

FutureLight: An Efficient Future Traffic Data-Driven Reinforcement Learning Framework for Traffic Signal Controls

Zizhuo Xu

HKUST, Hong Kong SAR, China
zxucj@connect.ust.hk

Haolun Ma

DSA Thrust, HKUST(GZ), China
hma987@connect.hkust-gz.edu.cn

Lei Li

DSA Thrust, HKUST(GZ), China
thorli@ust.hk

Zhiyuan Wang

DSA Thrust, HKUST(GZ), China
zwang895@connect.hkust-gz.edu.cn

Yunjie Huang

DSA Thrust, HKUST(GZ), China
yhuang863@connect.hkust-gz.edu.cn

Xiaofang Zhou

HKUST, Hong Kong SAR, China
zxf@cse.ust.hk

ABSTRACT

Reinforcement learning (RL) has emerged as a promising approach for Traffic Signal Control (TSC), enabling intersections to learn control policies under dynamic traffic conditions. However, existing RL-based methods only rely on current traffic states and expected rewards, without leveraging future traffic conditions for better decision making. To address this limitation, we propose *FutureLight*, the first RL-based TSC framework that leverages future route data. Specifically, we design *FutureLight-RouteSys*, a macroscopic, signal-aware, and lane-level simulator for efficient and accurate estimation of future traffic conditions. The predicted future information is then encoded by *FutureLight-Encoder* and integrated into *FutureLight-DQN* through state augmentation, reward shaping, and hybrid value estimation, which combines simulated near-future rewards with bootstrapped returns. Finally, we introduce pruning techniques to avoid redundant computation and improve training efficiency. Experimental results show that *FutureLight* consistently improves TSC performance while achieving more than 30× faster training.

PVLDB Reference Format:

Zizhuo Xu, Haolun Ma, Lei Li, Zhiyuan Wang, Yunjie Huang, and Xiaofang Zhou. FutureLight: An Efficient Future Traffic Data-Driven Reinforcement Learning Framework for Traffic Signal Controls. PVLDB, 19(9): 2344 - 2357, 2026.
doi:10.14778/3819518.3819555

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at <https://github.com/Allxz0523/FutureLight>.

1 INTRODUCTION

Background. *Traffic Signal Control* [20, 33] is a fundamental component of intelligent transportation systems, designed to manage urban traffic flow efficiently in complex and dynamic environments. Traffic management systems [11] rely on sensing and communication infrastructures, such as IoT-enabled platforms, to continuously

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 19, No. 9 ISSN 2150-8097.
doi:10.14778/3819518.3819555

* Lei Li is the corresponding author.

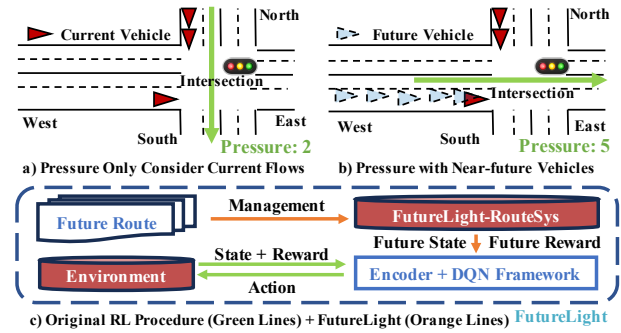


Figure 1: Procedure to Simulate Routes' Passing Sequence

collect traffic information from the physical environment. Recent advances have increasingly focused on reinforcement learning (RL)-based approaches [44, 60], where each signal is modeled as an agent that adapts its control policy under varying traffic conditions. In green lines of Figure 1-c), RL-based methods rely on microscopic simulators to provide traffic states for signal phase selection at each decision step. They aim to enhance the overall traffic efficiency by minimizing performance metrics such as the average travel time.

Limitation. During decision making, each RL-based agent only considers the current traffic state and optimizes the expected cumulative reward. However, they ignore future traffic flow, which could help reduce average travel time. Recent traffic prediction models [16, 18] have shown strong performance in short-term traffic forecasting. However, these predictions are aggregated over coarse time intervals, such as average flow or speed in the next few minutes, making them difficult to integrate into RL-based control frameworks that require fine-grained, time-specific traffic information.

Example 1.1 (Future Traffic Guidance). In Figure 1-a), a microscopic environment such as *CityFlow* [59] provides only the current state to the agent: two vehicles (red) are waiting from N-S, and one is waiting from W-E. The agent therefore sets NS to green, as releasing more pressure leads to a higher reward during training. In contrast, Figure 1-b) shows that if four additional vehicles (blue) are expected to arrive from WE in the near future (e.g., within 30 seconds). Therefore, setting WE to green becomes the better choice.

Motivations. In real life, navigation data, which contains planned road segments and departure times, reveals the near-future travel intentions of vehicles that have not yet reached the current intersection but are expected to do so. For example, an area that is rarely

congested may become heavily congested when a large event (e.g., a concert) attracts many incoming vehicles. Models based only on historical data cannot predict such situations, whereas navigation data can [55]. Such future route data is increasingly available from GPS-based services [52], urban electric vehicle monitoring platforms [26], and other mobility systems such as maritime vessels [24] and warehouse AGVs [50]. Thus, we propose *FutureLight*, the first RL-based TSC framework that integrates fine-grained, time-specific future traffic information to improve intersection efficiency, but achieving this goal is non-trivial due to the following challenges:

Challenge 1: Accurately and efficiently transform future route data into queryable lane-level future traffic states for decision making. Current traffic prediction methods are efficient but produce coarse-grained outputs, making them unsuitable for RL-based TSC, which requires fine-grained traffic information. Given future route data, microscopic simulators such as *SUMO* [1] can generate accurate near-future traffic conditions, but their high computational cost makes them impractical for real-time decision making. In contrast, macroscopic traffic models [54, 55] can efficiently estimate future traffic states, but they operate at the road level and do not model traffic signals, limiting their use in TSC. **Accordingly, we propose *FutureLight-RouteSys*, a macroscopic, signal-aware, and lane-level traffic simulation algorithm for deriving future traffic states from future route data.** Building on macroscopic traffic modeling principles, we incorporate signal waiting time retrieval and lane-level future traffic flow estimation. Specifically, we use a priority queue with route labels to maintain the traveling order of vehicles. When a vehicle enters a lane, a ML model estimates its traversal time from static and dynamic features (e.g., lane length and current traffic flow). This design enables accurate and efficient lane-level estimation of future traffic states.

Challenge 2: Frequently updated traffic signals trigger time-consuming re-simulations. This problem can be viewed as incremental maintenance of derived future traffic states, where recomputing the entire simulation after each update is inefficient (e.g., a round of RL training call macroscopic simulation 120 times). During training, traffic signal phases may change dynamically, making previously estimated near-future traffic information outdated. Recomputing the entire simulation is costly and redundant. For example, in Section 6.7.1, re-simulation in New York takes 48 min, whereas our method brings it down to 0.6 min. **To address this issue, we introduce two pruning techniques.** The *no repetition* technique identifies unaffected route labels and continues the simulation from them instead of restarting from the beginning. The *early termination* technique stops the macroscopic simulation once the required near-future information is obtained to avoid unnecessary computation. To further improve RL training efficiency, we replace the microscopic RL environment with a macroscopic one. Instead of microscopic simulator, we use the lane-level traffic flows generated by *FutureLight-RouteSys* for control. This replacement preserves the accuracy of future traffic flow estimation, maintains input equivalence, and enables faster and more stable RL training.

Challenge 3: Effectively incorporation of action dependent near future traffic information into RL-based TSC. Existing RL frameworks are designed around current traffic states and immediate rewards, without exploiting near-future traffic information. A

straightforward solution is to concatenate near-future information with the current state, but the value of near-future information cannot be fully utilized during training. In addition, the near-future traffic information is not directly available in advance, but depends on the selected control action. Therefore, the RL framework must be redesigned to exploit future information while preserving training stability and practicality. To address this challenge, we propose a **future-aware RL framework that leverages predicted traffic information.** The predicted traffic flow is incorporated into the framework through two components: (i) *FutureLight-Encoder* augments current observations with embedded near-future lane-level traffic distributions. (ii) *Future-aware reward* incorporates predicted future congestion into the learning objective. Specifically, at each decision step, the agent first estimates a *peek*-action, and *FutureLight-RouteSys* then estimates the near-future traffic state under that *peek*-action. This future feedback reveals the short-term consequence of the selected *peek*-action and helps guide the final decision. These designs enable each intersection to anticipate upcoming traffic and make more coordinated signal control decisions.

Contributions. The main contributions of this work are as follows.

- We introduce future route data as a new spatiotemporal modality for RL-based TSC, and propose *FutureLight* to use it effectively;
- We design *FutureLight-RouteSys*, a signal-aware and lane-level traffic simulation algorithm that accurately and efficiently derives and queries future traffic states from the future route data;
- We design *no-repetition* and *early termination* pruning techniques for incremental updates, and use *FutureLight-RouteSys* as a light alternative to microscopic environments for faster RL-training;
- We propose multiple strategies to incorporate predicted future traffic into RL-based TSC, including state augmentation, reward shaping, and value estimation, thereby improving agents' overall decision quality by anticipating near-future traffic information;
- We conduct comprehensive experiments to validate the value of future traffic data and show that the proposed data management design improves overall training efficiency by more than 30 times.

