



FLight: optimization algorithm for traffic lights based on short-term traffic state forecast

Daimin Liu¹ · Jian Huang¹

Received: 18 July 2024 / Revised: 17 April 2025 / Accepted: 21 April 2025 /

Published online: 21 May 2025

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2025

Abstract

Intelligent transportation systems play a crucial role in building smart cities. However, existing traffic signal control systems heavily rely on knowledge of experts and lack real-time adaptability to dynamic traffic states. Most reinforcement learning-based approaches define traffic states using features such as current signal phases and queue lengths in each lane. Unfortunately, these methods overlook the spatiotemporal characteristics of real-world traffic flow and also fail to consider the moving vehicles when designing signal strategies. In this paper, we propose a Short-Term Traffic Flow Forecast Model that captures both temporal and spatial features of traffic flow. Our approach combines Temporal Convolutional Networks for extracting time series features from historical traffic flow, and Long Short-Term Memory networks to model complex temporal dependencies, and obtains more accurate time dependency relationships by assigning weights through self-attention layers. We also use Graph Attention Networks to learn spatial features between intersections and neighboring junctions. In addition, we have proposed an Enhanced Traffic State representation that captures both waiting and moving vehicles. We divide the lane into multiple segments based on the distance to the intersection and count the number of vehicles driving on different segments. In comparison to conventional traffic signal control methods and several reinforcement learning-based algorithms, our proposed algorithm has achieved superior results, especially, the performance can be increased by up to 4.57% on the real-world datasets.

Keywords Machine learning · Deep reinforcement learning · Adaptive traffic signal control · Multi-agent system · Attention mechanism · Traffic state forecast

Editor: Javier Garcia.

Daimin Liu and Jian Huang have contributed equally to this work.

✉ Jian Huang
hjian@jxufe.edu.cn

Daimin Liu
1021299316@qq.com

¹ School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Yuping Avenue, Nanchang 330032, Jiangxi, China

1 Introduction

The challenge of traffic congestion is an undeniable facet of contemporary society. In the field of conventional traffic signal control methods, FixedTime (Koonce, 2008) based on expert experience, sets a fixed signal duration plan tailored to the traffic flow at intersections, and is suitable for stable and unchanged traffic flows. MaxPressure (Varaiya, 2013), on the other hand, detects traffic flow information near the roadway through loop sensors and optimizes vehicle throughput efficiency at intersections by calculating traffic pressure in different directions. These methods have contributed to the research of control algorithms based on predefined rules. However, these conventional traffic signal control methods are no longer adequate for the current complex and ever-changing traffic states. The conventional traffic signal control systems are difficult to handle complex dynamic traffic flow, but reinforcement learning (RL) is good at dealing with this problem (Wei et al., 2021).

The inception of RL's application to traffic control was marked by Aslani et al. (2017) in 1996, this work catalyzed a novel trajectory of research within the sector. El-Tantawy and Abdul Hai El-Tantawy and Abdulhai (2010) proposed an algorithm predicated on Q-Learning for the governance of single intersection controls. This algorithm utilized the length of vehicle queues as the traffic state input, focusing on a biphasic intersection as the experimental subject. The method's efficacy was corroborated through the alteration of phase sequences; building on this, Camponogara and Scherer Camponogara and Scherer (2010) executed enhancements and augmentations to the intersection's phase sequence architecture. The contribution of the above work introduced RL into traffic signal control, making RL become a feasible method in the field of intelligent transportation.

The advent of the Deep Q Network (DQN) algorithm by Van Hasselt and Mnih Van Hasselt et al. (2016) and Mnih et al. (2015) marked a significant milestone, addressing the challenge of data dimensionality explosion inherent in Q-Learning and propelling the vigorous advancement of deep reinforcement learning (DRL) within traffic signal control. IntelliLight (Wei et al., 2018) pioneered the utilization of DQN in this realm, the contribution of this work is to quantify complex traffic information into the states and rewards required for DRL. DemoLight (Xiong et al., 2019) suggested that agents could enhance initial parameters by learning from models with proven efficacy. This work implies that RL-based intelligent transportation models do not need to be trained from scratch. Instead, they can learn optimal initialization parameters from other models, significantly reducing the cost of model training. MetaLight (Zang et al., 2020) embraced meta-learning to boost the model's versatility across diverse intersections, obviating the need for retraining in novel environments. This research work demonstrates that the use of meta-learning can help models better adapt to traffic flows at different intersections, enhancing the transferability of intelligent transportation models across various intersections. PressLight (Wei et al., 2019a) integrated the concept of maximum pressure control from the MaxPressure, leveraging pressure as a metric for reward. This study reveals that improving the reward function can effectively enhance model performance. Additionally, incorporating methods from conventional traffic signal control can effectively improve RL-based intelligent transportation models. MPLight (Chen et al., 2020) has been tested for the first time in real-world scenarios involving one thousand intersections and traffic flows. This work confirms that RL models are capable of handling intelligent traffic signal control in large-scale areas.

In Li's research (Li et al., 2022), the Cooperative Max-Pressure was proposed as a system state that combines the max-pressure states of both the current intersection and neighboring

intersections, utilizing Graph Attention Network (GAT) (Veličković et al., 2017) to aggregate information from neighboring nodes. It was demonstrated that learning spatial characteristics of traffic flow could effectively enhance traffic signal control efficiency. Jiang's work (Jiang et al., 2021) integrated traffic flow forecast with traffic light control by employing Long Short-Term Memory (LSTM) networks (Hochreiter, 1997) to forecast traffic patterns. This approach enables early identification of potential congestion in subsequent time intervals, thereby providing sufficient time for optimizing signal timing plans or implementing other congestion mitigation measures, which validates the significance of traffic forecast for traffic signal control optimization. Previous research commonly utilized Simulation of Urban Mobility (SUMO) (Lopez et al., 2018) as a traffic simulation platform, which faced limitations in scaling road networks and traffic flows. To address this, Shanghai Jiao Tong University developed CityFlow (Zhang et al., 2019) as a novel simulation platform featuring fundamentally optimized data structures and efficient algorithms. CityFlow supports flexible definitions of road networks and traffic flows based on both synthetic and real-world data, while providing user-friendly interfaces for RL applications. Most notably, CityFlow achieves over 20 times faster computational speed compared to SUMO and enables city-wide traffic simulations. These findings collectively indicate that improved traffic state representation methods and traffic flow forecast can significantly enhance traffic signal control performance. Therefore, this research focuses on advancing both traffic state representation methodologies and traffic flow forecast techniques.

We have pinpointed several predominant issues with the existing methodologies:

1. The control of traffic signals has been myopically focused on the immediate state information, without considering the ramifications of past traffic states.
2. The algorithms have solely catered to the needs of queuing vehicles, failing to account for the requirements of vehicles currently moving.

In response to these challenges, we introduce the Short-Term Traffic Flow Forecast Model (STTFFM), designed to assimilate the short-term spatiotemporal dynamics of the road network to furnish real-time traffic state forecasts. Furthermore, we have integrated data pertaining to moving vehicles to refine the accuracy of traffic state assessments. Our contributions are encapsulated as follows:

1. The STTFFM is unveiled for the prognostication of short-term traffic states, leveraging Temporal Convolutional Networks (TCN) (Bai et al., 2018) and LSTM algorithms, complemented by a self-attention mechanism (Vaswani, 2017) to decode temporal patterns, and GAT to elucidate the spatial interconnections among roadways.
2. The incorporation of moving vehicle data substantively elevates the efficacy of the traffic state enhancements.
3. Extensive empirical verification has confirmed the effectiveness of our improved method.

