of vehicles at t and n denotes vehicle index; $d_n(t)$ is partial waiting time of vehicle n up to so far t; r is a function of the sum of vehicles' waiting time.	The design of the reward causes it to be a less than ideal trade-off between performance and fairness.
TFC [18]	$r=-\sum_l q_t(l)-\omega\lvert\xi\rvert$ $\xi(t)=\xi(t-1)+\beta(t)(\frac{\Psi_{l_1}(t)}{\psi_{l_1}}-\frac{\Psi_{l_2}(t)}{\psi_{l_2}})$	Ψ_l is current throughput; ψ_l is predefined fairness weight to guarantee phases are scheduled proportionally; ξ is the deviation of weighted throughput. $\beta(t)=1$ if all directions have incoming traffic, otherwise 0.	It is sensitive to the case with low road throughput. It is also required to pre-define the weights.
FELight	$r=-\sum_l q_t(l)-\omega\times F$	F is fairness based on extra waiting time (Definition 7)	– –

dynamic traffic signal control. Various RL-based traffic signal control methods have been proposed to improve the performance of traffic systems [4], [23]–[25]. According to whether fairness is taken into account, they are roughly divided into two categories: 1) w/o fairness and 2) w/ fairness.

The first group of methods (w/o fairness) focuses on improving transport efficiency. For example, IntelliLight [4] uses DQN to adjust traffic lights according to real-time traffic flow, improving traffic efficiency. PressLight [5] integrates the traffic pressure theory into reinforcement learning reward design to maximize the traffic network throughput. MPLight [12] proposes a decentralized deep reinforcement learning method for large-scale traffic signal control. CoLight [13] incorporates the graph attention mechanism in multi-intersection traffic signal control tasks, achieving cooperation among intersections. AttendLight [26] and GeneraLight [27] are designed to improve generalizations to different intersection structures and traffic flow distributions. However, most existing works [12], [13] primarily focus on optimizing the performance of traffic signal control, overlooking the fairness in phase scheduling.

Another group of methods (w/ fairness) is devoted to providing fair scheduling. For example, FIT [16] utilizes proportional fair scheduling for communications, measuring scheduling fairness based on the ratio of queue length to average throughput. TFC [18] uses weighted fairness by assigning pre-determined weights to traffic flows. The throughput of different roads is divided by their weights. Deviations from all ratios are used to measure the overall scheduling fairness. DFC [18] calculates the partial waiting time so far to measure fairness. FSPPO [28] controls traffic signals in a cyclic manner, avoiding the problem of long waiting times for individual vehicles by controlling the duration of each signal and achieving fair scheduling. FairLight [17] controls traffic signals in an acyclic, variable-duration manner. It uses the ratio of vehicle waiting time to driving time to measure the fairness of signal scheduling.

While some works [16], [18] attempt to improve fairness, they compromise the performance as we could observe in our experiments (in Table 3 and Figure 7). Besides, we will scrutinize the different formulations of fairness in Table 1. Specifically, we compare our proposed method (FELight) with RL-based methods w/ fairness in terms of reward design to highlight its advantages, and the details are presented in Table 1. Notably, FELight introduces a novel fairness concept, i.e., fairness based on extra waiting time F_i, **only** penalizing cases characterized by higher upper bounds on latency, ensuring a fair traffic signal schedule of the entire system while minimizing the impact on overall performance. Its details will be introduced in Section 4.

In addition, RL-based methods require diverse interaction data for training, making it difficult for data-limited traffic signal control tasks. Thus, we propose a traffic signal control method with fairness consideration and sample efficiency.

2.2 Sample-Efficient Reinforcement Learning

Sample efficiency is an urgent issue for RL-based TSC methods with limited data diversity. Counterfactual reinforcement learning and self-supervised representation learning are capable of improving sample efficiency.

2.2.1 *Counterfactual Reinforcement Learning*

Counterfactual learning is a method that learns from limited experience data and estimates the potential outcome of another action that was not taken. Recently, it has been widely applied for decision-making tasks in healthcare [29], finance, etc. CTRL [30] proposes a data-efficient reinforcement learning model for healthcare by leveraging counterfactual data augmentation. MABUC [31] estimates counterfactual quantities under heterogeneous data-collection modes to address the challenges posed by unobserved confounders in multi-arm bandit decision-making tasks. The work [29] proposes to extract the counterfactual trajectories from the original observed data based on the assumptions on the underlying probabilistic data generating mechanism.

In general, counterfactual thinking can generate hypothetical transitions that did not actually occur, which enriches the agents' experiences. Thus, we incorporate the counterfactual data augmentation into RL-based traffic signal control approach, which is beneficial to improve sample efficiency and avoid potential-risk actions.

2.2.2 *Self-supervised Representation Learning*

Self-supervised learning has emerged as a popular technique to extract meaningful feature representations from the original data without explicit supervision. It has succeeded in various domains, including computer vision [32]–[34], natural language processing [35], and smart mobility [36]. Besides, self-supervised learning is widely used as an auxiliary task in reinforcement learning to improve sample

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

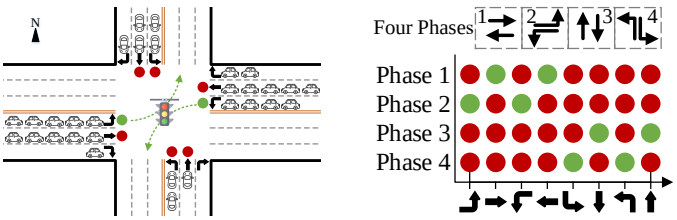
efficiency and model performance. For example, M-CURL [37] and C2E-MARL [38] employ contrastive learning to boost the sample efficiency in RL tasks. ATC [39] decouples representation learning from reinforcement learning by using a self-supervised auxiliary task, resulting in improved sample efficiency and performance.

Generally, state representations are critical to reinforcement learning tasks, particularly complex ones. Therefore, considering the high-dimensional state in multi-intersection traffic signal control, we incorporate self-supervised learning to boost the state feature representation learning by providing additional training signals, which enhances the sample efficiency in RL-based traffic signal control methods.

To the best of our knowledge, FELight is the first work that raises the awareness of sample efficiency in the TSC task and proposes counterfactual augmentation and bootstrapping-based self-supervised learning into TSC.

3 Preliminary

This section summarizes the relevant definitions used in the traffic signal control problem. The notations used in this problem are given in Table 2. The details are as follows:



(a) Traffic signal control (b) Phase configuration

Fig. 2: Illustration of traffic signal control and traffic signal phase configuration.

Definition 1 (Intersection)**.** The intersection I_i is a point where multiple roads meet. In our work, four-direction roads are considered: east (*E*), south (*S*), west (*W*), and north (*N*). Figure 2a shows that an intersection equips with a set of traffic signal phases, controlled by an RL-based agent.

Definition 2 (Road and Lane)**.** One road consists of several lanes l with vehicles running on it. The sets of incoming and outgoing lanes of intersection I_i are denoted as L_i^{in} and L_i^{out}, respectively. Incoming lanes $l_i^{in} \in L_i^{in}$ are those that lead to the intersection I_i, while outgoing lanes $l_i^{out} \in L_i^{out}$ are those that leave the intersection I_i.

Definition 3 (Traffic Movement)**.** Traffic movement m is defined as the traffic traveling across an intersection from one incoming lane to one outgoing lane. $m = (l^{in}, l^{out})$ denotes a traffic movement from lane l^{in} to lane l^{out}, $l^{in} \in L^{in}$ and $l^{out} \in L^{out}$. There are three different movements, i.e., straight, left, and right, in four directions of E, S, W, and N. Usually, right-movement doesn't need to be signal-controlled (we consider the context of a right-driving country in this paper). The set of all traffic movements is denoted as M.

Definition 4 (Traffic Signal Phase)**.** Traffic signal phase p is defined as a set of permitted traffic movements. Typically, going straight and turning left are controlled by signal phases. Figure 2b shows the phase set, which consists of four combinatorial phases, formulated as $P = \{p_{straight}^{WE}, p_{left}^{WE}, p_{straight}^{NS}, p_{left}^{NS}\}$, where $p_{straight}^{WE}$ denotes the traffic movements going straight from west and from east are permitted; p_{left}^{WE} denotes turning left from west and from east are permitted. The minimum duration for each phase is Δt, which is also a minimal time interval for phase decision.

Definition 5 (Queue Length)**.** The queue length $q(l)$ is the number of waiting vehicles in lane l. The queue length of an intersection I_i is the sum of the total queue length of the incoming lanes l in the intersection, formulated as follows:

$$q_i = \sum q(l), l \in L_i^{in}. \tag{1}$$

Problem 1 (Traffic Signal Control)**.** This paper investigates the problem of traffic signal control with K-intersections [22] ($K > 1$). Specifically, the problem is to decide on an intersection for arranging the traffic signal phase every Δt to minimize the average traveling time over all time periods. The RL agent controls the traffic signal light equipped with each intersection. It obtains the state s from the environment and takes action a to control the traffic signal.

Notation	Description
I_i	The intersection I_i
K	The number of intersections in the road network
l_i^{in}	Incoming lane for intersection I_i
l_i^{out}	Outgoing lane for intersection I_i
L_i^{in}	The set of the incoming lanes for intersection I_i
L_i^{out}	The set of the outgoing lanes for intersection I_i
$m = (l^{in}, l^{out})$	Traffic movement from lane l^{in} to lane l^{out}
M	The set of all traffic movements
p	Traffic signal phase
P	The set of the traffic signal phases
$q(l)$	Queue length for a lane l
q_i	Queue length for the intersection I_i
$\theta_i, \bar{\theta}_i$	Learnable parameters for Q-network and target Q-network
ϕ_o, ϕ_m	Learnable parameters for state encoder and target state encoder
ω	Trade-off parameter for tuning the importance of fairness and performance in the reward function
λ	Trade-off parameter for tuning the importance of self-supervised model in the loss function

TABLE 2: The notations used in traffic signal control.