2 RELATED WORK

2.1 Traffic Signal Control Algorithms

TSC has progressed since the early use of fixed-time plans [42]. Early adaptive systems like SCOOT [17] and SCATS [28] rely on the expert-defined rules, but cannot adapt to highly dynamic traffic conditions. The introduction of RL has significantly advanced real-time TSC, which can be categorized into two main lines: (1) **Core framework design**, which has established various RL paradigms, including value-based methods [2] [45], policy-based strategies [30], and actor-critic frameworks [4] [48]. The central focus has been on crafting state and reward representation that accurately reflects traffic dynamics. (2) **Traffic state representation design**, such as queue length [40] [49], vehicle counts [44] [51], and travel time [61]. CoLight [60] improves the state-of-the-art by modeling lane capacity dynamics, exploiting the number of running and waiting vehicles as inputs, but none of them considers future traffic data.

2.2 Traffic Signal Infrastructure Systems

Traffic light systems also rely on supporting sensing and communication infrastructure for real-time traffic monitoring. Such systems

enable wireless communication, centralized monitoring, and remote configuration of traffic signals, thereby providing the foundation for developing real-time traffic signal control algorithms [11] to better utilize traffic information for the real-time decision making.

2.3 Utilization of Future Traffic Data

Beyond TSC, existing traffic-related studies have also explored the use of navigation data by directly concatenating navigation-related features into model inputs, including traversed road segments [38], predicted travel times from travel time estimation [10], and aggregated traffic demand profiles [27, 58]. However, these features are typically coarse-grained and unsuitable for TSC. Future traffic data has also been utilized for path-rerouting [53] to reduce congestion.

2.4 Traffic Simulation Tools

(1) **Micro-level** traffic simulation (e.g., SUMO [1]) rely on physical models and assumptions (e.g., car following, acceleration, and lane change) to accurately simulate the behavior of vehicles at each time step. Therefore, they can effectively capture the impact of new road layouts and traffic signal policies, but they are too slow for real-time use. (2) **Macro-level** traffic simulation [13, 22, 41] uses simplified functions to coarsely model vehicle travel sequences. Although it is more efficient, it sacrifices simulation accuracy. We have proposed a road-level macroscopic traffic simulation algorithm [54–56] that efficiently and accurately estimates future traffic conditions from navigation data. However, it is not suitable for TSC tasks, because it does not model traffic signals and lacks the lane-level resolution.

3 PRELIMINARY

3.1 Basic Concepts of Road Network

Road Network: G . It is a graph $G = (V, E)$ structure, where V denotes a set of intersections, and E refers to a set of road segments. Each road consists of one or more lanes l , associated with a specific turn direction, such as going straight, turning left, or turning right. **Traffic Flow:** $f(u, v, l, t)$. It refers to the number of vehicles currently passing on the lane l of road segment $(u, v) \in E$ at the timestamp t . **Route:** r . Its passing road segments with temporal information from the origin u to the destination v is represented as $r = \langle (v_0, t_0), \dots, (v_k, t_k) \rangle$, where $v_0 = u, v_k = v, (v_i, v_{i+1}) \in E$, and t_i is the timestamp at $v_i, \forall 0 \leq i < k$. Travel time of route r departing at time t_0 is defined: $T(r, t_0) = \sum_{i=0}^{k-1} [c(v_i, v_{i+1}, t_i) + w(v_{i+1}, t_i + c(v_i, v_{i+1}, t_i))]$, where $c(v_i, v_{i+1}, t_i)$ denotes the travel time on the road (v_i, v_{i+1}) at time t_i , and $w(v_{i+1}, \cdot)$ represents the waiting time at v_{i+1} due to traffic signals, calculated based on route's arrival time on (v_i, v_{i+1}) . **Queue Length.** It represents the number of vehicles on an incoming lane that cannot pass through the intersection due to signal control or congestion, thereby quantifying the extent of accumulated delay. **Intersection.** As shown in Figure 2-a), the road network is represented as a collection of intersections, each with 12 incoming lanes. **Traffic Movement.** A vehicle's left turn, straight movement, or right turn through an intersection is defined as a traffic movement. In Figure 2-b), a four-way intersection contains 12 distinct movements. **Traffic Signal Phase:** p_i . Each consists of a set of non-conflicting traffic movements that are granted the right-of-way simultaneously,

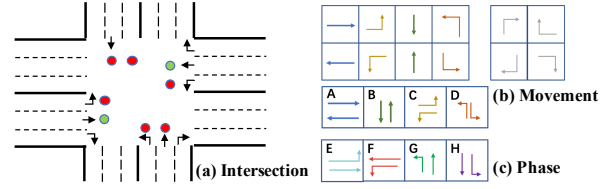


Figure 2: Definitions of traffic signal control.

while all other movements are held by red lights. In Figure 2-c), 8 distinct phases are utilized to coordinate the 12 traffic movements.

3.2 Basic Concepts of Traffic Signal Control

State: S and **Observation:** O . S is the global state of all traffic signal agents. At each time step t , each agent receives an observation $o \in O$, including a one-hot encoded phase representation of the currently active green phase and a flow representation that records the number of vehicles on each incoming lane of the intersection.

Near-Future Traffic State: S_f . For an intersection i and a given prediction horizon Δt , the near-future traffic state $S_{f,i}^{t+\Delta t}$ denotes the estimated lane-level traffic flow on all incoming lanes at time $t + \Delta t$. **Traffic Signal Action:** $\mathcal{A}$. It refers to the action space for each traffic signal agent, which includes all selectable green signal phases. At each time step t , an agent i selects an action $a_i^t \in \mathcal{A}_i$ from its available phase set. This action corresponds to one of the predefined green phases p that the intersection will maintain during the interval $[t, t + \Delta t]$. The action decides which phase will be activated for the next control period, which guiding traffic flow accordingly.

State Transition Function: $\mathcal{P}$. It shows how the environment evolves in response to signal control actions. At each time step t , given the current global state s_t and the joint action $\hat{a}_t = a_1 \times a_2 \times \dots \times a_N$ taken by all N traffic signal agents, the system transitions to a new state s_{t+1} according to the probability distribution $\mathcal{P}(s_{t+1} | s_t, \hat{a}_t)$. **Reward:** r . Each agent i receives an immediate reward r_i^t at t , determined by a combination of current and predicted traffic conditions on its incoming lanes. The reward is defined as follows: $r_i^t = -\alpha \cdot \text{current}_i^t - \beta \cdot \frac{1}{N} \sum_{n=1}^N \text{future}_i^{t+\Delta t}$, where current_i^t denotes the queue length at time t , and $\text{future}_i^{t+\Delta t}$ represents the predicted one at future step $t + \Delta t$. The coefficients α and β balance the contributions of present and future states. It enables the agent to make more informed decisions by considering the upcoming congestion. **Policy:** π . Each traffic signal agent follows a policy $\pi : O \rightarrow \mathcal{A}$, that maps observations to actions. At each time step t , agent i receives an observation $o_i^t \in O$ and selects an action $a_i^t \in \mathcal{A}$ according to its individual policy π_i . The objective of TSC is to learn a policy that maximizes the agent's expected cumulative reward over time. **Discount Factor:** γ . $\gamma \in [0, 1]$ determines the relative importance of future rewards. A higher γ places more emphasis on long-term outcomes, while a lower γ focuses on immediate rewards. The cumulative discounted return for the agent i at time t is defined as $R_i^t = \sum_{k=0}^{\infty} \gamma^k r_i^{t+k}$. This formulation ensures that rewards received sooner are weighted heavily, while future rewards are discounted.

3.3 Problem Definition

Definition 3.1 (Future Traffic-Driven TSC Problem). At each control step t , the agent at an intersection i receives an observation

$o_i^t = (S_{c,i}^t, S_{f,i}^{t+\Delta t})$, where $S_{c,i}^t$ denotes the current traffic state on its incoming lanes and $S_{f,i}^{t+\Delta t}$ denotes the near-future traffic state. The agent selects a signal phase $a_i^t \in \mathcal{A}_i$ from a policy π_i , and the objective is to learn the policy that maximizes the cumulative discounted reward $R_i^t = \sum_{k=0}^{\infty} \gamma^k r_i^{t+k}$. The final performance of the policy is evaluated by the average travel time (ATT) of all vehicles.

4 FUTURELIGHT TRAFFIC SIGNAL CONTROL

4.1 FutureLight Framework

The *FutureLight* framework enhances traffic signal control by incorporating future traffic information derived from route data. The overall workflow is illustrated in Figure 4 and summarized as follows. (a) **FutureLight-RouteSys**. In Figure 4-a), given the road network and route data with departure times, *FutureLight-RouteSys* efficiently estimates lane-level future traffic flows and travel times. The predicted future information is indexed by timestamp and stored, enabling efficient queries of near-future traffic states for any intersection. (b) **FutureLight-Encoder**. In Figure 4-b), for a target intersection (e.g., intersection B), the current traffic flows of neighboring intersections are incorporated to capture spatial dependencies. The predicted future traffic flow of the target intersection is then combined with the current flows of its neighbors to form an enhanced state representation. (c) **FutureLight-DQN**. In Figure 4-c), the encoded state is fed into a Q-value prediction module to evaluate all phases. The one with the highest Q-value is selected and applied to the environment producing the next observation. During simulation, the generated transitions are stored in a replay memory to train and update the Q-network. When the selected phase deviates from the predefined plan, the resulting traffic evolution also changes. Therefore, the future traffic estimation is updated to remain consistent with the executed traffic signal phase.