Sect. 2 delineates various methodologies pertinent to traffic signal control. Section 3 elucidates definitions germane to these methodologies. Section 4 expounds on our traffic light optimization algorithm predicated on short-term traffic state forecast. Section 5 showcases experimental comparisons between our approach and other models. Section 6 encapsulates a summation of the research presented in this paper.

2 Related work

2.1 Conventional traffic signal control

The conventional traffic signal control paradigms are quintessentially segmented into four classifications: FixedTime control (Koonce, 2008), which operates on a predetermined schedule, with established cycle lengths, phase sequences, and durations, offering consistency but lacking responsiveness to real-time conditions. While timing control algorithms are simple and practical, they do possess certain limitations. They rely on manual experience and historical data for control, lacking intelligence and adaptability, and are unable to make real-time adjustments based on traffic states. As a result, their effectiveness is limited. Actuated control (Gershenson & Rosenblueth, 2012; Cools et al., 2013), which dynamically adjusts signals based on immediate traffic data, with the green phase triggered when vehicle presence surpasses set thresholds, thus providing a degree of adaptability. Adaptive control (Lowrie & Scats, 1990; Hunt et al., 1982), which goes a step further by selecting the most suitable traffic plan from a prearranged array, informed by continuous data streams from loop sensors, aiming for optimal flow. Its primary advantage lies in its ability to adapt to real-time traffic states. However, it necessitates a significant number of sensors and devices, as well as precise actuarial calculations, for deployment. The most renowned examples of this approach are SCOOT (Hunt et al., 1982) and SCATS (Lowrie & Scats, 1990). The former is predominantly utilized in the UK and European markets, while the latter is widely used in Australia and Southeast Asian markets. Optimization-based control (Varaiya, 2013; Le et al., 2015; Sun & Yin, 2018) which frames signal control as an optimization issue within a defined traffic flow model, guided by empirical traffic observations. By monitoring traffic flow in real-time, the timing sequence of traffic lights is adjusted to maximize traffic flow.

2.2 RL-based traffic signal control

In recent years, thanks to the powerful fitting capabilities of deep neural networks, DRL has been extensively applied in traffic signal control. We will introduce the models, as well as the design of states and rewards.

The RL-based works have shown good performance, thanks to excellent deep neural networks and well-designed state.

(1) Model: AttendLight (Oroojlooy et al., 2020) uses the attention mechanism to obtain the features of each phase when constructing phase features. The FRAP (Zheng et al., 2019) brought forth the notion of phase competition, concentrating on diminishing the exploration space of the model via phase flipping to bolster performance. CoLight (Wei et al., 2019b) and PDLight (Zhao et al., 2020) consider the influence of surrounding intersections through graph neural networks. AttentionLight (Zhang et al., 2023) proposes an attention-based phase competition learning framework, which is automatically learned by the model. DERLight (Yang et al., 2024) adopts a dual experience replay pool method to enhance the learning efficiency of the agent by learning training data with different weights. SocialLight (Goel et al., 2023) collaborates to optimize traffic control strategies by estimating the contributions of its neighbor agents through distributed agents, hoping to achieve better scalability on larger-scale road networks. OfflineLight (Wu et al., 2024) introduces an offline adaptive decision-making

model trained using offline traffic data, demonstrating that high-performance RL models can be constructed without the need for online interaction. HG-M2I (Yang, 2023) introduces a hierarchical graph structure that provides us with an insight: we can utilize hierarchical networks to learn about regions and the whole separately, thereby enhancing the training efficiency of the model. Although the aforementioned methods enhance the effectiveness of agent decision-making by improving the model's network structure to more efficiently extract traffic features, they all overlook the fact that traffic flow is continuous in both time and space. Only by considering both the temporal and spatial characteristics of traffic flow can we improve vehicle travel efficiency across the entire road network. Our work employs TCN as the feature extractor in the first stage, followed by a LSTM network in the second stage for more advanced temporal modeling. This division of labor ensures that each component focuses on different aspects, thereby reducing redundancy. A self-attention layer is introduced after the LSTM to further concentrate on crucial time steps. GAT dynamically learns the spatial dependency strengths between nodes through an attention mechanism. In this way, both temporal and spatial characteristics of traffic flow in the road network are learned simultaneously.

(2) State: LIT (Zheng et al., 2019) uses the current signal phase and the number of vehicles on each lane as the state representation, and the reward is the queue length. IAColight (Hassanjani et al., 2023) reshapes the rewards of adjacent agents by combining the unfair aversion model to improve the transit time of vehicles within the network. PRGLight (Chenguang et al., 2021) found that the pressure mechanism ignored the remaining capacity of the exit lane, and then calculate the pressure mechanism of the remaining capacity of the exit lane. DataLight (Zhang & Deng, 2023) proposes to use offline data learning, which can narrow the gap between the traffic simulation simulator and the real world. GODA (Huang et al., 2024) learns to represent the original offline dataset with new data that is more selectively targeted towards higher reward objectives during comprehensive distribution generation, thereby maximizing the utility of limited optimal demonstrations. EfficientLight (Wu et al., 2021) proposes the concept of effective pressure to improve the state information features. IVPL (Yazdani et al., 2023) has designed an expanded reward function to capture delays caused by vehicle-to-vehicle, vehicle-to-pedestrian, and pedestrian-to-pedestrian interactions, as well as red light delays for both vehicles and pedestrians. Without increasing the complexity of the network structure, this clever reward design helps agents improve intersection traffic efficiency. PRLight (Han et al., 2023) recognizes the lag effect of traffic signals on traffic flow and mitigates this lag by using forecast states as the input states for the agent. It also demonstrates that learning the temporal relationships in traffic flow contributes to improving traffic efficiency. Salah Bouktif, Abderraouf Cheniki (Bouktif et al., 2023) adopts a strategy where states and rewards are aligned. This alignment facilitates the agent's optimization towards the goal, reducing fluctuations during the model training process and accelerating model convergence. The aforementioned work has demonstrated that precise and effective state representation can not only improve the accuracy of agent decision-making but also reduce unnecessary data feature dimensions, thereby accelerating model convergence. Despite the requirement for concise and effective state features, these studies have only considered the travel demands of vehicles stationary at intersections waiting to pass, neglecting the demand for green traffic signals by vehicles that are currently moving. As a result, excessive vehicles may accumulate at intersections before the next signal cycle begins, impacting the overall traffic efficiency of the road network.

2.3 Large language model based traffic signal control

LLMLight (Lai et al., 2023) uses the advanced generalization ability of large language models to make effective traffic signal control, using ChatGPT-4 to control traffic signal lights at intersections. TransformerLight (Wu et al., 2023) leverages sequence modeling to construct high-level adaptive models for traffic signal control systems. Decision Transformer (Meng et al., 2023) utilizes a pre-trained decision model based on offline data to serve as the decision-maker for traffic signal control systems. However, currently, general large language models do not exhibit significant performance advantages over RL models in the field of traffic signal control. Instead, their immense number of parameters can impose additional overhead on the control system.

3 Preliminary

3.1 Definition

Definition 1 (*Road Network*) A road network is typically viewed as a directed graph, where intersections are considered as nodes, and edges are considered as the roads connecting these intersections.

Definition 2 (*Entrance and Exit Lanes of Traffic Intersections*) As illustrated in Fig. 1a, each intersection has four approach channels, namely North, East, South, and West. Three lanes, based on their distance from the road centerline, from inner to outer, are denoted as l, k, m respectively.