4 Methodology

This section will introduce the proposed FELight in detail. First, a new fairness metric is devised to measure whether the traffic signal phase is scheduled fairly, and it is incorporated into FELight. Then, FELight employs counterfactual thinking for data augmentation to enrich the diversity of interaction data, thereby boosting its sample efficiency. Finally, self-supervised state representation is deployed to improve the traffic state representation for our FELight. The overall architecture of FELight is illustrated in Figure 3.

4.1 Fairness-Aware Scheduling

Most existing RL-based traffic signal control methods pursue maximizing the throughput by incorporating various

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

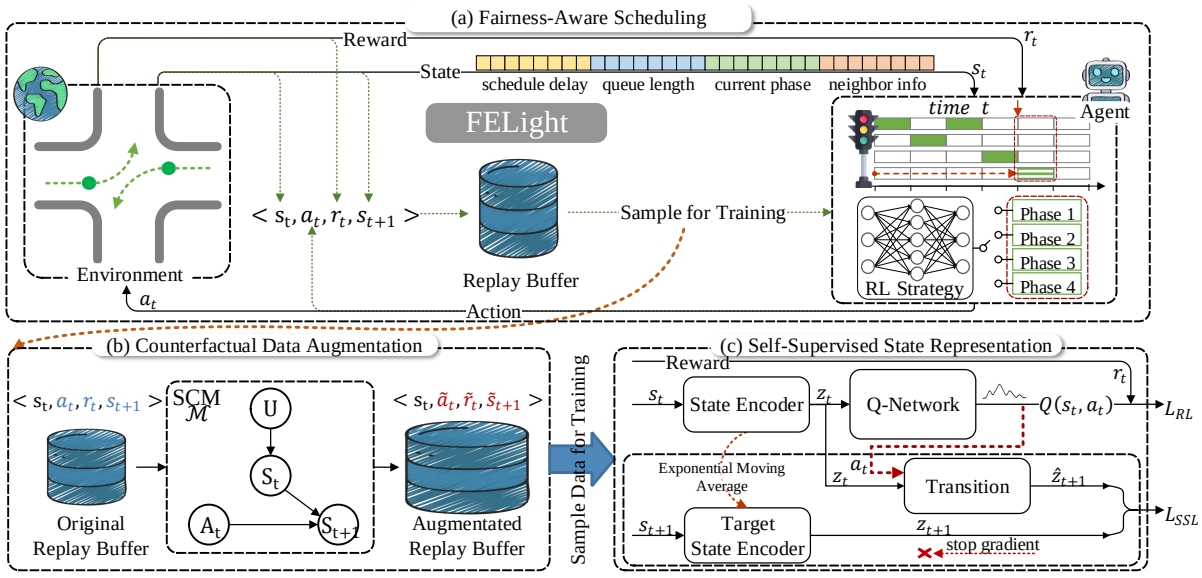


Fig. 3: Illustration of FELight, which comprises (a) Fairness-Aware Scheduling, (b) Counterfactual Data Augmentation and (c) Self-Supervised State Representation. In detail, (a) formulates the traffic signal control as a Markov Decision Process and fairly controls the traffic signal phase; (b) enriches the training samples by counterfactual thinking into the original interaction data; and (c) enhances the state encoder representation capability through a self-supervised auxiliary task.

traffic indicators, such as queue length, pressure, and the number of waiting vehicles. However, the aspect of fairness in scheduling the traffic signal phase has yet to receive adequate attention, leading to the traffic signal phase with low traffic loads being delayed in scheduling. While some works attempt to address fairness, they often come at the cost of compromising performance. Thus, to ensure the traffic signal phase is scheduled fairly, we devise fairness concepts specific to traffic signal control and incorporate them into the decision-making process.

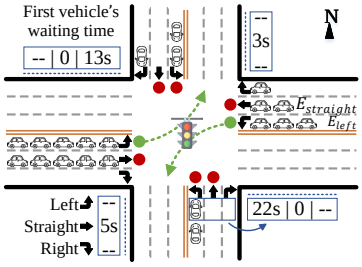
Definition 6 (Phase Waiting Time)**.** The phase waiting time T_i^p, is defined as the time interval measured from its last set time t_{set} to the current time t_{cur}. It is counted only when there exists a waiting vehicle on the incoming lane permitted by traffic signal phase p (the superscript and subscript are omitted for brevity). T_i^p is defined as follows:

$$T_i^p = \begin{cases} 0 & \text{if } p \text{ is scheduled,} \\ \max_m \sum_{t=t_{set}}^{t_{cur}} \mathbb{1}(\mathcal{V}_{m_in}^t \neq \emptyset) & \text{otherwise.} \end{cases} \tag{2}$$

where m is the traffic movement controlled by phase p. $\mathcal{V}_{m_in}^t$ denotes the set of waiting vehicles in incoming lanes l^{in} associated with traffic movement m (**Definition** 3) at time t. The indicator function $\mathbb{1}(\cdot)$ judges whether $\mathcal{V}_{m_in}^t$ is empty, i.e., whether there exists any waiting vehicle. Phase waiting time essentially calculates the accumulated waiting time when there is vehicle in a non-green phase. The calculation of T_t^p is illustrated in Figure 4.

According to the proposed **Definition** 6, a larger value of T_i^p indicates a greater delay, i.e., an unfair phenomenon in scheduling the phase p, which in turn implies higher scheduling priority for the phase p.

Definition 7 (Fairness based on *Extra* Waiting Time)**.** The fairness F_i based on extra waiting time evaluates the fairness of phase schedules among different traffic signal phases



Phase Waiting Time			
	Phase	Traffic movement m waiting time	T_i^p
1	p^{WE}_{straight}	W_{straight}: 5s, E_{straight}: 3s	5
2	p^{WE}_{left}	W_{left}: −−, E_{left}: −−	0
3	p^{NS}_{straight}	N_{straight}: 0s, S_{straight}: 0s	0
4	p^{NS}_{left}	N_{left}: 13s, S_{left}: 22s	22

Fig. 4: Illustration of phase waiting time calculation.

in intersection I_i. It is defined as the margin of the positive extra waiting time of the real phase waiting time compared to the expected waiting time under round-robin scheduling:

$$F_i = \max\Big(\underbrace{\max_p \mathcal{T}_i^p - E_{wait}}_{\text{Extra waiting time}}, 0\Big), \tag{3}$$
$$\text{where } \mathcal{T}_i = [T_i^{p^{WE}_{\text{straight}}}, T_i^{p^{WE}_{left}}, T_i^{p^{NS}_{\text{straight}}}, T_i^{p^{NS}_{left}}],$$

where vector $\mathcal{T}_i$ contains all phase waiting time in intersection I_i. Scalar E_{wait} denotes the expected waiting time for traffic signal phase scheduling, which is used to distinguish between reasonable (or cyclic) and unfair scheduling. Given the phase duration Δt, E_{wait} is set to $(|P|-1)\times\Delta t$, i.e., the waiting time under a fixed time, cyclic scheduling (which is absolutely fair). F_i measures the **extra** waiting time of the most waited phase over the expected waiting time if the extra time is non-negative.

Overall, we propose phase waiting time and fairness based on extra waiting time for traffic signal control. **Definition** 7 provides a quantitative measure for comparing the phase scheduling fairness, encouraging to prioritize scheduling the long-waiting phases. FELight incorporates F_i when deciding on the traffic signal phase to trade off

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

scheduling fairness and transport performance. Besides, FELight exhibits flexibility, allowing it to be adaptable to various intersection types by adjusting the dimensions of state, action, and fairness metric F_i. In particular, different intersection types primarily influence the number of traffic signal phases and lanes, which directly impact the dimensions of state, action, and fairness metric F_i. Therefore, only these dimensions need to be modified accordingly to accommodate the new intersection types.

4.2 RL Model for Traffic Signal Control

To adapt to dynamic traffic, we model traffic signal control in the multi-intersection scenario as a multi-agent reinforcement learning problem to optimize the throughput by perceiving traffic conditions and autonomously controlling the traffic signal phase. Each traffic intersection is controlled by an RL agent trained individually. In addition, we integrate fairness concepts to alleviate unfair scheduling. For an intersection I_i, we introduce the definitions of state, action, and reward as follows:

State/Observation. In a multi-intersection scenario, the local observation for each agent consists of the current phase p, the number of waiting vehicles on incoming lanes connected to the intersection I_i, the fairness F_i based on extra waiting time (**Definition** 7), and the neighbor intersections traffic information. Specifically, the number of vehicles is used to measure the traffic pressure for each direction. Fairness based on extra waiting time indicates the extra scheduling delays. Additionally, neighbor intersection traffic information is employed to achieve traffic signal coordination among multiple intersections.

Action. In traffic signal control, there are two types of action, i.e., switch phases in sequence [40] or select the phase from a set of candidate phases [3], [13]. In our work, assuming that each action lasts for the same duration Δt, we define each agent's action as selecting a phase for the next time interval $[t, t+\Delta t]$. We employ four phases (Figure 2b) as candidate actions, widely used in existing works. In action set $\mathcal{A}$, action a is represented by an eight-dimensional vector indicating permissible traffic movements.

Reward. Each agent i receives an immediate reward r_t from the environment. In this study, the reward function is defined as a function of the queue length and fairness F_i based on extra waiting time:

$$r_t = -\underbrace{\sum_{l \in L_i^{in}} q_t(l)}_{performance} - \omega \times \underbrace{F_i^t}_{fairness}, \tag{4}$$

where $q_t(l)$ denotes the queue length of approaching lane l. F_i^t is the fairness (**Definition** 7) at time t, used to measure the scheduling fairness. Additionally, ω is a hyperparameter to balance the model performance and scheduling fairness.