4.2 FutureLight-RouteSys

Existing macroscopic traffic simulation algorithms can estimate future traffic conditions. However, they cannot be directly applied to the TSC problem because they do not model traffic signal control and operate at the road level rather than the lane level. To address these limitations, we introduce *FutureLight-RouteSys*, an extension of the macroscopic *RouteSys* framework [54, 55] that explicitly incorporates traffic signal logic and supports lane-level simulation. This framework can efficiently and accurately estimate future traffic flow and travel time based on future route data and signal settings.

4.2.1 Route Label Initialization. To ensure vehicles' passing sequences are conducted in chronological order, we propose a data structure called as *route label*. It serves as the foundation of the simulation algorithm, guaranteeing that all vehicles preceding the current one have been processed before simulating the current one.

Route Label. $l(r) = \langle v_i, t_i \rangle$ is constructed for each route data r during the simulation algorithm, where $r.v_i$ indicates the currently arrived vertex of the route r , and $r.t_i$ shows its arriving time at v_i .

Example 4.1 (Route Labels Initialization). For ease of understanding, we use a road-level example instead of a lane-level one. In Figure 3, suppose there are two routes, $r_1 = \langle A, B, C \rangle$ and $r_2 = \langle D, B, C \rangle$, with departure times of 6:00 and 6:10. Their route

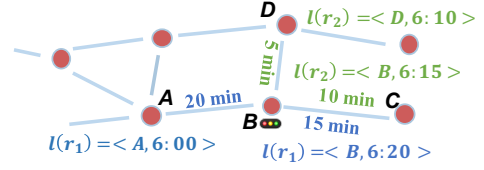


Figure 3: Procedure to Simulate Routes' Passing Sequence

labels are initialized as $l(r_1) = \langle A, 6:00 \rangle$ (blue) and $l(r_2) = \langle D, 6:10 \rangle$ (green). Suppose the travel times on $\langle A, B \rangle$ and $\langle D, B \rangle$ are estimated as 20 minutes and 5 minutes, respectively. During simulation, $l(r_1)$ and $l(r_2)$ are then updated to $\langle B, 6:20 \rangle$ and $\langle B, 6:15 \rangle$, respectively.

Simulation Procedure. It is illustrated as follows: (a) **Route labels initialization:** For each route, the current vertex v is initialized as its source $r.v_0$, and the corresponding arriving time t is the route's departure time $r.t_0$ initially. (b) **Iteration by earliest arrival time:** A priority queue L is used to maintain labels sorted by arrival time. In each iteration, label $l(r) = \langle v_i, t_i \rangle$ with the smallest arrival time is selected, and its travel time on the next road is estimated based on machine learning models (detailed in Section 4.2.2) with dynamically changing traffic flow and signal policies. (c) **Label and traffic flow update:** The route label is updated to the next vertex and time ($t \leftarrow t + T(v_i, v_{i+1}, t)$ and $l(r) \leftarrow \langle v_{i+1}, t \rangle$). Traffic flow is updated accordingly for the current lane when a route enters / leaves, and the count for the previous lane as one route leaves it. In addition, given the estimated arrival time at an intersection, the corresponding traffic signal state and remaining phase time can be queried in $O(1)$. If the signal is red, the expected waiting time is added to the travel time. (d) **Termination:** Once a route reaches its destination, it is removed from L . The simulation terminates when L becomes empty, indicating that all routes have been processed.

Example 4.2 (Simulation Example). The simulation shown in Figure 3 proceeds as follows: (1) $l(r_1)$ and $l(r_2)$ are initialized as $\langle A, 6:00 \rangle$ and $\langle D, 6:10 \rangle$. (2) $l(r_1)$ is selected first. The traffic flow on edge $\langle A, B \rangle$ increases by 1, and the estimated travel time, including the waiting time caused by the red signal, is 20 minutes. $l(r_1)$ is updated to $\langle B, 6:20 \rangle$ and reinserted into the priority queue L . (3) $l(r_2)$ becomes the top label in L . After the traffic flow on edge $\langle D, B \rangle$ is updated, the estimated travel time for r_2 on this edge is 5 minutes., and $l(r_2)$ is updated to $\langle B, 6:15 \rangle$ and re-inserted into L . (4) The same procedure is repeated until r_1 and r_2 reach the destinations. Their labels are then removed from L , and the simulation is terminated.

Complexity analysis. Assume the number of route data is $|R|$ with at most $|E|$ roads, time complexity of the simulation algorithm is $O(|E| \times |R| \times \log |R|)$, where $\log |R|$ is the priority queue operations.

4.2.2 Machine Learning Models. During simulation, when a vehicle enters a lane, its travel time on that lane is estimated using a learned machine learning model. The input features of the model are divided into two categories: (a) **Static features**, which remain constant over time, including the number of lanes, speed limit, and road length; (b) **Dynamic features**, which vary over time and reflect the traffic state when the vehicle enters the lane. These features include the traffic flow (i.e., the number of vehicles already in the lane), whether the vehicle will encounter a red signal when exiting the lane, and the vehicle's turning direction toward the next lane.

c) FutureLight-DQN (Section 4.4)

a) FutureLight-RouteSys (Section 4.2)

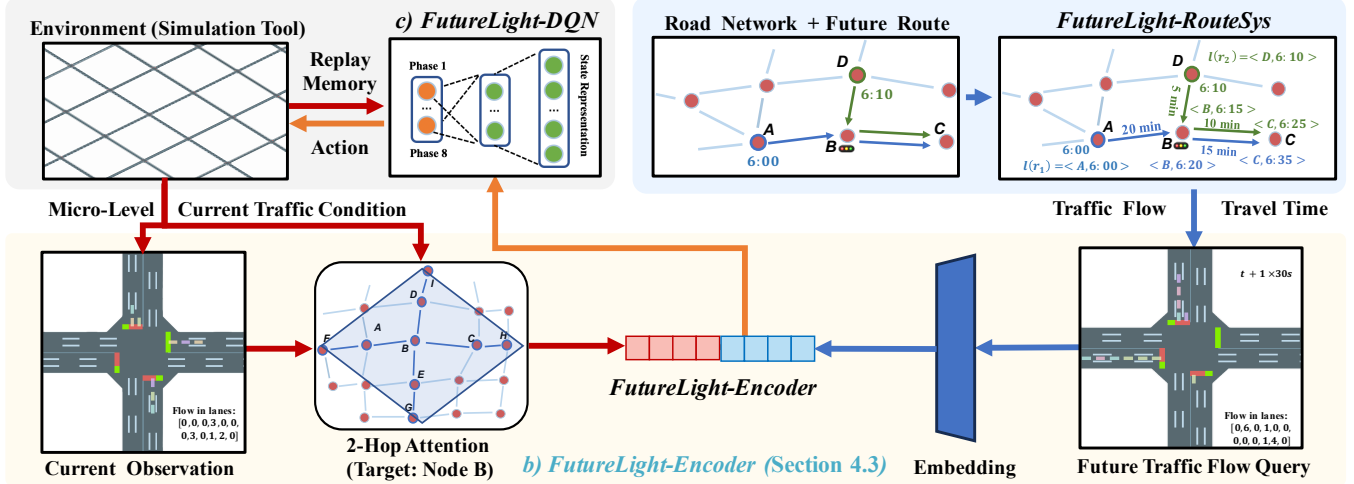


Figure 4: Overview architecture of FutureLight for the TSC problem: (a) FutureLight-RouteSys estimates lane-level future traffic conditions from route data, (b) FutureLight-Encoder integrates spatial dependencies with current and future flows to form enhanced state representations, and (c) FutureLight-DQN selects the optimal signal phase based on predicted Q-values.

We utilize CityFlow [59] to simulate traffic and generate training data, including road features and corresponding travel times on it. Models can be any mainstream machine learning approaches, categorized as follows: **(a) Tree-based methods.** Random Forest [6], LightGBM [19], XGBoost [9], CatBoost [35]. **(b) Linear models.** Linear [46], Ridge [15], and Lasso regression [36]. **(c) Neural networks.** MLP [39] and TabNet [3]. **(d) Ensemble methods.** Bagged [5], Stacked [47], and Weighted Ensemble [47]. **(e) Other methods.** KNN [34], SVM [32], and Naive Bayes [37]. Model selection depends on both predictive performance and interpretability. In our case, we choose tree-based methods because of the strong explainability.

4.2.3 Model Catching for Efficiency. Although machine learning models provide accurate predictions, their inference latency becomes a bottleneck under the large number of prediction requests during simulation, as discussed in Section 4.2. To reduce this overhead, we adopt a trade-off strategy that sacrifices little accuracy for faster inference. Given the small number of input features, we improve efficiency by caching model predictions, which requires around tens to hundreds of megabytes. Specifically, we precompute and store predicted travel times in a lookup table indexed by the corresponding input features. To reduce the space cost, these features are represented as integers or floating-point numbers with two decimal places. During simulation, model inference is replaced by table lookup, substantially improving the system responsiveness.

4.2.4 Route Records for Future Traffic Queries. During macroscopic traffic simulation algorithm, *route labels* record vehicle entry and exit events on lanes together with their timestamps. Based on these labels, the algorithm can dynamically maintain traffic flow changes on each lane over time. After simulation, the generated *route labels* and traffic flow records are reorganized by lane and timestamp to build *route records*, which describe how traffic flow changes as vehicles enter or leave a lane. Specifically, each *route record* is keyed by timestamp, and its values include the vehicle ID, the vehicle's entry or exit status, and the corresponding traffic flow on that lane. These *route records* serve as the basis for future traffic queries. Given

a specific lane and a future timestamp, the system can retrieve the corresponding *route records* on that lane to determine the number of vehicles will travel on the lane at the specific future timestamp.