Definition 3 (*Traffic Movement*) Traffic movement is defined as the movement of vehicles from intersection x to intersection y , shown as $e(x, y)$. As shown in Fig. 1b, there are 12 types of traffic movements, among which 4 are right-turn movements for each road.

Definition 4 (*Traffic Signal Phase Set*) As shown in Fig. 1d, an intersection typically sets up four or eight signal phases, referred to as the phase set P . Each signal phase p consists of two non-conflicting traffic movements in Fig. 1c.

3.2 Problem: multi-intersection traffic signal control

In this section, we formulate the traffic signal control problem as a Markov Decision Process (MDP). Each intersection equipped with traffic signal lights in the road network is regarded as an agent. Each agent selects the next action the execution signal for the upcoming interval that minimizes the average travel time for all vehicles. The aforementioned problem can be described as $(S_t, A_t, R_{t+1}, S_{t+1}, \gamma)$.

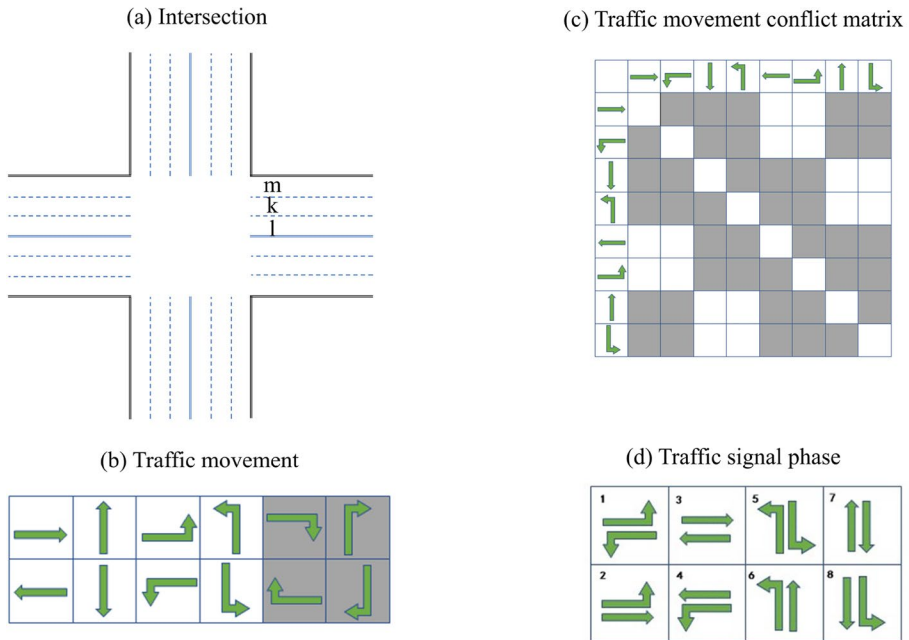


Fig. 1 **a** Is a standard 4-way intersection and there are 3 lanes in each approach, three lanes, based on their distance from the road centerline, from inner-to-outer, are denoted as l , k , m respectively; **b** shows the 12 traffic movements; **c** tell us the reason of two traffic movements can be combined to form a legal signal phase; **d** shows the legal traffic signal phases

3.2.1 State space S

At time step t , the agent observes state S from the environment, which serves as the basis for making decisions. In this paper, we select the traffic signals at the current intersection and the Enhanced Traffic State (ETS), which we construct for each lane to record queuing vehicles and running vehicles, as the state features. A detailed introduction to the ETS will be provided in the Sect. 4.

3.2.2 Action space A

At time t , the agent selects the signal phase to activate for the next time interval at the current intersection. We adopt an action space consisting of 8 phases.

3.2.3 Reward R

At time $t + 1$, the agent receives a reward for the action chosen at time t . Although we use average travel time as an evaluation metric, it cannot be directly optimized. Based on previous work (Wei et al., 2019b), we use average queue length as the reward.

3.2.4 Discount factor γ

The agent's goal is to maximize expected cumulative reward, which necessitates a focus on long-term future rewards. To balance between current and future rewards, the agent employs a discount factor γ . The value of γ is between 0 and 1.

The computational process of an MDP is formulated as shown in Eq. 1:

$$Q(s_t, a_t) = r(s_t, a_t; \theta) + \gamma * \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta) \quad (1)$$

4 Method

In this section, as illustrated in Fig. 2, we commence by introducing a straightforward yet effective enhancement to state design, termed ETS. Subsequently, we present the STTFFM, which is utilized to learn both the temporal and spatial characteristics of traffic flow within the road network. Our methodology employs ETS alongside the current signal phase as the state. After learning the temporal and spatial features through STTFFM, we leverage DQN as the Traffic Control Module (TCM) for training. A brief overview of this architecture is provided at the end of this section.

4.1 Enhanced traffic state

The state space is a crucial component of DRL algorithms, as it determines the extent of information the algorithm can extract from the environment. Enhancing the design of the state space, such as incorporating additional traffic parameters or adopting more efficient methods of information representation, can enable the algorithm to better comprehend and navigate complex traffic scenarios, thereby facilitating more accurate traffic signal control.

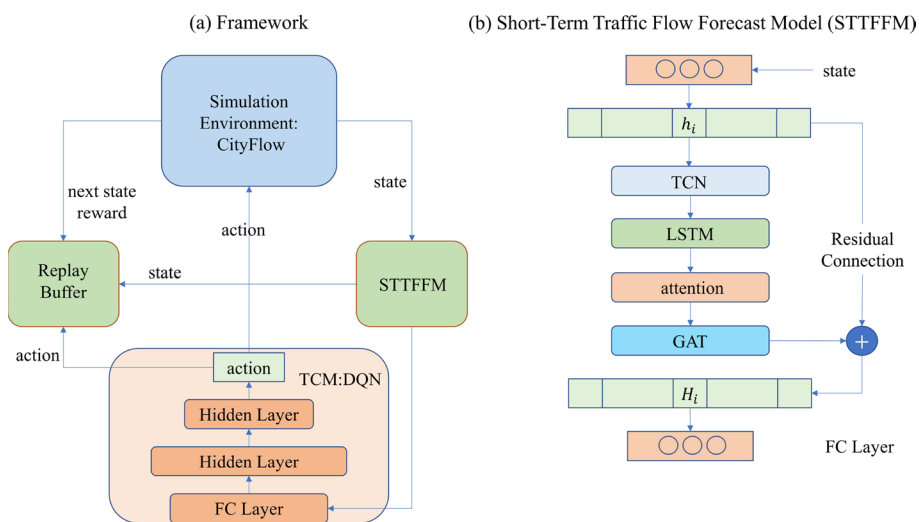


Fig. 2 **a** Is framework and **b** is network of STTFFM

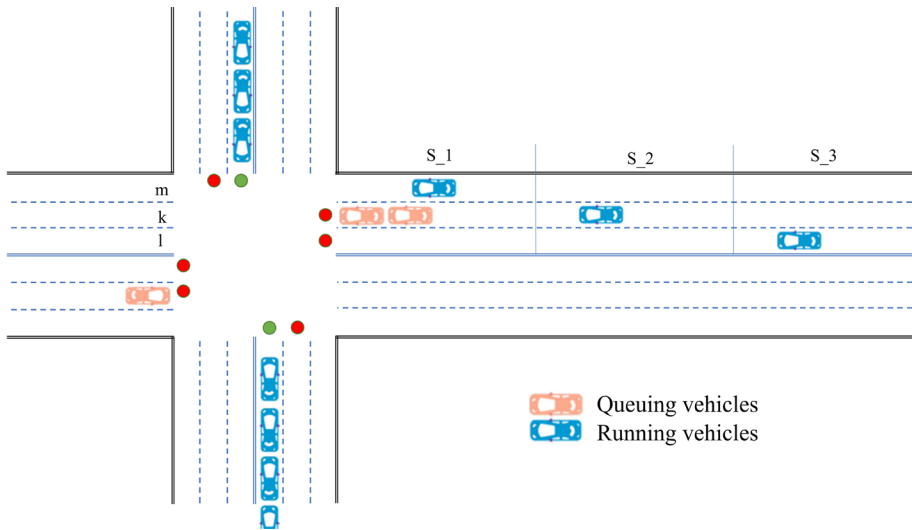


Fig. 3 If only the traffic demand of queuing vehicles at the intersection is considered, signal phase 3 in Fig. 1d should be activated at the current intersection. However, this would result in more vehicles waiting at the intersection. The correct approach is to incorporate the traffic demand of vehicles traveling on the S_1 into consideration and activate signal phase 5 in Fig. 1d. This allows more vehicles to pass through the intersection, thereby enhancing traffic efficiency