During the training, DQN [41] is utilized to make decisions, and it consists of Q-network Q_θ and target Q-network $Q_{\bar{\theta}}$, parameterized with θ and $\bar{\theta}$, respectively. The objective function is defined as the mean squared error between the estimated Q-value $Q(s_t, a_t)$ and target Q-value y_t, which is formulated as:

$$\mathcal{L}_{RL} = \mathbb{E}\big(Q_\theta(s_t, a_t) - y_t\big)^2, \tag{5a}$$

$$y_t = r_t + \gamma \max_a Q_{\bar{\theta}}(s_{t+1}, a), \tag{5b}$$

where γ is a discount factor of cumulative reward to balance the significance of immediate versus long-term reward. Replay buffer $\mathcal{D}$ is maintained to stabilize the training process.

4.3 Counterfactual Data Augmentation

To enhance sample efficiency, we propose a novel counterfactual data augmentation that uses counterfactual learning to infer the interaction outcomes, which enriches the interaction data diversity and facilitates sample efficiency. Specifically, it incorporates a structural causal model (SCM) [42] to capture the causal mechanism. Given a set of observed variables $\boldsymbol{Y} = \{Y_1, \cdots, Y_n\}$, it is defined as:

$$Y_i = f_i(X_i, U_i), \tag{6}$$

where X_i denotes the parent variable of Y_i and U_i is the noises due to ignored factors. The function f_i is the causal mechanism that captures the relationship between Y_i, X_i, and U_i. Typically, an agent takes action a in a given state s_t, and observes the resulting environmental feedback in next state s_{t+1}. It is natural to wonder what would happen if the agent took other actions. Based on the SCM, we infer the counterfactual outcomes, i.e., next state $\tilde{s}_{t+1}$, by simulating the effects of a random action $\tilde{a}_t$ in the same state s_t, resulting in the predicted next state $\tilde{s}_{t+1}$. In this way, the agent obtains new data without interacting with the environment.

Assume that the state at $t+1$ satisfies a structural causal model, which is formulated as follows:

$$S_{t+1} := f(S_t, A_t, U_t), \tag{7}$$

where $f(\cdot)$ denotes the causal mechanism function, S_t and A_t are the state and action at time t. U_t denotes the noise term serving as an unobserved factor and is independent of S_t and A_t. Considering that the causal mechanism is always unknown in real-world tasks, FELight deploys General Adversarial Network (GAN) [43] to approximate the causal function $f(\cdot)$ in traffic signal control. Its core idea is to train the discriminator and generator alternately to generate "fake" samples that can fool the discriminator. The generator network is used to estimate the causal function, predicting the next state $\tilde{S}_{t+1}$ conditioned on the tuple $< S_t, A_t, U_t >$. During the training period, the discriminator and generator networks are trained alternately, and its objective function is defined as follows:

$$\begin{aligned} \mathcal{L}_{GAN} &= \min_G \max_D V(D, G) \\ &= \mathbb{E}_{s_{t+1} \sim \rho_{data}(s_{t+1})}[\log D(s_{t+1})] \\ &+ \mathbb{E}_{h_t \sim \rho_{h_t}(h_t)}[\log(1 - D(G(h_t)))], \end{aligned} \tag{8}$$

where h_t denotes the interaction tuple of state s_t, action a_t and noise u_t. The function $V(\cdot)$ denotes the degree of difference between the generated and original data. $G(\cdot)$ is the generator network to approximate the causal mechanism function $f(\cdot)$. The discriminator $D(\cdot)$ is used to distinguish the real interaction data and generated data. ρ_{data} and ρ_{h_t} denote the distribution of next state and interaction tuple. During training, we first sample the interaction data and then train the generator and discriminator, which are updated by using stochastic gradient descent to the objective function Equation (8).

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

Based on the estimated causal function f, counterfactual thinking is adopted to infer what would happen if an agent had taken another action. Given a piece of interaction data $< s_t, a_t, s_{t+1} >$, we want to know what next state $\tilde{s}_{t+1}$ is, if action a' is taken instead of the original a. The alternative action a' is randomly sampled from the action space and fed into the generator network G together with state s_t and noise term u_t, producing the counterfactual outcome $\tilde{s}_{t+1}$. Besides, the reward r_t is calculated based on the state information. In this way, the counterfactual-augmented dataset $\tilde{D}$ enriches the diversity of the interaction data and facilitates the sample efficiency of FELight. This counterfactual data augmentation is shown in Figure 3b.

4.4 Self-Supervised State Representation

The performance of reinforcement learning is closely tied to the quality of the state representation vector, particularly in complex tasks. In the context of TSC with multiple intersections, the state is intricate. The agents need to not only consider the locally observed state but also incorporate the state of neighbor intersections to collaborate control. It poses a significant challenge, as the state is inherently high-dimensional and interdependent, making it difficult to represent and utilize it for effective decision-making.

To efficiently utilize state information in decision-making period, this work separates a state encoder network from the Q-network and employs a self-supervised learning auxiliary task for the state encoder module to improve its representation quality (Figure 3c). The state encoder network employs the multi-head attention technique to enhance the cooperation among multiple intersections, which is formulated as follows:

$$e_{ij} = sim\big(embed(s_t^i), embed(s_t^j)\big), \tag{9a}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j' \in \eta_i} \exp(e_{ij'})}, \tag{9b}$$

$$enc_o = ReLU\big(W_q \cdot (\frac{1}{H} \sum_{h=1}^{H} \sum_{j \in \eta_i} \alpha_{ij} (h_j W_c)) + b_q\big), \tag{9c}$$

where $embed(\cdot)$ and $sim(\cdot)$ denote the embedding function and similarity function. $embed(\cdot)$ is implemented by multi-layer perceptron with 32-32 neurons. H denotes the number of attention heads. η_i represents the set of neighbors (the top 4, ranked by distance) of intersection I_i. The attention weight α_{ij} is used to calculate the representation vector enc_o. W_q, W_c and b_q are learnable parameters.

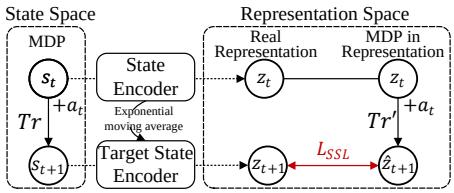


Fig. 5: Illustration of self-supervised learning loss.

The self-supervised information is derived from the transition relationship between sequential moments of the state. A good state encoder should well preserve the sequential dynamics of reinforcement learning, i.e., its Markov decision process (MDP) trajectory, with high fidelity, denoted as Tr: the predicted next state embedding $\hat{z}_{t+1}$ and the original next state embedding z_{t+1} should not be too far apart in the latent state representation space (Figure 5). Specifically, the proposed module comprises three networks: state encoder network, target state encoder network, and transition network (Figure 3c). The state encoder network, denoted as enc_o, maps the state s_t to the state embedding vector $z_t = enc_o(s_t)$, serving as the input of the Q-network. The transition network (multi-layer perceptron with 64-32 neurons), denoted as Tr', takes the state embedding vector z_t and action a_t as inputs and directly predicts the representation of the next state $\hat{z}_{t+1} = Tr'(z_t, a_t)$ for calculating the self-supervised loss. The target state encoder network,

Algorithm 1: Fairness-aware and Efficient Light (FELight)

Input: The number of the intersections N.
Output: Parameters for Q-network and state encoder.

Initialize network parameters $\theta_i, \overline{\theta}_i, \phi_o, \phi_m$ for each agent i, which controls the according intersection.
for $epoch = 1, 2, \cdots, max_epoch$ **do**
  Get the initial state s_0. ▷ *Interaction*
  for $t = 0, \Delta t, \cdots, T$ **do**
    Randomly select an action with probability ϵ otherwise select $a_t = \arg\max_a Q(s_t, a_t; \theta)$.
    Execute action a_t, receive the next state s'.
    for $t_{cur} = 0, 1, \cdots, \Delta t - 1$ **do**
      for each phase p **do**
        if $p = 0$ and $\mathcal{V}_{m_in}^{t_{cur}} \neq \emptyset$ **then**
          $T_i^p \leftarrow T_i^p + 1$.
        else
          $T_i^p \leftarrow 0$.
    Calculate the fairness metric F_i^t and reward r_t.
    Store transitions $< s_t, a_t, r_t, s_{t+1} >$ into replay buffer D, and set $s_t \leftarrow s_{t+1}$.
  ▷ *Data Augmentation*
  Sample the interact data from replay buffer D.
  Train the causal function GAN by Equation (8).
  Randomly sample alternative action $\tilde{a}$, generate the counterfactual interaction data $< s, \tilde{a}, \tilde{r}, \tilde{s} >$, and update it into augmented dataset $\tilde{D}$.
  ▷ *Training*
  Sample the interaction data $< s_t, a_t, r_t, s_{t+1} >$ from the counterfactual augmented datatset $\tilde{D}$.
  while *training for each intersection* **do**
    Calculate the current state embedding: $z_t \leftarrow enc_o(s)$.
    Calculate the target state embedding: $z_{t+1} \leftarrow enc_m(s_{t+1})$.
    Calculate the prediction of target state embedding: $\hat{z}_{t+1} \leftarrow Tr'(z_t, a_t)$.
    Calculate the target value y by Equation (5b).
    Update parameter θ by minimizing loss function: $\mathcal{L} \leftarrow \mathcal{L}_{RL} + \lambda \cdot \mathcal{L}_{SSL}$.
    Update target Q-network parameters: $\overline{\theta}_i \leftarrow \tau\overline{\theta}_i + (1-\tau)\theta_i$.
    Update target state encoder parameters: $\phi_m \leftarrow \tau\phi_m + (1-\tau)\phi_o$.
return The parameters θ_i and ϕ_o.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

denoted as enc_m, takes the next state s_{t+1} as input and then outputs the embedding $z_{t+1} = enc_m(s_{t+1})$. Finally, the estimated embedding $\hat{z}_{t+1}$ and the real embedding z_{t+1} are used in the self-supervised loss function, defined as:

$$\mathcal{L}_{SSL} = -\frac{1}{B}\left(\frac{z_{t+1}}{\|z_{t+1}\|_2}\right)^T\left(\frac{\hat{z}_{t+1}}{\|\hat{z}_{t+1}\|_2}\right), \tag{10}$$

where B is the batch size. Additionally, the gradient is halted in the target state encoder branch, and its parameters are updated as an exponential moving average of the parameters of the state encoder network:

$$\phi_m \leftarrow \tau\phi_m + (1-\tau)\phi_o, \tag{11}$$

where ϕ_o denotes the parameters of state encoder and ϕ_m is the parameters of target state encoder.