Example 4.3 (Example of Future Traffic Queries). Continues to the Example 4.2, the generated *route labels* on each lane are listed in the following: 1. Road AB: [6:00 : r_1 , in, 1], [6:20 : r_1 , out, 0], 2. Road BD: [6:10 : r_2 , in, 1], [6:15 : r_2 , out, 0], 3. Road BC: [6:15 : r_2 , in, 1], [6:20 : r_1 , in, 2], [6:25 : r_2 , out, 1], [6:35 : r_1 , out, 0]. To query the future traffic flow on road BC at timestamp 6:22, a binary search is performed over its timestamps to locate the [6:20 : r_1 , in, 2], indicating that two vehicles will travel on road BC.

4.3 FutureLight-Encoder

In this section, we first describe the backbone feature encoder that embeds the intersections' current and surrounding information, then we present how to embed the future navigation information.

4.3.1 Traffic Embedding.

Phase-based Intersection Encoder: At each time step t , the raw observation of an intersection i , denoted by o_i^t , is represented as a 20-dimensional vector. Specifically, the current signal phase is one-hot encoded into an 8-dimensional vector, while the traffic flows on 12 incoming lanes (3 lanes $\times$ 4 directions) constitute the remaining 12 dimensions. Observation o_i^t is then encoded into a dense representation h_i by a MLP layer: $h_i = \text{Embed}(o_i^t) = \sigma(o_i W_e + b_e)$.

Graph Attention-based Cooperation: After encoding the local observation of each intersection into h_i , we enable communication among intersections through a graph attention mechanism. Each intersection is modeled as a node, while the spatial relationships between intersections (e.g., adjacency or shared roads) are represented as edges. This design allows the agent at each intersection to make decisions based not only on its local traffic conditions but also on the contextual influence of the neighboring intersections.

Spatial Dependencies among Intersections: To capture the spatial influence between intersections, we compute an interaction score for each pair (i, j) , where i is the target intersection and j is a

neighboring one. The score e_{ij} measures the importance from j for determining the signal control policy at i : $e_{ij} = (h_i W_t) \cdot (h_j W_s)^T$, where $W_t, W_s \in \mathbb{R}^{m \times n}$ are learnable projection matrices for the target and source embeddings, respectively. Since traffic influence is directional (e.g., information is different in upstream and downstream directions), e_{ij} is not necessarily symmetric, e.g., $e_{ij} \neq e_{ji}$.

Attention Normalization within Neighborhood: Given the interaction scores e_{ij} , we assign a normalized importance weight to each neighbor of i . This normalization ensures stable weighting across neighbors and prevents large raw scores from dominating the aggregation. Attention coefficient α_{ij} is defined as: $\alpha_{ij} = \frac{\exp(e_{ij}/\tau)}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik}/\tau)}$, where $\mathcal{N}_i$ denotes the set of neighbors of i , and τ is a temperature coefficient that controls the sharpness of the attention distribution.

Neighborhood Representation Update: After obtaining the attention coefficients, the representation of intersection i is updated by aggregating information from its neighbors weighted by the corresponding attention scores: $h'_i = \sigma(W_q \sum_{j \in \mathcal{N}_i} \alpha_{ij} (h_j W_c) + b_q)$, where $W_q, W_c \in \mathbb{R}^{m \times m}$ are trainable parameters, b_q is a bias term, and $\sigma(\cdot)$ denotes a nonlinear activation function (e.g., ReLU). The resulting embedding h'_i summarizes both the local condition and the contextual information from neighboring intersections, enabling cooperative and coordinated traffic signal control in the network.

Multi-head Attention Extension: To capture diverse spatial dependency patterns, we further extend the single-head attention mechanism to a multi-head version. Multiple attention heads operate in parallel, each with its own learnable parameters $\{W_t^h, W_s^h, W_c^h\}$. Their outputs are then averaged to form the final neighborhood representation: $\tilde{h}'_i = \sigma(W_q \cdot \frac{1}{H} \sum_{h=1}^H \sum_{j \in \mathcal{N}_i} \alpha_{ij}^h (h_j W_c^h) + b_q)$, where H is the number of attention heads. This design allows the model to jointly capture different spatial correlations (e.g., upstream congestion) and improves the robustness of the intersection cooperation.

4.3.2 Future Navigation Embedding. We first describe how to efficiently retrieve the future navigation data generated by *RouteSys*, and then explain how to use it to enhance feature representation.

Future Traffic Flow Query: As described in Section 4.2.4, future traffic queries are supported by the constructed *route records*. Given a lane and a future timestamp, the system retrieves the corresponding record on that lane to determine traffic flow at that timestamp.

Feature Enhancement by Future Information: To further improve the decision-making ability of the RL agent, the simulated future traffic flow can be queried and integrated into the encoded observation $\tilde{h}'_i$ at time t . When the RL agent updates the signal phase at each time step (e.g., $\Delta t = 30$ seconds), the near-future traffic flow over the next Δt can be queried in advance. Let $f_i^{t+\Delta t}$ denote the predicted future traffic information for intersection i at time $t + \Delta t$, obtained from *FutureLight-RouteSys*. The future feature $f_i^{t+\Delta t}$ is first transformed by a multi-layer perceptron (MLP) into a dense representation: $\hat{f}_i^{t+\Delta t} = \sigma(f_i^{t+\Delta t} W_f + b_f)$. Then, the future-enhanced representation $\tilde{h}_i^t$ is constructed by concatenating these encoded future features with the current encoded observation $\tilde{h}'_i$, which aggregates information from the neighbors: $\tilde{h}_i^t = \text{Concat}(\tilde{h}'_i, \hat{f}_i^{t+\Delta t})$.

In addition, if the originally defined signal phase plans are altered during the RL procedure, the corresponding future traffic flow

information can be efficiently re-simulated and updated, ensuring that $\tilde{h}_i^t$ remains consistent with the updated signal control policies.

4.4 FutureLight DQN

After obtaining the future-enhanced intersection representation $\tilde{h}_i^t$, each control agent uses the same DQN to make two sequential decisions. It first predicts a *peek-action* to query the corresponding near-future traffic state, and then predicts the optimal *final-action* based on the current state and the resulting future representation. It aims to approximate the optimal action-value function $Q^*(s, a)$, which represents the expected cumulative discounted reward when selecting action a under state s and the following optimal policy.

4.4.1 Q-value Estimation and Action Selection. Given the aggregated intersection representation $\tilde{h}_i^t$, which encodes both current and future traffic flow information, the control agent evaluates the utility of each candidate traffic signal phase through a Q-value prediction network. The Deep Q-Network (DQN) [29] predicts the input state $\tilde{h}_i^t$ to Q-values for all possible actions: $Q(\tilde{h}_i^t, a; \theta) = f_\theta(\tilde{h}_i^t)$, where $f_\theta(\cdot)$ is the Q-value estimation network parameterized by θ .

Action corresponding to the highest predicted Q-value is selected by the ϵ -greedy strategy defined in the following: $a_i^t =$ random action, with probability ϵ or $\arg \max_a Q(\tilde{h}_i^t, a; \theta)$, otherwise.

This procedure allows each intersection agent to make decisions based on both current and future aggregated state representation, supporting proactive rather than the purely reactive signal control.

4.4.2 Experience Replay and Target Network. To stabilize learning, transitions (s_t, a_t, r_t, s_{t+1}) are stored in a replay buffer $\mathcal{D}$. During training, mini-batches are randomly sampled from $\mathcal{D}$ to perform gradient-based updates, which helps reduce the temporal correlation between consecutive experiences. A separate target network with delayed parameters θ^- is used to compute target Q-values and reduce training oscillations: $y_t = r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-)$, where γ is the discount factor to balance immediate and future rewards.

4.4.3 Q-Learning Optimization Objective. Q-network parameters θ are learned by minimizing the mean squared temporal-difference (TD) error: $\mathcal{L}(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} [(y_t - Q(s_t, a_t; \theta))^2]$, where $Q(s_t, a_t; \theta)$ is the predicted Q-value and y_t is the target Q-value. This objective enables the network to iteratively approximate the optimal Q-function. By minimizing $\mathcal{L}(\theta)$ with stochastic gradient descent, DQN gradually aligns its Q-value estimates with the expected returns, thereby improving the control policy over time.

4.4.4 Integration with Future Guidance. Unlike conventional DQN-based signal control methods, the proposed framework incorporates *FutureLight-RouteSys* results as future navigation guidance. The future-enhanced state $\tilde{h}_i^t$ integrates both the current traffic observation and the predicted flow information in the next horizon, allowing the Q-network to anticipate the near-future impact of phase changes. Such forward-looking guidance mitigates myopic decisions and accelerates convergence during the training, thereby globally coordinated traffic signal control across all intersections.

5 EFFICIENT SPATIOTEMPORAL SIMULATION AND RL ENVIRONMENT OPTIMIZATION

5.1 Efficiency Enhancement of Simulation

Challenges. During RL training, the signal phase at each intersection is continuously updated according to its current state (described in Section 4.4.1). These dynamic phase changes can make the estimated future traffic flow inaccurate. For example, when the signal phase changes, some vehicles that were predicted to pass through the intersection may instead be stopped under the updated phase.

Ideas. The most straightforward solution is to rerun the macroscopic traffic simulation whenever a signal phase is updated during the RL training. However, this approach is inefficient, as it may lead to excessive computational overhead due to repeated calculations. To maintain efficiency, two key principles can be considered: (1) **No repetition:** For the time range with already determined signal phases, the simulation should reuse and manage the previously generated route labels from the *FutureLight-RouteSys*, rather than restarting the entire process each time, (2) **Early termination:** For the future time range with undecided signal phases, *FutureLight-RouteSys* terminates early to prevent unreliable future traffic flow.