As shown in Fig. 3, red represents queuing vehicles and blue represents running vehicles. The roadway is divided into three segments based on proximity to the intersection: a near segment (S_1), a middle segment (S_2), and a far segment (S_3). Subsequently, the number of vehicles traveling in each approaching lane is counted separately.

We denote the number of queuing vehicles in the lanes at the intersection as $q(l)$, and the number of running vehicles in lanes S_1 , S_2 , and S_3 as $q(l, S_1)$, $q(l, S_2)$, and $q(l, S_3)$, respectively.

So, we set the equation of ETS as follow:

$$ETS(l) = (q(l), q(l, S_1), q(l, S_2), q(l, S_3)) \quad (2)$$

By utilizing the ETS to represent information about both queuing vehicles and running vehicles, the agent can rely on a more comprehensive set of traffic information to make more appropriate decisions.

4.2 Short-term traffic flow forecast model

As shown in Fig. 2b, the STTFFM can learn temporal and spatial relationships. Temporal relationships are learned through TCN, LSTM, and self-attention, while spatial relationships are learned through the GAT.

The advantages of TCN in the first stage include their proficiency in capturing long-range dependencies and their superior parallel computing capabilities. Through the use of dilated causal convolutions with exponential dilation factors, TCN can capture long-distance temporal dependencies at a relatively low computational cost, with their receptive field growing exponentially as the number of layers increases. Furthermore, the

convolutional operations in TCN possess inherent parallelism, allowing their time complexity to be reduced to $O(1)$ compared to the sequence-dependent computations of LSTM, which have a time complexity of $O(T)$.

The necessity of LSTM in the second stage arises from its capabilities in modeling temporal dynamics and enhancing sequence attention. LSTM's gating mechanisms (forget gate, input gate, output gate) excel at modeling nonlinear temporal dynamics and can handle time-varying features that are difficult for TCN to capture. The hidden states of LSTM can be viewed as dynamically weighted over time steps, complementing the fixed-weight convolutions of TCN. When GAT are employed after TCN and LSTM, TCN and LSTM handle the temporal dimension, while GAT focus on the spatial dimension, enabling decoupled learning of spatiotemporal features. In traffic flow forecast, TCN extract temporal flow patterns, while GAT model the impact of road topology.

Additionally, TCN extract locally invariant features through weight sharing in convolutional kernels, while LSTM model globally time-varying features through their gating mechanisms. These two components are orthogonal in feature space, producing a synergistic effect that enhances model performance.

We employ the TCN as the first stage of the STTFFM to extract time series features. In terms of representation, the sequence input can be denoted as $x \in R^n$, with the convolutional kernel represented as $f : \{0, \dots, k-1\} \rightarrow R$. Therefore, the dilated convolution operation at time t is as shown in Eq. 3:

$$x(t) = (x * f)(t) = \sum_{i=0}^{k-1} f(i) \cdot x_{t-i \cdot d} \quad (3)$$

d represents the dilation factor, and k is the size of the convolutional kernel.

After extracting locally invariant features using TCN, we then employ LSTM to model globally time features.

The LSTM is an enhanced Recurrent Neural Networks (RNN) (Cho et al., 2014) architecture. We utilize LSTM as the second part of network to learn time dependencies within the historical traffic flow sequences. The computation processes for the LSTM are detailed in Eq. 4 to Eq. 9.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_h) \quad (6)$$

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (7)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t * \tanh(C_t) \quad (9)$$

The sigmoid function is denoted as σ , input gate is i_t , forget gate is f_t , memory cell is c_t , and output gate is o_t , where $*$ represents multiplication, hidden layer is h_t .

The self-attention is a specialized attention network. We apply the self-attention to assign weights to the temporal features learned by TCN and LSTM, enhancing the model's performance. The computation formula is as follows.

$$s(h_i, q) = h_i^T q \quad (10)$$

$$\alpha_i = \frac{\exp(s(h_i, q))}{\sum_{j=1}^N \exp(s(h_j, q))} \quad (11)$$

$$att(H, q) = \sum_{i=1}^N \alpha_i h_i \quad (12)$$

In the self-attention layer, the input is denoted as $H = h_i, i = 1, 2, 3, \dots, n$, and q represents the query matrix.

In the GAT, assuming the number of intersections in the road network is N , the set of state features for the entire network can be represented as $h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}, \vec{h}_i \in R^F$. In the attention layer, at least one learnable linear transformation is required to maintain sufficient expressive power and transform the input features into higher-order features.

Then we normalize the importance scores for intersection i :

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_j]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i || W\vec{h}_k]))} \quad (13)$$

Finally, the feature that aggregates neighboring intersections' features for intersection i can be constructed as follows:

$$\vec{h}'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} W\vec{h}_j \right) \quad (14)$$

The concatenation operation is denoted by $||$.

Through the learning process of these four layers of networks, the STTFM can effectively learn both the temporal and spatial features of the traffic state within the road network.

L is TCN layers, k is convolutional kernel size, T is input sequence length and C is the number of input and output channels. TCN's time complexity are approximately equal to $O(LkTc^2)$.

L is LSTM layers, T is sequence length, H is hidden layer dimension. LSTM's time complexity are approximately equal to $O(LTH^2)$.

Benefit from the parameter sharing mechanism (L hidden layers and each layer has m neurons), the time and space required are independent of the number of intersections. GAT's time complexity are approximately equal to $O(m^2L)$.

4.3 Framework and algorithm

As shown in Fig. 2a, in this paper, we employ the DQN algorithm to train our agent. Data generated from the interaction between the agent and the environment are stored in an

experience replay buffer; neural networks are updated by replaying these experience samples. The pseudo code for our model is presented in Algorithm 1.

Algorithm 1 FLight:Optimization algorithm for traffic lights based on short-term traffic flow forecast

```

1: Initialize total training rounds: Episodes, learning rate:  $\alpha$ , discount factor:  $\gamma$ ,
   sample size:  $B$ , and experience replay pool size:  $D$ 
2: Initialize the model parameters of STTFFM and TCM
3: for episode = 0 to Episodes do
4:   Initialize the traffic simulator environment to get the initial environment state
   information
5:   Store the state information in the historical time series
6:   for episode = 0 to Episodes do
7:     Obtain the forecast state  $s_p$  through the STTFFM
8:     Use  $s_p$  as the state input for the agent model TCM
9:     TCM outputs action  $a_t$ , the environment executes the action and returns
   the new environment state  $s_{t+1}$  and reward  $r_{t+1}$ 
10:    Store the sample data  $\langle s_t, a_t, r_{t+1}, s_{t+1} \rangle$  in the experience replay pool
11:    Update TCM model parameters through experience replay
12:    Train the STTFFM and update parameters of STTFFM
13:   end for
14: end for

```

5 Experiment

5.1 Experimental environment

CityFlow (Zhang et al., 2019) is an open-source and efficient traffic simulation tool that permits flexible definition of road networks and traffic flows for the purpose of testing both conventional and RL-based traffic signal control methodologies.