The Final Loss: The objective function of Equation (10) is trained jointly with the reinforcement learning loss function, Equation (5), using stochastic gradient descent with Adam. The joint loss function is defined as follows:

$$\mathcal{L} \leftarrow \mathcal{L}_{RL} + \lambda \cdot \mathcal{L}_{SSL}, \tag{12}$$

where λ is a hyperparameter. The self-supervised state representation is jointly trained with the reinforcement learning agents, managing to improve the state representations quality that enhances the decision-making process and ultimately gains a sample-efficient model.

4.5 Model Learning Process

The overall training process and time complexity analysis are as follows:

Overall training process: The proposed FELight model is trained with the Equation (12) as the final loss function and optimized via the gradient descent method. The pseudo-code for the overall training process of FELight is presented in Algorithm 1. Specifically, Lines 4-14 conduct the interaction between environment and agents. Lines 7-12 compute the fairness metric for each time step. Lines 16-18 make the data augmentation through GAN. Lines 21-30 compute the overall loss function of reinforcement learning and self-supervised representation learning, updating the model parameters, i.e., $\theta_i, \bar{\theta}_i, \phi_o$ and ϕ_m.

Time complexity analysis: We perform the time complexity analysis on FELight. Let d_f and d_a denote the dimensions of state and action, and $d_{fa} = d_f + d_a$; d is the dimension of hidden embedding. For all K intersections, 1) state encoder contains input embedding (32-neurons) and a multi-head attention network. The time complexities are the sum of $\mathcal{O}(d_f d + d^2)$ and $\mathcal{O}(dK + K + Kd)$, respectively. 2) Time complexity of Q-network (64-32 neurons) is $\mathcal{O}(2d^2 + 2d^2 + dd_a)$. 3) Transition network contains a_t-embedding layer (32-neurons), z_t-embedding layer (32-neurons) and $\hat{z}_{t+1}$-prediction layer (64-neurons). Its time complexity is the sum of $\mathcal{O}(d_a d + d^2)$, $\mathcal{O}(d^2 + 2d^2)$, and $\mathcal{O}(3d^2 + 2d^2)$. 4) Causal mechanism includes the generator (64-128 neurons) and the discriminator (64-128 neurons). Their time complexities are $\mathcal{O}(2d_{fa}d + 8d^2 + 4dd_f)$ and $\mathcal{O}(2d_f d + 8d^2 + 4d)$, respectively. Therefore, the overall time complexity of FELight is the sum of aforementioned time complexities, approximated as $\mathcal{O}(Kd + d^2)$, considering that d is greater than d_f, d_a and $d_f a$.

In summary, the advantages of FELight are three-fold: 1) It is the first work to balance scheduling fairness and performance in traffic signal control, mitigating unfair phenomena without compromising performance. 2) FELight approximates the causal mechanism behind state transitions and enriches the diversity of interaction data by counterfactual augmentation. 3) Our model is sample efficient due to the auxiliary self-supervised state representation task to mine more information from the original state.

5 Experiment

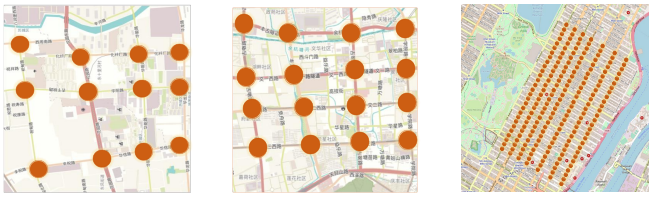
To empirically evaluate our approach, extensive experiments are conducted on seven real-world datasets and compared with state-of-the-art methods. Specifically, three research questions are investigated as follows:

- **RQ1**: Whether the proposed model is effective in traffic signal control tasks in an efficient manner?
- **RQ2**: Could FELight alleviate the scheduling unfairness in traffic signal control tasks?
- **RQ3**: What influence do the various hyperparameter settings have on the proposed model?

5.1 Experiment Setup

We follow existing studies [12], [13] and conduct experiments on CityFlow [20] traffic simulator, which simulates the kind of data that would typically be collected by devices at real-world intersections for TSC, such as cameras and radar, to provide vehicle position, speed, etc. The agent obtains states from this simulator and sets the traffic signal phase according to its policy. Each green light signal is followed by three-second yellow light signal and two-second all red time to prepare the transition. Following most existing methods, the minimum action duration Δt is 10 seconds. The hyperparameter ω used in the fairness-aware reward function is set as 0.4. The trade-off parameter λ used in the final loss function is set as 0.01.

Datasets. The experiments are conducted on seven real-world traffic flow datasets collected in three cities with different network scales: Jinan (3×4), Hangzhou (4×4), and New York (28×7) [13]. The visualization of different network scales is shown in Figure 6. Each intersection has four incoming roads and four outgoing roads, and each road has three lanes: turn left, go straight, and turn right lane.



(a) Jinan (b) Hangzhou (c) New York

Fig. 6: Illustration of network scales for different traffic flow datasets. The orange dots mark the traffic signal lights controlled by RL-agent.

Baselines. To assess the effectiveness of FELight, we compare it with ten existing approaches, including three traditional methods, as well as seven RL-based methods.

Traditional Methods:

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

TABLE 3: Performance comparison of different traffic signal control methods on real-world traffic datasets. The evaluation metrics are the average travel time (the lower, the better) and its standard deviation. The best results are highlighted in boldface. "Improvement" under FELight shows the improvement over the best baseline.

	Method	Jinan			Hangzhou		New York	
		Dataset-1	Dataset-2	Dataset-3	Dataset-4	Dataset-5	Dataset-6	Dataset-7
Traditional	FixedTime [2]	$442.98_{\pm 0.00}$	$366.55_{\pm 0.00}$	$389.53_{\pm 0.00}$	$475.00_{\pm 0.00}$	$396.02_{\pm 0.00}$	$1444.80_{\pm 0.00}$	$1697.79_{\pm 0.00}$
	MaxPressure [3]	$355.11_{\pm 0.00}$	$337.15_{\pm 0.00}$	$331.66_{\pm 0.00}$	$416.82_{\pm 0.00}$	$408.23_{\pm 0.00}$	$1215.37_{\pm 0.00}$	$1568.20_{\pm 0.00}$
	SOTL [9]	$440.27_{\pm 0.00}$	$403.65_{\pm 0.00}$	$406.72_{\pm 0.00}$	$487.61_{\pm 0.00}$	$435.05_{\pm 0.00}$	$1482.68_{\pm 0.00}$	$1790.84_{\pm 0.00}$
RL-based	MPLight [12]	$318.16_{\pm 6.66}$	$294.38_{\pm 5.51}$	$285.34_{\pm 2.04}$	$318.18_{\pm 5.61}$	$364.25_{\pm 10.90}$	$1864.32_{\pm 48.24}$	$1922.49_{\pm 41.49}$
	AttendLight [26]	$310.68_{\pm 15.18}$	$255.70_{\pm 1.78}$	$260.85_{\pm 5.26}$	$310.79_{\pm 65.6}$	$377.54_{\pm 19.13}$	$1743.88_{\pm 50.94}$	$1921.69_{\pm 37.10}$
	CoLight [13]	$298.12_{\pm 14.14}$	$253.55_{\pm 1.47}$	$278.26_{\pm 7.85}$	$295.01_{\pm 4.07}$	$376.01_{\pm 9.25}$	$1086.28_{\pm 57.01}$	$1400.75_{\pm 34.29}$
	UniLight [44]	$339.78_{\pm 6.81}$	$318.47_{\pm 5.49}$	$323.88_{\pm 4.17}$	$333.13_{\pm 8.89}$	$412.13_{\pm 6.01}$	$1202.18_{\pm 82.45}$	$1589.39_{\pm 19.99}$
	3DQN-like [45]	$309.44_{\pm 28.63}$	$262.01_{\pm 9.40}$	$257.79_{\pm 6.33}$	$304.53_{\pm 10.59}$	$371.49_{\pm 16.01}$	$1098.16_{\pm 58.04}$	$1457.76_{\pm 28.91}$
	FIT-like [16]	$1033.85_{\pm 117}$	$431.69_{\pm 25.79}$	$574.60_{\pm 44.21}$	$627.36_{\pm 129}$	$634.54_{\pm 54.55}$	$1756.44_{\pm 33.61}$	$2047.35_{\pm 65.88}$
	DFC-like [18]	$500.55_{\pm 30.38}$	$347.42_{\pm 17.95}$	$610.79_{\pm 83.23}$	$301.25_{\pm 3.71}$	$510.45_{\pm 28.35}$	$1735.20_{\pm 35.87}$	$1915.91_{\pm 56.53}$
	TFC-like [18]	$581.79_{\pm 185.2}$	$381.58_{\pm 36.12}$	$483.49_{\pm 127.5}$	$354.26_{\pm 7.29}$	$456.70_{\pm 22.49}$	$1223.69_{\pm 18.35}$	$1651.25_{\pm 45.23}$
	FELight (**ours**)	$\mathbf{261.37_{\pm 1.46}}$	$\mathbf{233.97_{\pm 0.62}}$	$\mathbf{234.55_{\pm 0.51}}$	$\mathbf{272.92_{\pm 0.26}}$	$\mathbf{342.92_{\pm 9.42}}$	$\mathbf{871.93_{\pm 14.97}}$	$\mathbf{1273.55_{\pm 39.04}}$
	Improvement	12.33%	7.72%	9.45%	7.49%	5.86%	19.73%	9.08%

- **FixedTime** [2]. It is a traditional method, scheduling with a pre-defined periodic phase scheduling plan. It is widely used due to its simplicity.
- **MaxPressure** [3]. It schedules the phases based on pre-defined rules, favoring the phase with the highest traffic pressure to maximize traffic throughput.
- **SOTL** [9]. It triggers the phases when the number of queued vehicles exceeds a hand-tuned threshold.