5.1.1 No Repetition. By reviewing the route label update procedure in the *FutureLight-RouteSys* algorithm, when a vehicle enters a new road, its travel time is estimated based on both the static and dynamic characteristics of that road. Then, a waiting time is computed by determining whether the vehicle will be delayed by the current traffic signal. The sum of the estimated travel time and waiting time is then used to update the corresponding route label.

Assuming that the RL training procedure updates signal phases every t seconds, we define this interval as the *period of validity*, while the signal phases beyond this interval have not yet been determined. Accordingly, the route labels generated by the *FutureLight-RouteSys* algorithm can be categorized into reliable and unreliable route labels, as defined below. Reliable route labels correspond to vehicle trajectories that are fully determined within the current validity period, while unreliable route labels may be affected by future signal phase updates and therefore need to be re-simulated.

Reliable and Unreliable Route Labels. Given a route label $l(r) = \langle v_i, t_i \rangle$, we categorize it based on a *period of validity* t : **1. Reliable Route Label.** A route label is reliable if the vehicle can pass through the current road without being stopped by a red signal and $t_i \leq t$. **2. Unreliable Route Label.** Otherwise, the route label is unreliable.

To improve computational efficiency, we continue the simulation from reliable route labels instead of restarting from the initial state. In contrast, unreliable route labels will be re-simulated in subsequent periods when more signal phase information becomes available. Nevertheless, these unreliable route labels are still maintained for future traffic flow estimation, as they represent vehicles currently traveling behaviors on the corresponding road segments.

Example 5.1 (No Repetition). In Figure 5, three routes are initialized with labels: $l(r_1) = \langle A, 0 \rangle$, $l(r_2) = \langle D, 5 \rangle$, and $l(r_3) = \langle D, 10 \rangle$. The *period of validity* is set to 30s. During the signal phase 0s–30s at B, movements from A to B and D to B are allowed, while the movement from D through B to C is red. Based on *FutureLight-RouteSys*, the simulation proceeds as follows: (1) The estimated travel time of r_1 on AB is 20s, and the signal phase is green, so its route label is

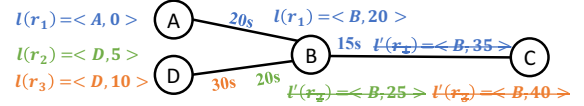


Figure 5: Example of No Repetition

updated to $l(r_1) = \langle B, 20 \rangle$, marked as reliable. (2) Travel time of r_2 on DB is 20s. Since the signal phase is red and waiting time is unknown, its route label is updated to $l(r_2) = \langle B, 25 \rangle$, but it's marked unreliable and excluded from the next simulation. (3) Travel time of r_3 on DB is 30s. With a departure time of 10s, the arrival time becomes 40s, which exceeds the validity period, so its label $\langle B, 40 \rangle$ is also unreliable. (4) The earliest reliable label is $\langle B, 20 \rangle$. The travel time on BC is 15s, leading to an arrival time of 35s, again beyond the validity period. (5) Unreliable labels will be re-simulated in the next period (30s–60s). If a label becomes reliable, it won't need re-estimation in future simulations. (6) In 30s–60s, the simulation starts with label $l(r_1) = \langle B, 20 \rangle$, $l(r_2) = \langle D, 5 \rangle$, and $l(r_3) = \langle D, 10 \rangle$.

5.1.2 Early Termination. Because signal phases may be updated during RL training, the future traffic flows estimated beyond the *period of validity* can no longer be trusted to guide action selection and may introduce noise. Therefore, the *FutureLight-RouteSys* algorithm can be terminated early, and the termination point is defined by the look-ahead window of $N \times t$. For time ranges beyond the *period of validity*, where signal phases remain undecided, the predefined phase plan is adopted as a placeholder. However, if the value of N is set too large, excessive noise may be introduced, which could degrade the performance rather than provided guidance.

5.2 Efficiency Enhancement on Environment

Motivation. In the previous sections, we proposed an efficient and accurate *FutureLight-RouteSys* algorithm. Since the gap between *FutureLight-RouteSys* and microscopic simulation environment is very small, it can also support traffic flow estimation with high efficiency and accuracy (as shown in Section 6.6 and 6.7). In addition, since the decision making in RL-based traffic signal control mainly depends on aggregated quantities (e.g., queue length or flow difference) rather than tracking the fine-grained information, the unbiased *FutureLight-RouteSys* is sufficient to preserve the Markovian property and ensure stable policy improvement. These observations form our motivation to explore whether traditional microscopic simulation environments can be replaced with the proposed *FutureLight-RouteSys* to accelerate the training efficiency.

To replace microscopic simulation environment in RL-based TSC, *FutureLight-RouteSys* must satisfy theoretical requirements that ensure information equivalence, adaptability, and Markov property.

5.2.1 Consistency with RL State and Reward Representation. As introduced in Section 3.2, the traffic signal control problem can be formulated as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where each intersection is modeled as an individual agent. At each decision step t , agent i observes $o_i^t \in \mathcal{O}_i$, selects an action $a_i^t \in \mathcal{A}_i$, and receives an immediate reward r_i^t , following the DQN-based learning procedure in the RL framework. To theoretically ensure the substitution in the RL training process, we analyze the consistency between the two simulation paradigms from following perspectives:

(1) **State and Observation Representation.** As summarized in Section 4.3, each intersection observation o_i^t is encoded into a low-dimensional vector h_i^t that represents the current signal phase and the lane-level traffic flow statistics. Because this design relies solely on aggregated lane-level quantities rather than microscopic vehicle trajectories, it provides an abstract yet informative representation that the proposed macroscopic simulator can efficiently reproduce.

(2) **Reward Representation.** Each agent receives a reward r_i^t at the end of each control interval to evaluate its phase selection. In existing RL-based frameworks, the reward is typically designed based on congestion- or pressure-related metrics, which can be written as: $r_i^t = f(\{q_\ell^t \mid \ell \in \mathcal{L}_i^{\text{in}}\}) = -\sum_{\ell \in \mathcal{L}_i^{\text{in}}} q_\ell^t$ or $-\sum_{(m,n)} (q_{i,m}^{\text{in},t} - q_{i,n}^{\text{out},t})$, q_ℓ^t denotes the number of vehicles on the incoming lane ℓ at time t , while $q_{i,m}^{\text{in},t}$ and $q_{i,n}^{\text{out},t}$ represent the aggregated inflow and outflow volumes of intersection i at time t , corresponding to its m -th incoming and n -th outgoing directions. It should be noted that the reward formulation above omits the incorporation of future information. As these reward functions depend solely on aggregated lane-level statistics, they are insensitive to the microscopic environment, such as individual vehicle positions or acceleration.

(3) **Requirement for Information Equivalence.** To serve as a substitute for microscopic environments, the proposed macroscopic simulator must provide consistent states and rewards. Formally, for each intersection i and lane ℓ , the macroscopic model should satisfy: $\tilde{q}_\ell^t \approx q_\ell^t$, $\tilde{r}_i^t = f(\tilde{q}_\ell^t) \approx r_i^t = f(q_\ell^t)$, where $\tilde{q}_\ell^t$ and $\tilde{r}_i^t$ denote the lane flow and reward estimated by the macroscopic simulator, and $f(\cdot)$ is the same reward function as in the original environment. If the discrepancies between $\tilde{q}_\ell^t$ and q_ℓ^t are bounded by a small error ε_s , then the resulting reward deviation ε_r can be remained negligible.

THEOREM 5.2 (RL CONSISTENCY UNDER INFORMATION EQUIVALENCE). *If the macroscopic simulator satisfies the information equivalence (in 5.2.1-(3)), then the RL process maintains theoretical validity. Specifically, the agent perceives consistent state and reward feedback, and the Markov Decision Process (MDP) formulation as well as the optimization behavior of the DQN-based framework remain preserved.*

5.2.2 Adaptive Re-simulation for Phase Changes. To efficiently respond to dynamic signal updates during RL training, the macroscopic simulator adopts an *adaptive re-simulation* strategy (in Section 5.1). Instead of rerunning the entire simulation whenever a phase changes, it reuses previously computed *reliable route labels* within the current *period of validity* and re-simulates only those *unreliable labels* whose future trajectories are affected by the new phase configuration. This mechanism, combined with the *early termination* strategy, enables the environment to maintain accurate and efficient flow statistics while avoiding redundant computations.

Therefore, *FutureLight-RouteSys* can replace the microscopic environment in standard DQN-based frameworks without significant changes to the learning formulation. We validate the accuracy and efficiency of this macroscopic RL environment in the Section 6.8.

6 EXPERIMENTS

6.1 Environment setup

6.1.1 Datasets. We evaluate our method on three widely used real-world TSC datasets, i.e., New York, Hangzhou, and Jinan [25, 44].