The green signal duration of each signal phase is set to 10 s, and after each green signal ends, a three seconds yellow signal and a two seconds red signal will be added to clear the vehicles in the intersection, so as to start the next duration. Each experimental simulation lasts for 3600 s.

All experiments were conducted using the same simulator environment, and the traffic flow data utilized were complete and identical across all tests.

5.2 Dataset

To comprehensively compare the effectiveness of the proposed method with previous approaches, we have selected classic and commonly used datasets, specifically real-world datasets from Jinan (China) and Hangzhou (China), as shown in Fig. 4.

Jinan dataset: As shown in Table 1, there are a total of 12 intersections in the road network. Each intersection is four way, with two north–south 800 ms long road sections and

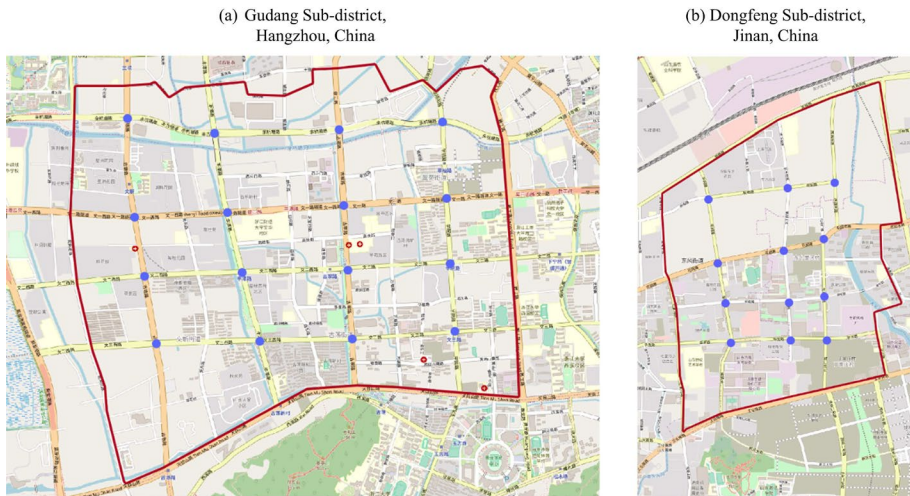


Fig. 4 Road networks for real-world datasets. Red polygons are the areas we select to model, blue dots are the traffic signals we control (Color figure online)

Table 1 Traffic data statistics

Datasets	Intersections	Vehicles	Arrival rate (vehicles/5 min)				(east–west, north–south)
			Mean	Std	Max	Min	
Jinan1	12 (3x4)	6295	532.67	98.52	671	255	(400 m, 800 m)
Jinan2	12 (3x4)	4365	362.83	74.81	493	236	(400 m, 800 m)
Jinan3	12 (3x4)	5495	456.92	46.38	544	362	(400 m, 800 m)
Hangzhou1	16 (4x4)	2983	247.67	40.44	332	211	(800 m, 600 m)
Hangzhou2	16 (4x4)	6984	581.08	318.43	1145	202	(800 m, 600 m)

two east–west 400 ms long road sections. There are a total of three traffic flow datasets in Jinan dataset.

Hangzhou dataset: There are a total of 16 intersections in the road network, shown in Table 1. Each intersection is four-way, with two east–west 800 ms long road sections and two north–south 600 ms long road sections. There are a total of two traffic flow datasets in Hangzhou dataset.

The datasets employed in this study constitute open-access benchmarks originally released with CityFlow, derived from real-world vehicle trajectory data to ensure ecological validity of traffic patterns. The traffic flow integration framework within CityFlow is structured around two core configuration files: "Roadnet" defining road network topology and "Flow" specifying vehicle movement parameters. Roadnet file defines the roadnet structure. CityFlow's roadnet mainly consists of intersections and roads (see them as nodes and edges of a graph). Flow file defines the traffic flow. Each flow contains following field. Vehicle: Defines the parameter of vehicle. Route: Defines the route, all vehicles of this flow will follow the route. Specify the source and the destination, optionally some anchor points and the router will connect them with shortest paths automatically. Interval: Defines

Table 2 Attributes of vehicle

Attribute	Value	Unit
Length	5	m
Width	2	m
Maximum acceleration	2	m/s ²
Maximum deceleration	4.5	m/s ²
Usual acceleration	2	m/s ²
Usual deceleration	4.5	m/s ²
Maximum cruising speed	11.11	m/s ²
Minimum acceptable gap with leading vehicle	2.5	m
Desired headway time with leading vehicle	2	s

Table 3 Hyper-parameters of models

Model parameter	Value
Learning rate	0.001
Batch size	20
Sample size	3000
Maximum memory	12000
γ	0.8
ϵ	0.8
ϵ decay	0.95
Minimum ϵ	0.01

the interval of consecutive vehicles (in seconds). If the interval is too small, vehicles may not be able to enter the road due to blockage, it will be held and let go once there are enough space. StartTime: Flow will generate vehicles between startTime and endTime (in seconds).

5.3 Experiment setting

In this section, we will introduce the settings of the experimental environment along with the parameter configurations for all models.

(1) Environment Setting: In all datasets we used, vehicles share the same attributes as shown in Table 2.

(2) Experimental Setting: In the TCM, to ensure methodological consistency and mitigate potential discrepancies arising from variations in network parameter configurations, the reinforcement learning model employs identical hyper-parameters to those utilized in the AttentionLight method. The parameter settings are detailed in Table 3.

(3) Parameters of STTFFM: The parameters of STTFFM are shown in Table 4. Based on empirical observations from prior research, excessive stacking of LSTM layers not only fails to enhance performance but significantly increases network complexity. Our architectural design therefore leverages the parallel computing advantages of TCN to improve computational efficiency, while subsequently employing LSTM's sequential modeling

Table 4 Parameters of STTFFM

Model parameter	Value
Learning rate	0.001
Batch size	8
Dropout	0.2
Kernel size	3
Layers of TCN	2
Layers of LSTM	1
Layers of GAT	2

capabilities to reconstruct temporal dependencies. The final parameters were systematically optimized through grid search guided by architectural considerations.

5.4 Evaluation metrics

Average Travel Time: This calculates the average travel time consumed by all vehicles entering and leaving the road network. This is the most commonly used indicator in traffic signal control problems. The unit of this indicator is seconds, and the smaller it is, the better the performance.

5.5 Comparison methods

We will compare the performance with the following methods, including conventional traffic methods and DRL methods.

Conventional Methods:

FixedTime (Koonce, 2008) is a fixed duration set by experts, using predefined schemes to divide different phase durations, and the scheme cannot be adjusted by itself.

MaxPressure (Varaiya, 2013) is based on the concept of maximum traffic pressure in the field of traffic control, and each time the signal phase with the maximum traffic pressure is selected.

Max-QueueLength (Zhang et al., 2023) uses max queuing vehicles as the state and reward.

RL-based Methods:

MPLight (Chen et al., 2020) uses FRAP (Zheng et al., 2019) as the main framework and introduces the concept of maximum pressure into state and reward design.