RL-based Methods w/o Fairness:

- **MPLight** [12]. MPLight extends FRAP method [46] for large-scale traffic signal control, and incorporates pressure concept into the state and reward design.
- **AttendLight** [26]. It employs an attention mechanism in RL algorithm to improve the model's generalization across different intersection structures.
- **CoLight** [13]. CoLight integrates a graph attention network into multi-intersection traffic signal control to enable collaboration among multiple intersections.
- **UniLight** [44]. UniLight is proposed to coordinate the communication among multiple intersections and improve performance.
- **3DQN-like** [45]. It modifies CoLight and uses Double Dueling DQN (3DQN) as underlying algorithm.

RL-based Methods w/ Fairness: [1]

- **FIT-like** [16]. FIT-like draws inspiration from proportional fair scheduling, where the ratio of queue length to average throughput per lane is used to measure fairness.
- **DFC-like** [18]. DFC-like uses the partial waiting time so far to measure the fairness, and the reward is defined as the sum of all vehicles' waiting time.
- **TFC-like** [18]. TFC-like adopts the idea of weighted fairness, where different road traffic throughputs are divided by their arrival-related weights. Ratio deviation is used as a fairness metric (the lower, the fairer).

Accordingly, we compare FELight with RL-based methods w/ fairness in terms of reward design to highlight its advantages, and the details are presented in Table 1. Notably, FELight introduces a novel fairness concept, i.e., fairness based on extra waiting time F_i, **only** penalizing cases characterized by higher upper bounds on latency, ensuring a fair traffic signal schedule of the entire system while minimizing the impact on overall performance.

Evaluation Metrics. We adopt the average travel time (ATT) and its standard deviation as performance metrics, consistent with established practices in previous studies [13], [22]. The average travel time refers to the duration in seconds that vehicles spend on the incoming lanes. Given that all methods use the same number of interaction rounds during training, sample efficiency is assessed based on performance. In other words, the better performance of the model means that it is more sample-efficient. Additionally, we incorporate the hindsight waiting time for vehicles on incoming lanes as a metric of traffic signal scheduling fairness.

5.2 Results and Analysis

5.2.1 Overall Performance (*RQ1*)

To verify the effectiveness of FELight, we conduct the experiments on real traffic datasets, which include data from three cities with different traffic patterns. We investigate the performance of FELight w.r.t the average travel time and compare it with state-of-the-art traffic signal control methods. Table 3 reports the performance of different methods, with lower values of this metric indicating better model performance. We can find that:

1) FELight achieves consistent performance improvements over state-of-the-art methods across diverse traffic road networks and traffic patterns, where the average performance improvement is $+10.34\%$ over the best baseline. It is attributed to the benefits of scheduling the phase in a fair manner, avoiding the extremely long waits for individual vehicles found in other RL-based methods, such as MPLight, AttendLight, CoLight, UniLight, and 3DQN-like. Besides, self-supervised state representation and counterfactual data augmentation are conducive to improving the sample efficiency with diversity-limited traffic datasets, i.e., achieving higher performance with the same amount of training data.

1. To apply to multi-intersection tasks, we use our basic model as the backbone and integrate these methods into it, referred to as **X-like**, adopting the state and reward calculations mentioned in original paper.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

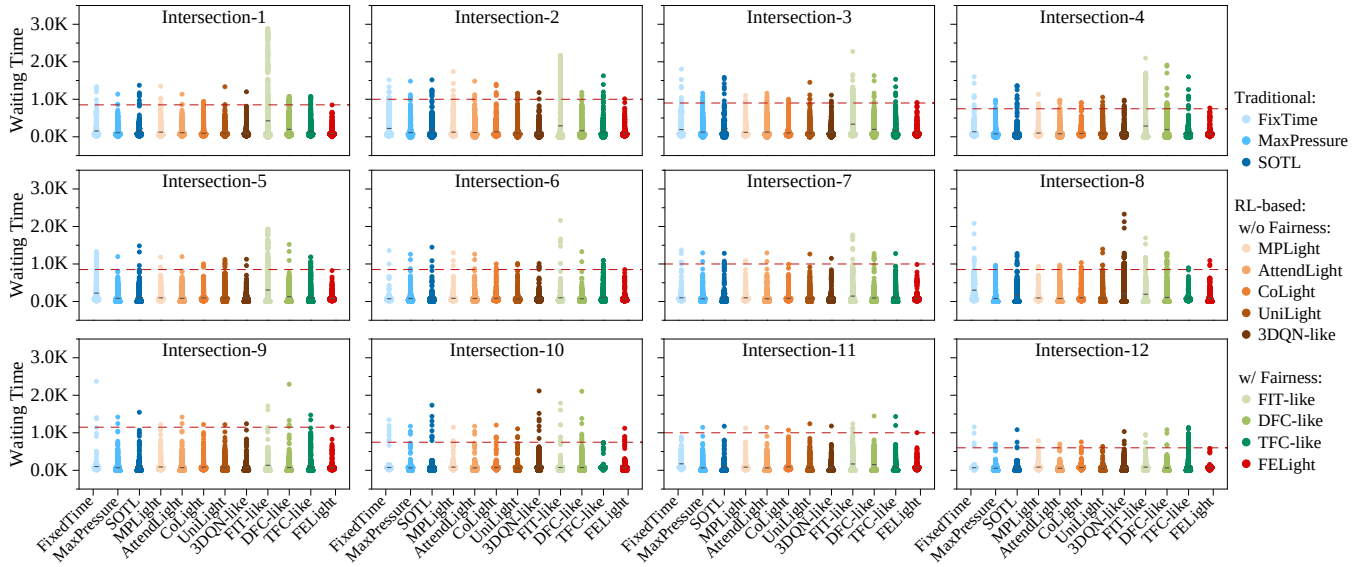


Fig. 7: Visualization of vehicle waiting time in seconds at each intersection in Jinan, and its value is measured in hindsight.

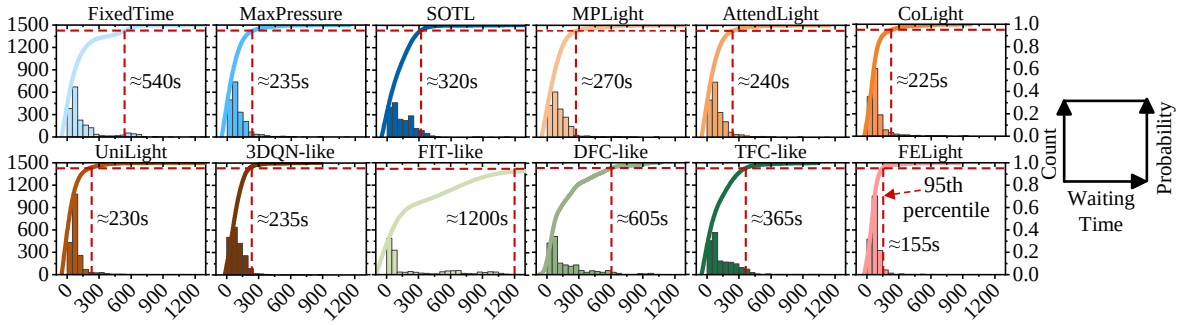


Fig. 8: Distribution histograms of vehicle waiting time and cumulative distribution function (CDF) curves. The X-axis denotes the waiting time in seconds, the left Y-axis denotes the count within the corresponding waiting time interval, and the right Y-axis is the probability. The red dotted line indicates the 95th percentile of the vehicle waiting time.

2) The performance gap between FELight and other w/ fairness methods complies with the inherent weaknesses of existing fairness methods. Namely, their fairness mechanisms are preferred to be invoked frequently, which is harmful to performance due to it is incapable of balancing fairness with model performance. In contrast, FELight penalizes the unfair cases with a higher value on scheduling delay. In this way, FELight not only provides fair phase scheduling but also reduces the impact to model performance.

3) The proposed FELight outperforms traditional traffic signal control methods. This superiority stems from traditional methods' limitations, which rely heavily on predefined rules for phase scheduling and lack the ability to dynamically adapt to changing traffic conditions. Consequently, these traditional methods result in longer vehicle travel times and diminished model performance.

5.2.2 *Fairness Comparison (**RQ2**)*

This subsection investigates whether FELight is fairer than other traffic signal control methods. Figure 7 presents the vehicle waiting time at each intersection of FELight as well as other methods in Jinan dataset. The red dashed line indicates the maximum vehicle waiting time for FELight. Clearly, FELight achieves the smallest value, i.e., it effectively reduces the upper bound of vehicle waiting times and prevents the occurrence of excessive waiting, which indicates improved fairness. We also find that other methods w/ fairness have higher waiting time, which is likely because such methods frequently trigger the fairness mechanism. These discrepancies in vehicle waiting time align with our earlier experimental analysis that methods w/o fairness result in a longer vehicle waiting time due to unfair scheduling. In contrast, methods incorporating fairness mechanism experience may increased waiting times due to traffic jam caused by an excessive focus on fairness.