Table 1: Statistics Description of Route Data

Datasets/ Description	JiNan		HangZhou		New York	
	1	2	1	2	1	2
Num of routes	4365	4365	2983	2983	2212	2212
Average time	297s	291s	327s	326s	1245s	1419s
Average roads	4.44	4.32	4.65	4.37	15.99	16.53
Average length	2667m	2486m	3335m	3269m	4795m	4973m

Table 2: Average Travel Time Comparison (ATT) (Unit: s)

Methods	JiNan		HangZhou		New York	
	1	2	1	2	1	2
FixedTime [21]	369.6	331.4	497.1	437.8	1053.3	1036.8
MaxPressure [40]	279.5	276.3	325.3	310.4	654.2	648.3
MPLight [7]	294.4	283.3	355.4	333.3	498.0	449.6
CoLight [44]	271.2	266.7	310.7	310.2	420.1	416.0
E-MPLight [49]	268.4	262.6	310.5	302.5	654.1	637.8
E-CoLight [49]	267.6	260.9	302.4	296.6	478.2	473.7
E-MaxPressure [49]	267.0	260.5	318.2	302.2	632.1	634.9
E-PressLight [43]	303.1	298.4	376.9	364.0	1183.7	1281.8
A-MPLight [60]	265.9	260.8	309.8	292.8	427.2	423.8
A-CoLight [60]	270.4	260.0	304.5	288.4	416.9	412.2
A-MaxPressure [60]	265.9	260.8	314.2	299.0	499.7	493.5
LLMLight [23]	274.0	263.6	310.8	294.3	-	-
Macro-Sim [55]	265.1	259.6	299.2	290.0	423.3	413.6
Future-STGCN [57]	264.5	258.9	312.2	289.2	421.6	412.7
FutureLight	262.1	256.2	290.9	285.6	412.1	408.0

In addition, to assess the effect of route distributions, we construct three synthetic datasets on the same road networks, preserving the route counts while randomly sampling OD pairs and generating routes with Dijkstra’s algorithm [12]. The details are as follows:

1. Hangzhou Dataset. It consists of 16 intersections (4×4) in Hangzhou, China. The corresponding statistics are summarized in Table 1. Entries labeled as ‘1’ denote real-world datasets collected from roadside surveillance cameras, while entries labeled as ‘2’ denote synthetic datasets generated on the same road network.

2. JiNan Dataset. It includes 12 intersections (3×4) in Jinan, China. Similar to Hangzhou, labeled ‘1’ denotes the real-world datasets, while ‘2’ denotes the synthetic datasets, shown in Table 1.

3. New York Dataset. It includes 196 intersections (28×7) in Manhattan, New York City. Traffic demand is derived from New York City taxi trip records [31], where each trip provides an OD pair. The corresponding routes are generated using Dijkstra’s algorithm [12] to simulate realistic traffic. The statistics are shown in Table 1.

6.1.2 Compared Baselines. Baselines are categorized into traditional methods and RL-based methods. All RL-based models are trained for 80 rounds, with each round corresponding to a 60-minute simulation. Results in Table 2 shows the mean computed over the last ten testing episodes. A brief introduction of them is in below:

1. Traditional Methods. (1) **FixedTime** [21]: Static signal cycle with the fixed cycle time. (2) **MaxPressure** [40]: Selects phase maximizing pressure (queue difference) to stabilize the traffic flow. **2. Reinforcement Learning Methods.** (1) **MPLight** [7]: A decentralized RL framework combining MaxPressure with neural policy learning. (2) **CoLight** [44]: A cooperative RL framework using graph attention for intersection interaction. (3) **E-PressLight** [43]: A pressure-based RL framework minimizing queue imbalance. (4) **E-MPLight** [49]: Simplifies pressure computation to accelerate MPLight training. (5) **E-CoLight** [49]: Incorporates efficient

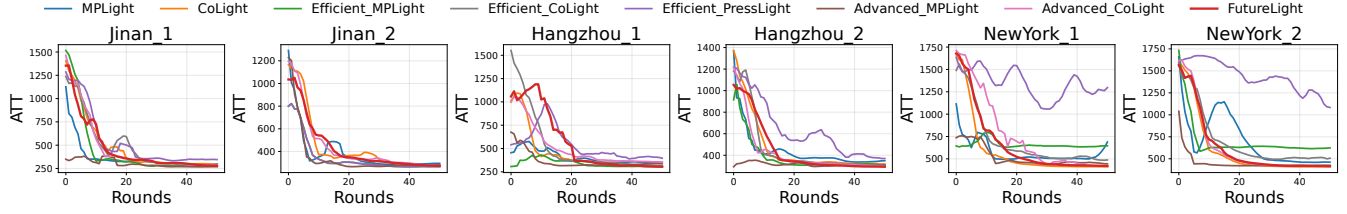


Figure 6: The Training Convergence of FutureLight (In 50 Rounds)

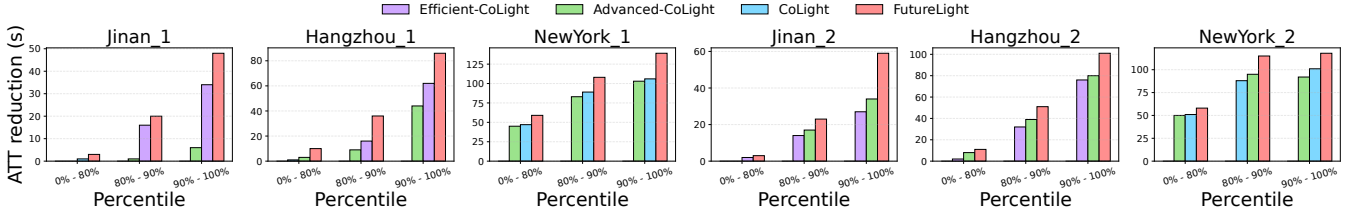


Figure 7: ATT Reduction (Compared with the Highest ATT) Across Congestion Percentiles. The Larger the Better.

Table 3: Ablation Study (Average Travel Time)

Variant	Jinan		Hangzhou		New York	
	1	2	1	2	1	2
Baseline	277.11	263.93	313.22	300.49	454.38	489.11
w/o Attention	263.61	258.68	296.00	289.23	427.01	426.86
w/o RouteSys	274.87	261.68	296.95	304.56	430.23	472.34
w/o F-Reward	264.68	258.37	293.87	289.49	416.29	416.86
FutureLight	262.10	256.24	290.85	285.64	412.15	407.97

Table 4: Analysis on Action Time and Look-ahead Horizon

Action Time	Ahead	Jinan		Hangzhou		New York	
		1	2	1	2	1	2
30s	15s	266.24	261.23	296.94	287.75	416.22	417.34
30s	30s	262.10	256.24	290.85	285.64	412.14	407.97
45s	15s	299.51	295.10	321.10	316.99	477.80	480.09
45s	30s	298.30	293.49	319.49	313.11	475.57	458.95
45s	45s	296.19	292.12	317.51	311.57	468.02	468.74
60s	30s	337.01	339.92	351.63	356.16	517.31	504.73
60s	45s	341.71	328.76	358.78	348.99	514.47	488.70
60s	60s	339.21	331.15	351.36	349.36	530.38	503.47

pressure design for faster cooperative learning. (6) **E-PressLight** [49]: Optimizes PressLight with lightweight reward computation. (7) **A-MPLight** [60]: Enhances MPLight with demand-aware state encoding. (8) **A-CoLight** [60]: Combines GAT-based cooperation with advanced representations. (9) **A-MaxPressure** [60]: Extends MaxPressure with advanced pressure-demand representations. (10) **LLMLight** [23]: A LLM framework that uses LLM reasoning for signal decision making, enhanced by the domain-specialized Light-GPT backbone. (10) **Macro-Sim** [55]: A macroscopic simulation algorithm for traffic prediction that does not consider traffic signals. (11) **Future-STGNN**: Near-future feature is generated by STGCN [57] using the previous four time intervals to predict the next. All the above “E-” stands for Efficient, and “A-” stands for Advanced.

6.1.3 Evaluation Metrics and Settings. We use: average travel time (ATT) [8, 14] as evaluation metrics for the simulated environment.

6.1.4 Environment. The macroscopic simulation algorithm is implemented in C++ and compiled with full optimizations. It is evaluated on a server with two Xeon Gold 8375C 2.9GHz CPUs (32 cores and 64 threads) and 2TB of memory. *FutureLight* and all baselines are implemented and evaluated on an NVIDIA RTX 4090 GPU with 24GB of memory. *FutureLight-RouteSys* is encapsulated as a C++

module, which is invoked by *FutureLight* whenever near-future information is needed, enabling interaction during the RL training.

6.1.5 Implementation Details. The key hyperparameters in experiments are summarized as follows: (1) **Action Time.** The signal control policy is updated every 30 seconds during RL training. (2) **Look-ahead Horizon.** The future prior uses a prediction horizon of future 30 seconds, so that the predicted traffic conditions correspond to the same time span as the current control phase. (3) **Prior Encoder.** The future prior is encoded by a two-layer MLP with hidden sizes of 64 and 32, providing sufficient representation capacity while keeping the model compact and stable. (4) **Prior Integration.** The predicted future inflow vector is concatenated with the original traffic state to form the final state representation.

6.2 Comparison Performance

This section analyzes the results in terms of average travel time and convergence between the proposed FutureLight and all baselines.

6.2.1 Comparison with Baselines on Average Travel Time. Table 2 describes the results of all baseline methods in terms of average travel time (ATT) across the JiNan, HangZhou, and New York datasets. The analysis is summarized as the following perspectives:

(1) **Overall Performance.** FutureLight consistently achieves the lowest ATT among all compared methods and outperforms both traditional and RL-based baselines. It shows that incorporating near-future traffic flow information derived from future route data can improve signal control decisions, reduce overall travel time, and thus alleviate network congestion. *LLMLight* is not evaluated on New York because its large scale and long episode duration make the LLM-based control prohibitively time- and memory-consuming.