CoLight (Wei et al., 2019b) uses GAT to implement cooperation between adjacent intersections.

Efficient-MPLight (Wu et al., 2021) is based on the FRAP (Zheng et al., 2019) model, using effective traffic movement pressure and the current signal phase as observation states, and the reward uses intersection pressure.

Efficient-CoLight (Wu et al., 2021) is based on the CoLight model, using effective traffic movement pressure and the current signal phase as observation states.

AttentionLight (Zhang et al., 2023) compares the advantages and disadvantages of queue length and intersection pressure as state information, and at the same time proposes an attention-based phase competition learning framework, which is automatically learned by the model.

OfflineLight (Wu et al., 2024) introduces an offline adaptive decision-making model trained using offline traffic data, demonstrating that high-performance RL models can be constructed without the need for online interaction.

GODA (Huang et al., 2024) learns to represent the original offline datasets with new data that is more selectively targeted towards higher reward objectives during comprehensive distribution generation, thereby maximizing the utility of limited optimal demonstrations.

Decision Transformer (Meng et al., 2023) utilizes a pre-trained decision model based on offline data to serve as the decision maker for traffic signal control systems.

TransformerLight (Wu et al., 2023) leverages sequence modeling to construct high-level adaptive models for traffic signal control systems.

5.6 Experimental results

In this section, we compare our algorithm against other comparative algorithms in terms of evaluation metrics and convergence rate. The comparative data in Table 5 are all quoted from previous work. The results of FixedTime, MaxPressure, Max-QueueLength, CoLight, FRAP, MPLight and AttentionLight are referenced from AttentionLight (Zhang et al., 2023), The results of Efficient-MPLight and Efficient-CoLight come from Efficient Pressure (Wu et al., 2021), The data of the OfflineLight method comes from OfflineLight (Wu et al., 2024), The results of GODA comes from GODA (Huang et al., 2024), The results of Decision Transformer and TransformerLight comes from TransformerLight (Wu et al., 2023).

Table 5 Performance comparison of different methods evaluated on real datasets from Jinan and Hangzhou (average travel time (s), the smaller the better)

Model	Average travel time				
	Jinan			Hangzhou	
	1	2	3	1	2
Fixed time	429.27	370.34	384.89	497.87	408.31
Max pressure	274.99	246.41	244.63	289.54	349.85
Max-QueueLength	268.87	240.02	238.51	284.32	325.44
CoLight	271.17	251.22	248.87	300.07	339.76
FRAP	299.56	268.57	269.20	308.73	355.80
MPLight	297.68	274.32	268.00	313.16	355.35
Efficient-MPLight	261.81	241.35	238.80	284.49	321.08
Efficient-coLight	256.84	239.58	236.72	282.07	324.27
Attention light	254.82	239.68	236.64	283.64	316.38
Offline light	263.31	241.13	241.30	281.50	335.33
GODA	255.30	235.10	230.10	278.00	315.40
Decision transformer	274.79	244.19	254.45	284.02	372.01
Transformer light	256.16	235.78	232.39	273.18	313.91
FLight (ours)	246.62 ± 3.84	231.63 ± 0.78	228.04 ± 0.57	268.01 ± 0.51	303.05 ± 3.58

Our method has achieved the best performance on average travel time, both better than conventional traffic signal control methods and RL-based methods. The indicators of average travel time are shown in Tables 5.

Our method consistently demonstrates performance improvements across five datasets. For different traffic flow datasets in Jinan, the average improvement rate is 3.40%, while in Hangzhou, it is 4.57%. I attribute this performance enhancement to learning the spatiotemporal relationships of traffic flow and employing more accurate representations of traffic states. As the road network scales up, the effectiveness of our proposed method becomes even more pronounced, further validating the importance of understanding spatiotemporal traffic dynamics in complex scenarios.

FixedTime heavily relies on the prior knowledge of experts in the transportation field, necessitating expert input to allocate fixed signal phase times, which is inadequate for managing dynamically fluctuating traffic volumes. The MaxPressure approach hinges on the concept of maximum pressure control based on the assumption that downstream lanes possess unlimited vehicle capacity. However, in real-world traffic scenarios, the traffic flow at upstream intersections directly impacts those downstream. Consequently, the greedy strategy employed by the MaxPressure method in addressing multi-intersection signaling issues leads to interference between adjacent intersections and fails to effectively learn strategies from environmental feedback. Our methodology, grounded in RL, further enhances vehicle throughput efficiency across the entire road network by learning the temporal and spatial relationships within traffic flows.

CoLight employs queuing vehicle information as input states while aggregating information from neighboring intersections through graph neural networks to make decisions. OfflineLight learns from offline traffic data, achieving performance close to that of models trained on online data. AttentionLight replaces manual phase competition settings by leveraging an attention mechanism to identify competitive relationships among different signal phases. TransformerLight utilizes large language models and Transformers to learn how to optimize traffic signals, with its learning data sourced from RL-based model state-of-the-art (SOTA) approaches. While it contributes to the interpretability of decisions, it does not outperform RL methods in terms of performance. The aforementioned research efforts primarily rely on queuing vehicle information, neglecting the needs of running vehicles to cross intersections. Furthermore, they fail to exploit the temporal continuity of traffic flow, causing agents to overly focus on immediate rewards. Our method incorporates running vehicles into the state representation, enabling agents to access more comprehensive traffic information. Additionally, it employs STTFFM to learn the temporal and spatial characteristics of traffic flow, thereby enhancing the overall traffic efficiency of the road network and reducing vehicles' waiting time at intersections.

We compared the convergence speed of our model with several representative research methods, as shown in Fig. 5. The red curve demonstrates superior convergence rate and stability. Our method tends to converge after 100 epochs, exhibiting better stability and outstanding performance.

5.7 Ablation experiment

In order to more intuitively display the effectiveness of STTFFM based on spatiotemporal relationships and ETS. Ablation experiments were conducted under 5 different traffic configurations in Jinan and Hangzhou. The average travel time are shown in Table 6.

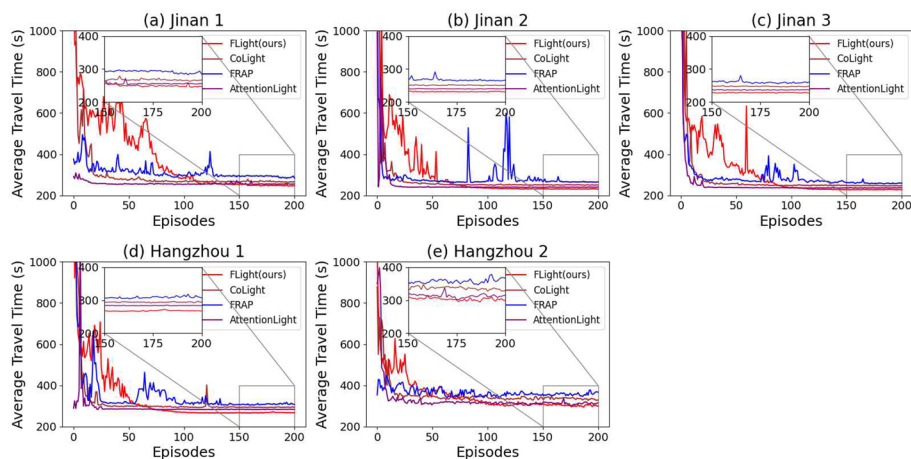


Fig. 5 The convergence rate of the model is based on raw data without any smoothing processing. The red line represents our proposed method (Color figure online)

Table 6 Performance comparison of different methods evaluated on real datasets from Jinan and Hangzhou (Average travel time (s), the smaller the better)

Model	Jinan			Hangzhou	
	1	2	3	1	2
origin	279.88	254.08	263.35	303.38	398.17
FLight-ETS	263.53	245.49	242.48	287.33	327.31
FLight-temporal	250.61	233.66	234.24	272.12	316.40
FLight-spatial	251.38	236.52	231.10	274.23	330.26
FLight	246.62	231.63	228.04	268.01	303.05

We conducted ablation experiments for the proposed method, as shown in Table 6. FLight represents our complete model. FLight-ETS replaces our proposed ETS with a simple state that only uses queuing vehicle information. FLight-temporal utilizes LSTM to replace the TCN-LSTM model in STTFM. FLight-spatial employs Graph Convolutional Networks (GCN) (Kipf & Welling, 2016) to learn spatial relationships without utilizing GAT. Origin denotes the unmodified baseline model, which solely employs DQN for training. All our proposed modifications effectively enhance the model's performance.