Furthermore, to comprehensively analyze the vehicle waiting time, we plotted the distribution histogram and cumulative distribution function curve, as shown in Figure 8. The red dashed line represents the 95th percentile of vehicle waiting time, which serves as a threshold for comparison. The 95th percentile is an important metric when analyzing, and it indicates that 95% of the vehicles have waiting time below this threshold. We aim to control the traffic signal strategy in a way that keeps the waiting time of the majority of vehicles relatively low. From Figure 8, we can better understand the overall characteristics of vehicle waiting time. It is evident that under the control of FELight, the overall vehicle waiting time in the traffic system is shorter. In conclusion, the control of FELight achieves a good balance between fairness and performance. It means that FELight effectively reduces vehicle waiting time and

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

provides a fairer traffic signal control.

5.2.3 Analysis

To illustrate the superior performance of FELight over existing methods, we give an in-depth analysis as follows. First, average travel time (ATT) represents the arithmetic mean of the travel times for all vehicles from the starting point to the destination, and it is defined in the following form:

$$ATT = \frac{1}{N}\sum_{n}^{N} w_n + T_n^{travel} = \frac{w_1 + \cdots + w_N + C}{N}, \quad (13)$$

where N is the number of vehicles and n denotes the vehicle index. ATT is made up of waiting time w_n and travel time T_n^{travel}. The waiting time w_n is primarily influenced by traffic signal light scheduling and road conditions. Travel time is expressed as the ratio of distance to speed $T_n^{travel} = \frac{distance}{speed}$. Since the vehicle trajectories in the dataset are fixed (i.e., distance remains constant) and variations in driving speed can be ignored, the actual travel time T_n^{travel} can be approximated as a constant c_n. $C = \sum_{n=1}^{N} c_n$ is the sum of all vehicles' travel time. For clarity, we simplify the problem based on several assumptions:

Assumption 1. *Policy replacement for w/o and w/ fairness is practiced in one execution interval for a given intersection, while the remaining time adopts the policy w/o fairness. For simplicity, it is assumed that the intersection is equipped by two traffic signal phases, controlling a high-load lane (the number of vehicles is h) and a low-load lane (the number of vehicles is l, and $h > l$).*

Assumption 2. *Except for the vehicles on high-load or low-load lane mentioned in **Assumption** 1, other vehicles' waiting time is unaffected or its time difference is negligible.*

Assumption 3. *Vehicles delayed in high-load lane are released in the next Δt. According to **Assumption** 1, the policy replacement only affects one decision interval. Under the policy w/ fairness, the vehicles on high-load lane are delayed, which will be permitted to pass in the next interval.*

Based on assumptions, the ATT is updated as follows:

$$ATT = \frac{w_1 + \cdots + \sum_{n_h \in \mathcal{N}_h} w_{n_h} + \sum_{n_l \in \mathcal{N}_l} w_{n_l} + \cdots + w_N + C}{N}, \quad (14)$$

where $\mathcal{N}_h$ and $\mathcal{N}_l$ denote the index sets of vehicles on high-load and low-load lanes ($|\mathcal{N}_h| = h, |\mathcal{N}_l| = l$). Secondly, ATT_{noF} and ATT_F are presented, corresponding to the control policy w/o and w/ fairness:

$$ATT_{noF} = \frac{w_1 + \cdots + \sum_{n_h \in \mathcal{N}_h} w_{n_h} + \sum_{n_l \in \mathcal{N}_l} w_{n_l} + \cdots + w_N + C}{N},$$
$$ATT_F = \frac{w_1 + \cdots + \sum_{n_h \in \mathcal{N}_h} w'_{n_h} + \sum_{n_l \in \mathcal{N}_l} w'_{n_l} + \cdots + w_N + C}{N}, \quad (15)$$

where w and w' (the subscripts are omitted for brevity) denote the waiting time under w/o and w/ fairness policies.

Finally, the time difference δ (Figure 9) in ATT under w/o and w/ fairness policies is defined as Equation (16). Based on **Assumption** 2, the change of other vehicles' waiting time is negligible, then δ is simplified as follows:

$$\begin{aligned}\delta &= ATT_{noF} - ATT_F \\ &= \left(\sum_{n_h \in \mathcal{N}_h} (w_{n_h} - w'_{n_h}) + \sum_{n_l \in \mathcal{N}_l} (w_{n_l} - w'_{n_l})\right)/N \quad (16) \\ &\approx \left(h \times (-\Delta t) + l \times \kappa \Delta t\right)/N,\end{aligned}$$

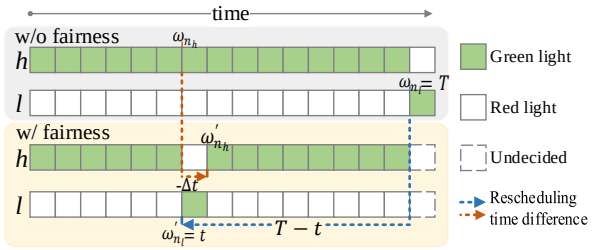


Fig. 9: Visualization of phase scheduling time difference under control policies w/o and w/ fairness.

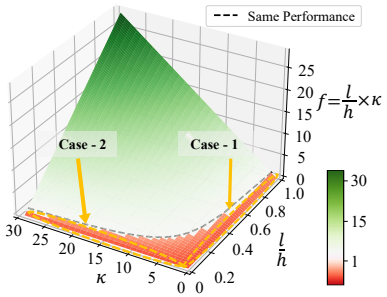


Fig. 10: Visualization of the solution space, where all the white to green regions mean positive effect of introducing fairness and the orange regions mean negative effect. The yellow circle denotes the case of existing RL-based methods insensitive (Case - 1) or oversensitive (Case - 2) to fairness.

where $w_{n_h} - w'_{n_h}$ is equal to $-\Delta t$ (**Assumption** 3). $T = k\Delta t$ denotes the moment when the low-load phase is scheduled under the non-fairness policy; $t = k'\Delta t$ denotes the moment when the low-load phase is scheduled under the fairness policy. Then, we get $w_{n_l} - w'_{n_l} = T - t = \kappa \Delta t$ (Figure 9). The difference between k and k' is denoted as κ. Based on Equation (16), we solve for $\delta > 0$, in which case w/ fairness method outperforms the method w/o fairness. The objective function after simplification is as follows:

$$\delta > 0 \Rightarrow 1 < \frac{l \times \kappa}{h}, \text{s.t.} \begin{cases} 0 < \kappa < max_round \\ 1 \le l < h \le max_vehicle \end{cases}, \quad (17)$$

where max_round is the maximum interaction times, and $max_vehicle$ denotes the maximum number of vehicles that can be loaded on the lane.

To display the solution intuitively, we make a visualization of the objective function as shown in Figure 10. The X-axis represents κ, and a larger value of κ indicates a stronger emphasis on fairness while keeping T constant. The Y-axis denotes traffic flow ratio between low-load and high-load lanes, and a smaller ratio l/h indicates an uneven traffic flow distribution in the traffic network, where fairness is needed to be appropriately considered. The Z-axis is the product of κ and l/h, i.e., $f = \kappa \times l/h$. When $f > 1$ (Equation (17)), it indicates better performance of fairness-aware methods.

By intuitively observing the patterns in Figure 10, we can find that methods w/ fairness are more likely to achieve better performance, as shown in the white to green areas. Among them, the orange region represents situations where the performance is not ideal when fairness is considered, which can be divided into two cases:

Case - 1: Insensitive The right yellow dashed circle represents cases with a smaller κ, indicating a lower emphasis

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

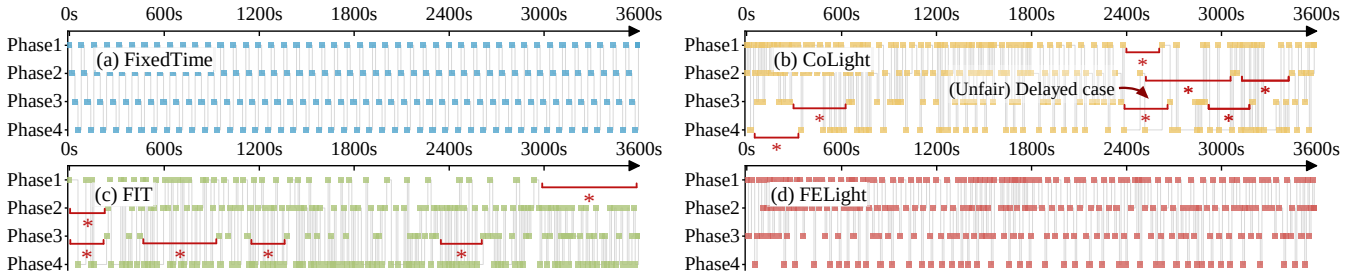


Fig. 11: Case study of traffic signal phase scheduling. The colored sections indicate scheduled time, and the others denote unscheduled. Red bracket ($\overset{*}{\sqcap}$) marks the instances of delayed scheduling cases (the larger the span of $\overset{*}{\sqcap}$, the more unfair).

TABLE 4: Results of ablation study."F_i", "CF", and "SSL" denote fairness-aware scheduling, counterfactual data augmentation, and self-supervised state representation. The evaluation metrics are average travel time and its standard deviation.