(2) **Cross-Dataset Consistency.** The performance improvement of FutureLight is consistent across all three road networks. These results demonstrate the robustness and generalizability of FutureLight under diverse network structures and traffic conditions.

(3) **Interpretation and Insights.** The performance of *FutureLight* can be attributed to its integration of future traffic flow guidance into the RL framework. Leveraging macroscopic simulation-based predictions, it enables agents to anticipate the potential congestion and optimize phase transitions before the bottleneck occur.

Table 5: Robustness of Macroscopic Simulation under Different Future Route Availability (MAE/ RMSE/ MAPE %)

Percent	Task	Jinan 1	Hangzhou 1	NewYork 1
100%	Time	21.4 / 48.8 / 6.5%	5.8 / 15.5 / 2.3%	174.1 / 246.0 / 16.5%
	Flow	1.6 / 2.7 / 47.8%	0.7 / 1.4 / 35.1%	2.1 / 3.5 / 58.4%
70%	Time	20.4 / 47.9 / 6.6%	5.8 / 15.6 / 2.4%	177.0 / 248.1 / 16.9%
	Flow	2.3 / 3.7 / 51.9%	1.1 / 1.8 / 44.0%	2.0 / 3.5 / 57.7%
50%	Time	23.7 / 52.2 / 9.8%	6.5 / 17.8 / 2.6%	237.8 / 342.9 / 25.4%
	Flow	2.8 / 4.2 / 58.2%	1.4 / 2.1 / 51.6%	2.6 / 4.2 / 58.1%
30%	Time	22.6 / 51.3 / 9.4%	7.1 / 19.1 / 2.8%	297.5 / 411.3 / 29.6%
	Flow	3.6 / 5.1 / 69.2%	1.8 / 2.5 / 64.2%	3.7 / 5.8 / 66.7%

Table 6: Robustness of Macroscopic Simulation under Different Departure Time Perturbations (MAE/RMSE/MAPE%)

Noise	Task	Jinan 1	Hangzhou 1	NewYork 1
15s	Time	21.5 / 49.1 / 6.5%	6.4 / 16.7 / 2.5%	172.5 / 242.6 / 16.6%
	Flow	1.7 / 2.8 / 51.6%	0.9 / 1.5 / 40.5%	2.0 / 3.4 / 58.2%
30s	Time	23.1 / 51.0 / 7.0%	7.6 / 18.0 / 2.8%	174.5 / 245.8 / 16.6%
	Flow	1.8 / 2.8 / 52.5%	0.9 / 1.5 / 41.2%	2.1 / 3.5 / 59.9%
1 min	Time	26.8 / 54.2 / 8.1%	11.7 / 24.2 / 3.9%	174.3 / 241.5 / 17.3%
	Flow	1.8 / 2.9 / 54.5%	1.0 / 1.6 / 43.6%	2.0 / 3.5 / 59.2%
2 min	Time	34.7 / 61.4 / 10.8%	20.0 / 35.1 / 6.0%	182.0 / 246.4 / 19.0%
	Flow	2.0 / 3.0 / 58.5%	1.1 / 1.7 / 47.8%	2.0 / 3.4 / 60.4%
5 min	Time	46.5 / 72.1 / 14.4%	33.5 / 52.5 / 9.4%	190.7 / 253.3 / 20.4%
	Flow	2.2 / 3.3 / 65.2%	1.3 / 1.8 / 56.9%	2.0 / 3.4 / 62.0%

(4) **Analysis Under Different Congestion Levels.** To further understand the performance gains, we analyze ATT under different congestion levels. For each dataset, trips are divided by travel-time percentiles (*no congestion* 0–80%, *medium congestion* 80–90%, and *severe congestion* 90–100%). Figure 7 compares *FutureLight* with three strongest baselines. The y -axis reports the ATT reduction within each group, normalized by setting the highest ATT in that group to 0. Results show that under no congestion (0–80%), all methods have similar ATT. Traffic is close to free flow and signal optimization has limited impact. As they take 80% of the trips, the average improvement in Table 2 is marginal. In these highly congested scenarios (80–100%), *FutureLight* not only achieves the lowest ATT, but also delivers a substantially larger reduction than the baselines. For example, on NewYork-1, the ATT of *FutureLight* is around 140s lower than that of *Efficient-CoLight* in the 90–100% group. In addition, the congestion mitigation effect is more pronounced compared with static baselines. In the 90–100% group, the ATT is 2594.81s for *FixedTime* and 1558.5s for *MaxPressure*, while *FutureLight* reduces them to 840.42s. In summary, these results indicate the effectiveness of *FutureLight* under severe congestion.

6.2.2 Comparison on Convergence. We analyze the training convergence of *FutureLight* to validate its efficiency and stability during training. In Figure 6, *FutureLight* converges faster than all baselines, reaching stable performance after fewer training rounds, which can be attributed to the integration of near-future traffic flow, which provides each agent with knowledge of upcoming traffic dynamics.

6.3 Ablation Study

To evaluate the contribution of each component, we conduct an ablation study on Graph Attention, FutureLight-RouteSys, and the future-aware reward design. Since the future-aware reward is coupled with FutureLight-Encoder in FutureLight-RouteSys when near-future information is used, we do not evaluate the FutureLight-Encoder separately. In Table 3, baseline denotes the variant without

both Graph Attention and FutureLight-RouteSys. All components improve upon the baseline, while the full model achieves the best performance. Specifically, Graph Attention brings consistent but relatively moderate gains, indicating its effectiveness in capturing spatial dependencies among intersections. FutureLight-RouteSys leads to a larger ATT reduction, showing the benefit of action-dependent near-future traffic information. In addition, removing the future-aware reward also degrades performance, confirming the importance of using near-future information in reward design. In summary, the near-future information can substantially reduce the ATT by enabling more effective traffic prediction and control.

6.4 Sensitivity Analysis on Look-Ahead Horizon

We test two temporal hyperparameters: 1) **Action Time** is the duration for which the selected signal phase is maintained before the RL agent makes the next decision. 2) **Look-Ahead Horizon** is the future time span explored by the macroscopic simulation algorithm to provide future-aware information for the current decision. In our implementation, the look-ahead horizon is the early termination window, and we only consider settings where it does not exceed the action time. We do not evaluate shorter action times because real-world signal control must reserve sufficient yellow change, red-clearance, and pedestrian service intervals, making overly short action times less practical in the real-world scenario.

Table 4 reports the results with the following observations: 1) Under the 30s action time, the 30s horizon consistently outperforms the 15s horizon on all datasets, indicating that an overly short horizon cannot fully exploit useful future information. 2) When the action time increases, a longer horizon is not always beneficial. For example, under the 45s action time, the 15s horizon always performs worse, while in New York, the 30s horizon performs better than the 45s horizon. These results show that simply extending the look-ahead horizon does not necessarily improve the performance.

6.5 Sensitivity to Future Route Data Quality

6.5.1 Impact of Route Coverage. To evaluate the robustness of the proposed framework under partial future route data coverage, we conduct experiments with four coverage levels (*e.g.*, 100%, 70%, 50%, and 30%). The prediction accuracy of the macroscopic simulation is compared with ground-truth observations, as described in Table 5.

As future route data becomes less available, the prediction error of the macroscopic simulation algorithm increases consistently across all datasets. This effect is relatively mild on Hangzhou, but becomes more pronounced on Jinan and New York, where congestion is more severe. These results show that the accuracy of macroscopic simulation depends on the availability of future route data and becomes more sensitive under more congested conditions.

In addition, the RL-based signal control performance also decreases as future route data becomes less available, as shown in Table 7. However, with moderate future route data coverage (*e.g.*, 50%), the framework still performs effectively across different datasets, demonstrating its robustness under partial observability. This robustness mainly comes from two factors: 1) The macroscopic simulation algorithm still provides relatively stable traffic flow estimation under partial route information. 2) the RL agent uses both predicted future traffic states and current traffic observations, where

Table 7: RL Performance under Different Route Availability

Dataset	100%	70%	50%	30%	Best Base
Hangzhou-1	290.9±0.8	292.8±0.7	293.2±0.8	300.3±2.7	302.4±1.04
Jinan-1	262.1±0.7	263.1±1.0	264.7±1.2	266.2±1.6	265.8±0.00
NewYork-1	412.2±0.8	416.5±1.0	417.4±1.2	419.4±1.6	416.9±0.84

Table 8: RL Performance with Departure Time Perturbations

Dataset	15s	30s	1min	2min	5min
Hangzhou 1	292.1±0.8	292.8±0.9	293.2±1.1	295.3±1.5	296.6±1.3
Jinan 1	263.3±0.6	263.8±0.9	264.8±1.3	265.5±1.1	268.0±1.4
NewYork 1	415.4±3.4	417.4±2.4	420.1±2.2	423.8±8.3	426.3±4.8

the latter provide complementary information and help maintain performance when future route data is incomplete or noisy. Overall, although higher future route data availability leads to better performance, the framework does not critically rely on full information.

6.5.2 Impact of Departure Time Deviation. To evaluate the robustness of the proposed framework under noisy departure times, we conduct additional experiments by perturbing only the departure times used by the macroscopic simulation with Gaussian noise, while keeping the CityFlow-generated ground truth unchanged. We consider five perturbation levels: 15s, 30s, 1min, 2min, and 5min. Table 6 summarizes the prediction accuracy of the macroscopic simulation. As the departure time perturbation increases, the prediction error of the macroscopic simulation generally increases across all datasets. Hangzhou is the most sensitive to departure time deviation, Jinan shows a moderate increase, and the larger, more spatially distributed New York network exhibits smoother error growth. Moreover, although traffic flow error increases with larger perturbations, it is more stable than the travel time error.