In FLight-spatial, we utilized GCN instead of GAT to learn spatial relationships, resulting in an increase in average travel time. Our analysis indicates that GCN assigns equal weights to adjacent intersections, whereas traffic flow exhibits directional, with upstream intersections being more crucial to downstream intersections than vice versa. GAT, however, can learn this importance through its attention mechanism, focusing on more critical intersection information and thereby effectively enhancing model performance.

Similarly, when we replaced our proposed model with LSTM, the average travel time also increased, suggesting that our model combination using TCN-LSTM is more effective than LSTM in learning the temporal relationships of traffic flow and improving overall model performance.

5.8 Extended study 1

TCN is good at capturing long-term dependencies and demonstrates remarkable parallel computing capabilities. By employing dilated causal convolutions with exponential dilation factors, TCN can capture long-range dependencies with relatively low computational costs, as the receptive field grows exponentially with the increase in layers. Furthermore, the convolutional operations in TCN are parallelizable, significantly reducing computation time. The gating mechanism of LSTM excels at modeling nonlinear temporal relationships and can handle time-varying features that are challenging for TCN to capture. While TCN and LSTM address the temporal dimension, GAT focus on learning spatial relationships, achieving decoupled learning of spatiotemporal features. In traffic flow forecast, TCN extract temporal flow patterns, whereas GAT simulate the topological influence structure of roads. Additionally, TCN extract locally invariant features through weight-sharing convolutional kernels, while LSTM, with their gating mechanisms, synergize to enhance model performance. Consequently, we employ the combined model of TCN-LSTM-GAT.

The experiment depicted in Fig. 6 utilizes time-series data collected from traffic flow within the Hangzhou1.

As illustrated in Fig. 6, the Mean Absolute Error (MAE) of the combined TCN-LSTM model not only surpasses the MAE of standalone TCN and LSTM models but also outperforms the LSTM-TCN combination, aligning with our previously analyzed results.

5.9 Extended study 2

In Sect. 5.1, we introduce a unified setting for environmental parameters within the traffic simulation environment: The green signal duration of each signal phase is set to 10 s, and after each green signal ends, a three-second yellow signal and a two-second red signal will be added to clear the vehicles in the intersection, so the action interval of the agent is 15 s. If the signal phase is set to 15 s, the action interval of the agent will be 20 s.

We aim to investigate the impact of different signal phase durations on the model.

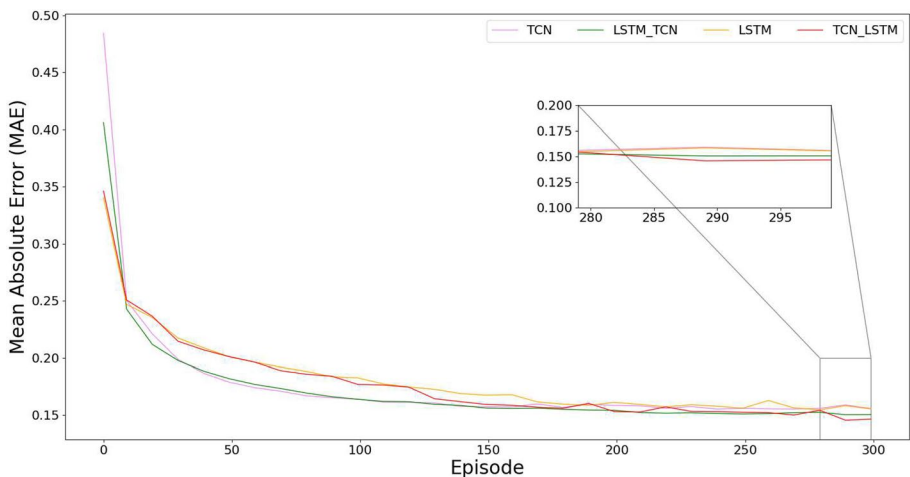


Fig. 6 The evaluation is mean absolute error (MAE), the smaller the better

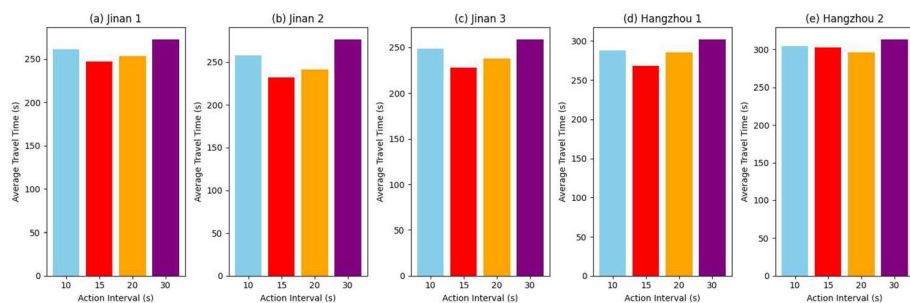


Fig. 7 The evaluation is average travel time, the smaller the better

As illustrated in Fig. 7, the model exhibits optimal overall performance when the action interval is set to 15 s. As the duration of signal phases increases, the average travel time rises, and model performance declines. Our analysis suggests that excessively long signal durations result in excessive waiting vehicles in lanes from other directions, leading to reduced traffic efficiency. However, we also observe an increase in average travel time when the action interval is 10 s. We have pondered over this phenomenon, noting that the simulator incorporates a 3 s yellow light and a 2 s red light to ensure the safe passage of vehicles within the intersection. Consequently, the remaining green light time available for vehicles to pass through is too short to guarantee that all vehicles within the road segment can clear the intersection, leading to excessive waiting vehicles. Therefore, an appropriate signal duration can similarly enhance model performance and improve traffic efficiency.

6 Conclusion

This paper presents a traffic state forecast model based on spatiotemporal relationships and an enhanced method for representing the state of vehicles of intersections. By forecasting traffic states and representing state features in more detail and effectiveness, the approach aids the agent in making more accurate strategies. Extensive experiments on real datasets with five different traffic flow configurations in Jinan and Hangzhou have demonstrated the effectiveness of our proposed methods.

Currently, our research employs fixed signal durations, neglecting the variability among vehicles on different roadways. In future work, we will consider utilizing dynamic signal durations to better optimize the travel time for vehicles in different directions.

Author Contributions Huang conceived this idea and conducted theoretical analysis, Liu conducted experiments and data analysis, Huang supervised and guided the experiment, Liu drafted the manuscript, and all authors reviewed the manuscript.

Funding Not applicable.

Data Availability The data used in this work is all public.