Method	Jinan			Hangzhou		New York	
	Dataset-1	Dataset-2	Dataset-3	Dataset-1	Dataset-2	Dataset-1	Dataset-2
FELight (**ours**)	$\mathbf{261.37_{\pm 1.46}}$	$\mathbf{233.97_{\pm 0.62}}$	$\mathbf{234.55_{\pm 0.51}}$	$\mathbf{272.92_{\pm 0.26}}$	$342.92_{\pm 9.42}$	$\mathbf{871.93_{\pm 14.97}}$	$\mathbf{1273.55_{\pm 39.04}}$
FELight w/o F_i	$267.65_{\pm 3.63}$	$236.02_{\pm 1.36}$	$235.34_{\pm 0.55}$	$284.13_{\pm 6.95}$	$345.64_{\pm 4.08}$	$1095.83_{\pm 36.74}$	$1433.30_{\pm 41.03}$
FELight w/o CF	$273.72_{\pm 1.66}$	$241.54_{\pm 1.29}$	$242.49_{\pm 1.73}$	$278.28_{\pm 1.91}$	$347.67_{\pm 9.88}$	$1000.76_{\pm 41.16}$	$1319.62_{\pm 41.48}$
FELight w/o SSL	$263.27_{\pm 3.11}$	$234.11_{\pm 0.50}$	$233.17_{\pm 0.08}$	$273.36_{\pm 0.80}$	$\mathbf{341.87_{\pm 3.75}}$	$938.53_{\pm 81.77}$	$1292.29_{\pm 105.1}$
FELight w/o F_i-CF	$288.49_{\pm 2.28}$	$260.82_{\pm 3.06}$	$259.41_{\pm 1.94}$	$300.01_{\pm 2.06}$	$362.65_{\pm 2.89}$	$1043.40_{\pm 43.67}$	$1471.31_{\pm 33.38}$
FELight w/o F_i-SSL	$267.13_{\pm 5.65}$	$233.97_{\pm 0.37}$	$234.61_{\pm 0.14}$	$278.94_{\pm 3.98}$	$346.03_{\pm 3.43}$	$1072.42_{\pm 42.93}$	$1405.64_{\pm 26.81}$
FELight w/o CF-SSL	$273.52_{\pm 5.10}$	$240.42_{\pm 0.39}$	$241.79_{\pm 0.65}$	$280.82_{\pm 4.61}$	$343.62_{\pm 0.45}$	$1079.57_{\pm 38.42}$	$1361.80_{\pm 28.70}$

on fairness, which can be regarded as RL-based methods w/o fairness considerations, such as MPLight and CoLight.

Case - 2: Oversensitive The left yellow dashed circle represents cases where there is a significant difference in flow between low-load and high-load conditions. In the case of disparity in traffic flow distribution, activating fairness (to scheduling low-load lane) impairs the overall performance. The failure case is observed in existing w/ fairness approaches, like FIT, DFC, and TFC, which are oversensitive to fairness and frequently trigger the fairness mechanisms.

In summary, the design of a fairness mechanism with a threshold allows for balancing performance while providing fair scheduling, highlighting the advantage of FELight.

5.3 Case study

Figure 11 shows a case study, i.e., the scheduling patterns of traffic signal phase under the same traffic flow, comparing our model with FixedTime, CoLight, and FIT, which are more representative. In the example, we explore the issue of fairness in traffic signal phase scheduling. Specifically, FixedTime proposes to allocate equal time for each phase, ensuring absolute fairness in traffic signal phase scheduling (Figure 11a). CoLight, a method not considering fairness, prioritizes phases with higher traffic flow, which leads to significant scheduling delays, as marked by the red brackets in Figure 11b. FIT exhibits an excessive sensitivity to fairness. It prioritizes the scheduling of perceived unfair phases (i.e., Phase 3 with lower traffic flow), which in turn impairs the overall performance (Table 3). Unlike existing methods, our proposed FELight balances performance and scheduling fairness, providing relatively fair scheduling while ensuring model performance. The result indicates that the traffic signal phase scheduling pattern of FELight (Figure 11d) is more similar to that of FixedTime (Figure 11a), which demonstrates that FELight schedules the traffic signal phases in a relatively fair manner.

5.4 Ablation Study

To evaluate the contributions of all components in FELight, regarding fairness-aware scheduling (F_i), counterfactual data augmentation (CF), and self-supervised state representation (SSL) modules, we conduct ablation experiments on real-world datasets. Table 4 reports that FELight consistently outperforms other variations except in one case (where it is the second best) highlighting the positive impact of its components on traffic signal control.

Specifically, when any component is removed, the model's performance is noticeably compromised compared to the complete model. Among the three components, fairness-aware scheduling module has been identified as the most crucial, exhibiting the highest importance by contributing to a performance decline of approximately 6.68% when removed. Counterfactual data augmentation module ranks as the second most important component, causing a performance decrease of around 4.73% when omitted. Lastly, self-supervised state representation component, while having the least impact, still influences the performance with a decline of approximately 1.31% when excluded.

This observation suggests that all components are valuable and contribute synergistically to the overall effectiveness of FELight. It is likely because the fairness-aware scheduling module considers the long waiting time of low-load lanes in the scheduling process, allowing vehicles on these lanes to leave the traffic system earlier, thereby reducing waiting times and decreasing overall average travel time. Furthermore, data augmentation module increases the diversity of interaction data. Such diversity provides richer information during the training process, thereby enhancing the performance of the model. Additionally, self-supervised state representation module enables us to capture key state feature embeddings from the rich traffic information, leading to more efficient training of the Q-network. In all, the

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

analysis reveals that each component plays a crucial role in enhancing the performance and sample efficiency.

5.5 Parameter Sensitivity Analysis (RQ3)

In this section, we study the effect on the overall performance (i.e., the smaller the value of ATT, the better the performance) of our proposed FELight with two hyperparameters, namely the fairness weight parameter ω in the reward function and λ for the self-supervised loss weight.

First, we study the effect of hyperparameter ω for fairness weight in the reward function and take the values in the range $[0.2, 0.4, 0.6, 0.8, 1.0]$ for five datasets. The overall performance results are plotted in Figure 12a and Figure 12b. Specifically, if we set its value as 0.2, fairness is not adequately considered in the reward function. We can observe that the overall performance is poor in 4 out of 5 datasets, indicated by high ATT. The best results are obtained when setting $\omega = 0.4$ or $\omega = 0.6$ for most datasets. If setting $\omega > 0.6$, the value of ATT increases across most datasets, i.e., overall performance declines consistently, which shows that the fairness awareness needs to be moderately considered in the reward function and should not be set too large weights to make a balance between overall performance and decision fairness. Overall, the optimal value of ω is around 0.4 to 0.6, and we set $\omega = 0.4$ in the main experiments. Taking fairness into the reward function design is conducive to providing a well-performing and fair traffic signal control policy, alleviating the unfair phenomenon of long vehicle waiting, and improving overall performance.

Second, we study the effect of the trade-off parameter λ for the self-supervised module and take the values in the range $[0.01, 0.03, 0.05, 0.07, 0.09]$. Figure 12c and Figure 12d presents the overall performance results. Specifically, the best results are obtained when setting $\lambda = 0.01$. If setting $\lambda > 0.01$, the ATT increases with varying degrees, and the performance declines consistently. It demonstrates that the proposed self-supervised state representation module can improve the performance, but it is only an auxiliary loss in the final loss function and should not be given too large weights to restrict its role in the final loss function. Overall, the self-supervised state representation module benefits the performance of the proposed FELight in traffic signal control tasks. We set $\lambda = 0.01$ in the main experiments.

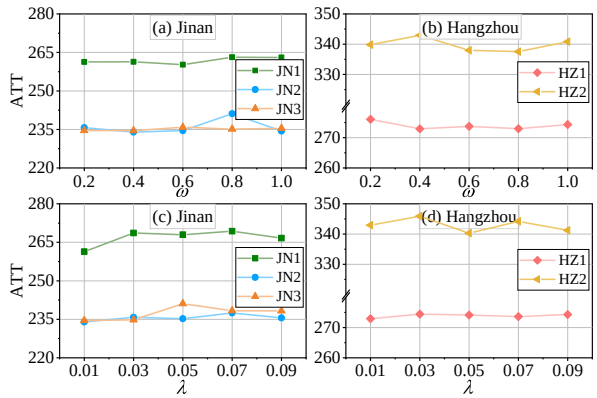


Fig. 12: Performance comparisons of average travel time with different hyperparameter settings.

6 Conclusion

In this paper, we propose FELight, a novel method for fair and sample-efficient traffic signal control. To ensure fairness, we introduce a fairness metric F_i, which is integrated into the state and reward functions to activate fairness mechanisms and penalize unfair scheduling scenarios. Theoretical comparison with other fairness works indicates why and when our fairness could bring advantages. Additionally, counterfactual data augmentation and self-supervised state representation are employed to enhance interaction data diversity and capture rich state features, respectively. These techniques contribute to the sample efficiency of FELight, enabling it to achieve superior model performance even with diversity-limited traffic data. Experimental results demonstrate that FELight effectively provides fair traffic signal control without compromising performance. In the future, FELight can be readily extended to more flexible control methods, such as variable time slot decisions, to address future challenges in transport efficiency.

Acknowledgments

This work is partially supported in part by the National Key R&D Program of China (2021ZD0112501 and 2021ZD0112502); the International Cooperation Project of Jilin Province (20220402009GH); the National Natural Science Foundation of China (61976102 and U19A2065).