Code Availability Code will be made available after publication.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

References

- Aslani, M., Mesgari, M. S., & Wiering, M. (2017). Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies*, 85, 732–752.
- Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. [arXiv:1803.01271](https://arxiv.org/abs/1803.01271)
- Bouktif, S., Cheniki, A., Ouni, A., & El-Sayed, H. (2023). Deep reinforcement learning for traffic signal control with consistent state and reward design approach. *Knowledge-Based Systems*, 267, Article 110440.
- Camponogara, E., & Scherer, H. F. (2010). Distributed optimization for model predictive control of linear dynamic networks with control-input and output constraints. *IEEE Transactions on Automation Science and Engineering*, 8(1), 233–242.
- Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., & Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, pp. 3414–3421).
- Chenguang, Z., Xiaorong, H., & Gang, W. (2021). Prgligh: A novel traffic light control framework with pressure-based-reinforcement learning and graph neural network. In *IJCAI*.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. [arXiv:1406.1078](https://arxiv.org/abs/1406.1078)
- Cools, S.-B., Gershenson, C., spsampsps D’Hooghe, B. (2013). Self-organizing traffic lights: A realistic simulation. In *Advances in applied self-organizing systems* (pp. 45–55).
- El-Tantawy, S., & Abdulhai, B. (2010). An agent-based learning towards decentralized and coordinated traffic signal control. In *13th International IEEE conference on intelligent transportation systems* (pp. 665–670). IEEE.
- Gershenson, C., & Rosenbluth, D. A. (2012). Self-organizing traffic lights at multiple-street intersections. *Complexity*, 17(4), 23–39.
- Goel, H., Zhang, Y., Damani, M., & Sartoretti, G. (2023). Sociallight: Distributed cooperation learning towards network-wide traffic signal control. [arXiv:2305.16145](https://arxiv.org/abs/2305.16145)
- Han, X., Zhao, X., Zhang, L., & Wang, W. (2023). Mitigating action hysteresis in traffic signal control with traffic predictive reinforcement learning. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 673–684).
- Hassanjani, M., Alamiyan-Harandi, F., & Ramazi, P. (2023). Inequity aversion reduces travel time in the traffic light control problem. [arXiv:2302.12053](https://arxiv.org/abs/2302.12053)
- Hochreiter, S. (1997). *Long short-term memory*. Neural Computation MIT-Press.
- Huang, X., Member, D. W., & Boulet, B. (2024). Goal-conditioned data augmentation for offline reinforcement learning. [arXiv:2412.20519](https://arxiv.org/abs/2412.20519)
- Hunt, P., Robertson, D., Bretherton, R., & Royle, M. C. (1982). The scoot on-line traffic signal optimisation technique. *Traffic Engineering and Control* 23(4).
- Jiang, C.-Y., Hu, X.-M., & Chen, W.-N. (2021). An urban traffic signal control system based on traffic flow prediction. In *2021 13th international conference on advanced computational intelligence (ICACI)* (pp. 259–265). IEEE.
- Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
- Koonce, P., et al. (2008). *Traffic signal timing manual*. Technical report: United States. Federal Highway Administration.
- Lai, S., Xu, Z., Zhang, W., Liu, H., & Xiong, H. (2023). Large language models as traffic signal control agents: Capacity and opportunity. [arXiv:2312.16044](https://arxiv.org/abs/2312.16044)

- Le, T., Kovács, P., Walton, N., Vu, H. L., Andrew, L. L., & Hoogendoorn, S. S. (2015). Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, 58, 431–450.
- Li, L., Li, R., Peng, Y., Huang, C., & Yuan, J. (2022). Cooperative max-pressure enhanced traffic signal control. In *Proceedings of the 31st ACM international conference on information and knowledge management* (pp. 4173–4177).
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)* (pp. 2575–2582). IEEE.
- Lowrie, P. (1990). Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic.
- Meng, L., Wen, M., Le, C., Li, X., Xing, D., Zhang, W., Wen, Y., Zhang, H., Wang, J., Yang, Y., et al. (2023). Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2), 233–248.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.
- Oroojlooy, A., Nazari, M., Hajinezhad, D., & Silva, J. (2020). Attendlight: Universal attention-based reinforcement learning model for traffic signal control. *Advances in Neural Information Processing Systems*, 33, 4079–4090.
- Sun, X., & Yin, Y. (2018). A simulation study on max pressure control of signalized intersections. *Transportation Research Record*, 2672(18), 117–127.
- Van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30).
- Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36, 177–195.
- Vaswani, A. (2017). Attention is all you need. In *Advances in neural information processing systems*.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. [arXiv:1710.10903](https://arxiv.org/abs/1710.10903)
- Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., & Li, Z. (2019). Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1290–1298).
- Wei, H., Xu, N., Zhang, H., Zheng, G., Zang, X., Chen, C., Zhang, W., Zhu, Y., Xu, K., & Li, Z. (2019). Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1913–1922).
- Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2496–2505).
- Wei, H., Zheng, G., Gayah, V., & Li, Z. (2021). Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(2), 12–18.
- Wu, Q., Li, M., Shen, J., Du, B., Zheng, H., & Wang, J. (2024). Offlinelight: An offline reinforcement learning model for traffic signal control. In *International conference on learning representations*.
- Wu, Q., Li, M., Shen, J., Lü, L., Du, B., & Zhang, K. (2023). Transformerlight: A novel sequence modeling based traffic signaling mechanism via gated transformer. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 2639–2647).
- Wu, Q., Zhang, L., Shen, J., Lü, L., Du, B., & Wu, J. (2021). Efficient pressure: Improving efficiency for signalized intersections. [arXiv:2112.02336](https://arxiv.org/abs/2112.02336)
- Xiong, Y., Zheng, G., Xu, K., & Li, Z. (2019). Learning traffic signal control from demonstrations. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2289–2292).
- Yang, S. (2023). Hierarchical graph multi-agent reinforcement learning for traffic signal control. *Information Sciences*, 634, 55–72.
- Yang, Z., Kong, Y., & Hsia, C.-H. (2024). Derlight: A deep reinforcement learning traffic light control algorithm with dual experience replay. *Journal of Internet Technology*, 25(1), 79–86.
- Yazdani, M., Sarvi, M., Bagloee, S. A., Nassir, N., Price, J., & Parineh, H. (2023). Intelligent vehicle pedestrian light (ivpl): A deep reinforcement learning approach for traffic signal control. *Transportation Research Part C: Emerging Technologies*, 149, Article 103991.
- Zang, X., Yao, H., Zheng, G., Xu, N., Xu, K., & Li, Z. (2020). Metalight: Value-based meta-reinforcement learning for traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, pp. 1153–1160).

- Zhang, L., & Deng, J. (2023). Data might be enough: Bridge real-world traffic signal control using offline reinforcement learning. [arXiv:2303.10828](https://arxiv.org/abs/2303.10828)
- Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., & Li, Z. (2019). Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference* (pp. 3620–3624).
- Zhang, L., Xie, S., & Deng, J. (2023). Leveraging queue length and attention mechanisms for enhanced traffic signal control optimization. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 141–156). Springer.
- Zhao, C., Hu, X., & Wang, G. (2020). Pdlight: A deep reinforcement learning traffic light control algorithm with pressure and dynamic light duration. [arXiv:2009.13711](https://arxiv.org/abs/2009.13711)
- Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., Li, Y., Xu, K., & Li, Z. (2019). Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1963–1972).
- Zheng, G., Zang, X., Xu, N., Wei, H., Yu, Z., Gayah, V., Xu, K., & Li, Z. (2019). Diagnosing reinforcement learning for traffic signal control. [arXiv:1905.04716](https://arxiv.org/abs/1905.04716)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.