References

[1] M. A. Fattah, S. R. Morshed *et al.*, "Insights into the socio-economic impacts of traffic congestion in the port and industrial areas of chittagong city, bangladesh," *Transportation Engineering*, 2022.
[2] L. Singh, S. Tripathi, and H. Arora, "Time optimization for traffic signal control using genetic algorithm," *International Journal of Recent Trends in Engineering*, vol. 2, no. 2, p. 4, 2009.
[3] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
[4] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *ACM SIGKDD*, 2018, pp. 2496–2505.
[5] H. Wei, C. Chen, G. Zheng *et al.*, "Presslight: Learning max pressure control to coordinate traffic signals in arterial network," in *ACM SIGKDD*, 2019, pp. 1290–1298.
[6] X. Guo, Z. Yu, P. Wang *et al.*, "Urban traffic light control via active multi-agent communication and supply-demand modeling," *IEEE TKDE*, vol. 35, no. 4, pp. 4346–4356, 2023.
[7] P. Lowrie, "SCATS, Sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic," 1990.
[8] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic engineering*. Pearson/Prentice Hall, 2004.
[9] S.-B. Cools, C. Gershenson, and B. D'Hooghe, "Self-organizing traffic lights: A realistic simulation," in *Advances in applied self-organizing systems*. Springer, 2013, pp. 45–55.
[10] Z. Chenguang, H. Xiaorong, and W. Gang, "Prg-light: A novel traffic light control framework with pressurebased-reinforcement learning and graph neural network," in *IJCAI*, 2021.
[11] F.-X. Devailly, D. Larocque, and L. Charlin, "Ig-rl: Inductive graph reinforcement learning for massive-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
[12] C. Chen, H. Wei, N. Xu *et al.*, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *AAAI*, 2020, pp. 3414–3421.
[13] H. Wei, N. Xu *et al.*, "Colight: Learning network-level cooperation for traffic signal control," in *CIKM*, 2019, pp. 1913–1922.
[14] X. Zang, H. Yao *et al.*, "Metalight: Value-based meta-reinforcement learning for traffic signal control," in *AAAI*, 2020, pp. 1153–1160.
[15] H. Jiang, Z. Li, L. BAI, R. Zhao *et al.*, "A general scenario-agnostic reinforcement learning for traffic signal control," 2023.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2024.3376745

[16] J. Chen, Z. Zhang, J. Feng, and K. Zhu, "Fit: Fairness-aware intelligent traffic signal control with deep reinforcement learning," in *IEEE HPCC*, 2021, pp. 846–852.

[17] Y. Ye, J. Ding, T. Wang *et al.*, "Fairlight: Fairness-aware autonomous traffic signal control with hierarchical action space," *IEEE TCAD*, pp. 1–1, 2022.

[18] M. Raeis and A. Leon-Garcia, "A deep reinforcement learning approach for fair traffic signal control," in *IEEE International Intelligent Transportation Systems Conference*, 2021, pp. 2512–2518.

[19] M. Behrisch, L. Bieker *et al.*, "SUMO–Simulation of Urban MObility: An overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, 2011.

[20] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou *et al.*, "Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *WWW*, 2019, pp. 3620–3624.

[21] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator vissim," *Fundamentals of traffic simulation*, pp. 63–93, 2010.

[22] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," *arXiv preprint arXiv:1904.08117*, 2019.

[23] Y. Xiong, G. Zheng, K. Xu, and Z. Li, "Learning traffic signal control from demonstrations," in *CIKM*, 2019, pp. 2289–2292.

[24] B. Xu, Y. Wang, Z. Wang *et al.*, "Hierarchically and cooperatively learning traffic signal control," in *AAAI*, 2021, pp. 669–677.

[25] L. Zhu, P. Peng, Z. Lu, and Y. Tian, "Metavim: Meta variationally intrinsic motivated reinforcement learning for decentralized traffic signal control," *IEEE TKDE*, pp. 1–18, 2023.

[26] A. Oroojlooy, M. Nazari, D. Hajinezhad, and J. Silva, "Attendlight: Universal attention-based reinforcement learning model for traffic signal control," *NeurIPS*, vol. 33, pp. 4079–4090, 2020.

[27] H. Zhang, C. Liu, W. Zhang, G. Zheng, and Y. Yu, "Generalight: Improving environment generalization of traffic signal control via meta reinforcement learning," in *CIKM*, 2020, pp. 1783–1792.

[28] C. Li, X. Ma, L. Xia *et al.*, "Fairness control of traffic light via deep reinforcement learning," in *IEEE CASE*, 2020, pp. 652–658.

[29] M. Oberst and D. Sontag, "Counterfactual off-policy evaluation with gumbel-max structural causal models," in *ICML*, 2019, pp. 4881–4890.

[30] C. Lu, B. Huang, K. Wang *et al.*, "Sample-efficient reinforcement learning via counterfactual-based data augmentation," *arXiv preprint arXiv:2012.09092*, 2020.

[31] A. Forney, J. Pearl, and E. Bareinboim, "Counterfactual data-fusion for online reinforcement learners," in *ICML*, 2017, pp. 1156–1164.

[32] K. He, H. Fan, Y. Wu *et al.*, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[33] J.-B. Grill, F. Strub, F. Altché *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *NeurIPS*, pp. 21 271–21 284, 2020.

[34] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021, pp. 15 750–15 758.

[35] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *EMNLP*, 2021, pp. 6894–6910.

[36] Z. Mao, Z. Li *et al.*, "Jointly contrastive representation learning on road network and trajectory," in *CIKM*, 2022, pp. 1501–1510.

[37] J. Zhu, Y. Xia, L. Wu *et al.*, "Masked contrastive representation learning for reinforcement learning," *IEEE TPAMI*, 2022.

[38] X. Du, H. Chen *et al.*, "A contrastive-enhanced ensemble framework for efficient multi-agent reinforcement learning," *Expert Systems with Applications*, p. 123158, 2024.

[39] A. Stooke, K. Lee *et al.*, "Decoupling representation learning from reinforcement learning," in *ICML*, 2021, pp. 9870–9879.

[40] T. Wu, P. Zhou *et al.*, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE TVT*, vol. 69, pp. 8243–8256, 2020.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

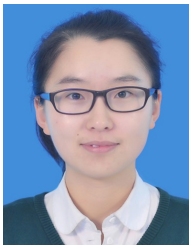
[42] J. Pearl *et al.*, "Models, reasoning and inference," *Cambridge, UK: CambridgeUniversityPress*, vol. 19, no. 2, p. 3, 2000.

[43] I. Goodfellow, J. Pouget-Abadie *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, pp. 139–144, 2020.

[44] Q. Jiang, M. Qin, S. Shi *et al.*, "Multi-agent reinforcement learning for traffic signal control through universal communication method," in *IJCAI*, 2022, pp. 3854–3860.

[45] X. Liang, X. Du *et al.*, "A deep reinforcement learning network for traffic light cycle control," *IEEE TVT*, vol. 68, pp. 1243–1253, 2019.

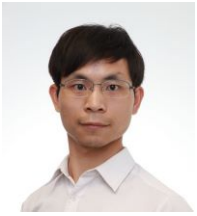
[46] G. Zheng, Y. Xiong, X. Zang *et al.*, "Learning phase competition for traffic signal control," in *CIKM*, 2019, pp. 1963–1972.



Xinqi Du is currently a Ph.D. student at the School of Artificial Intelligence, Jilin University, China. She received the B.E degree from the Computer Science and Technology, Jilin University, in 2018. Her research interests include multi-agent reinforcement learning, deep reinforcement learning and machine learning. She has served as a peer reviewer for the international journals, including IEEE IoTJ and IEEE Vehicular Technology Magazine.



Ziyue Li is an assistant professor in the Information System Department, WiSo Faculty, University of Cologne. He is also the Chief Machine Learning Scientist at EWI. His research targets at high-dimensional data mining and deep learning methodologies for real-world spatiotemporal problems. His expertises are tensor analysis, spatiotemporal data, and statistical machine learning. Those methods have been applied to various industries, mainly in smart transportation, as well as in smart manufacturing and multimedia. Dr. Li's works have been awarded various Best Paper awards in INFORMS, IISE, and IEEE CASE.



Cheng Long (S'11-M'15-SM'22) is currently an Assistant Professor at the School of Computer Science and Engineering, Nanyang Technological University. From 2016 to 2018, he worked as a lecturer at Queen's University Belfast, UK. He received his PhD degree from the Hong Kong University of Science and Technology, Hong Kong, in 2015, and his BEng degree from South China University of Technology, China, in 2010. His research interests are broadly in data management, data mining and big data analytics.



Yongheng Xing is currently pursuing the Ph.D. degree in the College of Computer Science and Technology, Jilin University, China. He received the B.E degree from the Computer Science and Technology, Jilin University, in 2018. His research interests include graph neural network, Internet of Things (IoTs) Data Analytics and machine learning. He has served as a peer reviewer for the international journals, including IEEE TII, IEEE IoTJ and Neurocomputing.



Philip S. Yu (Fellow, IEEE) is a Distinguished Professor and the Wexler Chair in Information Technology at Department of Computer Science, University of Illinois at Chicago. He is a Fellow of ACM and IEEE. Dr. Yu is the recipient of ACM SIGKDD 2016 Innovation Award, IEEE Computer Society's 2013 Technical Achievement Award and Research Contributions Award from ICDM in 2013. He has published more than 1,300 referred conference and journal papers cited more than 173,000 times with an H-index of 187. He has applied for more than 300 patents. Dr.Yu was the Editor-in-Chief of ACM TKDD (2011-2017) and IEEE TKDE (2001-2004).



Hechang Chen is an associate professor at the School of Artificial Intelligence, Jilin University, China. He received his PhD degree from the College of Computer Science and Technology, Jilin University, in 2018. From 2015 to 2016, he was a joint training PhD student with the Department of Computer Science, University of Illinois at Chicago (UIC), Chicago, USA. From 2017 to 2018, he was a visiting PhD student with the Department of Computer Science, Hong Kong Baptist University (HKBU), Hong Kong, China. He has published over 60 articles in many top international conferences and journals, e.g., IEEE TPAMI, TNNLS, NeurIPS, AAAI, IJCAI, SIGIR, WWW, ICDE, etc. His current research interests include machine learning, data mining, reinforcement learning, complex system, and knowledge engineering. He has served as a program chair (PC) for several international conferences and journals, including IEEE TKDE, TCYB, TITS, KDD, ICML, AAAI, IJCAI, WWW, WSDM, CIKM, etc.

Authorized licensed use limited to: JILIN UNIVERSITY. Downloaded on April 02,2024 at 07:31:33 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.