Table 8 reports the RL performance under different levels of departure time perturbation. The RL-based signal control performance degrades as the perturbation becomes larger, but the overall degradation remains limited across different datasets, which is consistent with the observations in Section 6.5.1. The main reason is that the RL agent relies on the future traffic flow prior rather than the simulated travel time, and the traffic flow estimation remains relatively stable even when the departure times are noisy. Moreover, the agent also uses current observations, including signal phases, queue lengths, and lane-level traffic states, which can compensate for errors in the prior information. Therefore, although departure time deviation reduces the accuracy of the macroscopic simulation, the proposed framework remains robust under such uncertainties.

6.6 Accuracy of Macroscopic Traffic Simulation

To evaluate the reliability of estimated future traffic information, we compare the traffic flow and travel time generated by *FutureLight-RouteSys* (Section 4.2) with ground-truth results obtained from environment *CityFlow* by widely used metrics MAE/RMSE/MAPE.

6.6.1 Accuracy on the Estimated Future Traffic Flow. During the simulation process in the *CityFlow* environment, we record the traffic flow on each lane together with the corresponding timestamps as ground truth. Then, given the route data and a static signal phase plan, the proposed *FutureLight-RouteSys* algorithm can efficiently estimate future traffic flow information. Based on the query mechanism described in Section 4.2.4, we query the predicted lane flow at

each recorded timestamp and compare it with the ground truth. The row corresponding to 100% route data coverage in Table 5 illustrates the evaluation results on three datasets. Taking the results from the first route dataset in Hangzhou as an example, the MAE is less than one, indicating that the estimated future traffic flow generated by the proposed *FutureLight-RouteSys* algorithm is highly reliable.

6.6.2 Accuracy on The Future Travel Time. Similarly, after the simulation process in the *CityFlow* environment, each vehicle obtains its route travel time as the ground truth value. The proposed *FutureLight-RouteSys* algorithm can also estimate the corresponding travel time for each vehicle. The predicted and ground truth travel times are evaluated using the same metrics as in Section 6.6.1, and the results are summarized in the row corresponding to 100% route data coverage in Table 5. Taking the first route dataset in Hangzhou as an example, for routes with an average length of approximately 3 km, the MAE is less than ten seconds, which demonstrates the high accuracy and reliability of the *FutureLight-RouteSys* algorithm.

Similarly, as discussed in Section 6.6.1, the predicted travel time is also influenced by the highly severe and unusual congestion, where vehicles may need to wait through multiple traffic signal cycles. For instance, with a signal cycle of approximately 250 seconds, waiting for one more cycle can introduce an additional error of about 250 seconds, let alone the cases with multiple rounds of waiting. In addition, numerous acceleration and deceleration delays caused by such repeated stops can further deteriorate both prediction errors.

6.7 Efficiency of Macroscopic Traffic Simulation

6.7.1 Efficiency of Macroscopic Simulation. This section evaluates the runtime performance of *FutureLight-RouteSys* and the efficiency improvements from the optimization strategies. As discussed in Section 5.1, *No Repetition* and *Early Termination* are designed to reduce the computational overhead during RL training. Table 9 shows the average runtime per simulation for each method, including the baseline *Rerun Simulation*, *Early Termination*, *No Repetition*, and their combination. Specifically, when the agent queries near-future traffic flow (e.g., $t+30s$), *Rerun Simulation* executes the full simulation, *Early Termination* stops at $t+30s$, *No Repetition* runs from t to the end time, and the combination simulates only from t to $t+30s$.

In Table 9, the runtime of the *Rerun Simulation* reflects the baseline computational cost of the macroscopic algorithm, which mainly depends on the number of routes and road segments each route traverses. For instance, the runtime on the New York network is higher than on Hangzhou and Jinan, as each route in New York covers more roads and involves longer travel times, as shown in Table 1. Both optimization strategies effectively improve *Rerun Simulation* efficiency. Specifically, *Early Termination* outperforms *No Repetition*, possibly because the vehicle distribution in the simulation is left-skewed. By pruning later computations, *Early Termination* saves more time than *No Repetition*, which only eliminates redundant early-stage computations. Combining both strategies results in the lowest runtime, highlighting that near-future information from the *FutureLight-RouteSys* significantly enhances RL training efficiency.

6.7.2 Impact of Pruning on Training Efficiency. We evaluate the impact of pruning on the overall training efficiency of the RL framework using the per-round training time, including environment

Table 9: Comparison of Pruning during Simulation (Unit: s)

Efficiency Methods	JiNan		HangZhou		New York	
	1	2	1	2	1	2
Rerun Simulation	0.2288	0.2089	0.2358	0.1903	0.3045	0.2654
No Repetition	0.1068	0.1140	0.1213	0.1109	0.1921	0.1696
Early Termination	0.0024	0.0023	0.0024	0.0021	0.0047	0.0045
Combination	0.0021	0.0019	0.0020	0.0016	0.0042	0.0040

Table 10: Environment Comparison (Training Time/ ATT) (s)

Datasets/ Environments	JiNan		HangZhou		New York	
	1	2	1	2	1	2
CityFlow	26.7/ 279	23.3/ 276	29.9/ 325	28.5/ 310	275.7/ 654	270.1/ 648
RouteSys	0.98/ 267	0.97/ 261	1.3/ 301	1.39/ 306	14.7/ 512	15.6/ 512

Table 11: Comparison for Near-Future Traffic Estimation

Datasets/ Simulation Tools	JiNan		HangZhou		New York	
	1	2	1	2	1	2
CityFlow	0.3751	0.2910	0.2832	0.2961	1.8701	1.7438
FutureLight-RouteSys	0.0021	0.0019	0.0020	0.0016	0.0042	0.0040

rollout, sample construction, and network update. On the Jinan and Hangzhou, standard RL baselines require about 50s–80s per round, while *FutureLight* without pruning requires about 85s–120s. On the larger New York network, the baselines require about 400s–850s, while *FutureLight* without pruning techniques requires about 1300s.

Pruning substantially reduces this overhead. The per-round training time decreases from 115s to 51s on Jinan-1, from 123s to 58s on Hangzhou-1, and from 1301s to 973s on NewYork-1. As a result, the training cost of *FutureLight* becomes much closer to strong RL baselines, showing that pruning effectively controls the additional cost and keeps *FutureLight* practical even on large-scale networks.

6.8 Accuracy and Efficiency Validation of the Substituted Macroscopic RL Environment

In this section, we evaluate the computational cost of RL training and ATT after replacing CityFlow with the *FutureLight-RouteSys*.

Most RL-based TSC methods discussed in Section 6.1.2 rely on microscopic traffic features to make decisions. For example, *Advanced-CoLight* identifies which vehicles are truly waiting near the stop line by checking each vehicle’s exact position and determining whether its speed falls below a small threshold. Vehicles that are still moving in the upstream portion of the lane are intentionally excluded. Such calculations depend on precise vehicle features and cannot be reliably reconstructed by a macroscopic traffic simulator.

To evaluate whether the proposed macroscopic traffic simulation algorithm has the potential to replace the fine grained environment such as CityFlow, we check it using a simple static decision logic for phase selection. Specifically, whenever an intersection needs to choose the next signal phase, the algorithm predicts the traffic flow of each lane over a short future time slice (e.g., 30s) and selects the phase that serves the lane with the highest predicted demand. After choosing the phase, the algorithm predicts the flows for the next time window and repeats this procedure until a complete sequence of signal decisions is produced. Finally, we apply CityFlow to execute this entire signal plan to compute the resulting average travel time, enabling a fair microscopic evaluation of the decision.

6.8.1 Time Consumption and Average Travel Time of Replacement. Table 10 reports the time consumption and ATT for one training round (1 hour) using either CityFlow or *FutureLight-RouteSys* as

the environment. The row for CityFlow shows the results achieved by the *MaxPressure* baseline with CityFlow as the environment: 1. **ATT.** The results show that while our macroscopic simulator achieves comparable ATT, it offers significantly higher efficiency, particularly on larger networks like New York. However, as shown in Table 2, RL methods based on fine-grained microscopic features still outperform in terms of overall performance. This suggests that *FutureLight-RouteSys* as environment can maintain effective decision-making while significantly reducing time consumption during RL training. 2. **Time Consumption.** The macroscopic simulator with static decision logic is nearly twenty times faster than CityFlow. For instance, in the Jinan 1 network, CityFlow requires about thirty seconds for one round, whereas *FutureLight-RouteSys* takes around one second. This demonstrates significant potential for reducing simulation costs, especially for larger or more complex networks. However, the current macroscopic simulator is limited to static decision logic. Supporting fine-grained traffic features would require designing specialized mechanisms to align with specific baselines, which could further enhance efficiency. These extensions are beyond the scope of this study and are left for future exploration.

6.8.2 Estimating Future Traffic Flow by CityFlow. When near-future traffic information is required to enhance the agents’ decision-making, a straightforward approach is to use *CityFlow* to simulate the upcoming traffic state. As a micro-level traffic simulator, *CityFlow* can produce accurate near-future estimations. However, in Table 11, this accuracy comes at a significant computational cost. For instance, in New York, *FutureLight-RouteSys* produces a near-future estimate in around 0.004s, while *CityFlow* takes several seconds for the same prediction. This latency is incompatible with real-world TSC, where intersections must respond immediately to changing traffic conditions. Therefore, *FutureLight-RouteSys* is basis for supporting the near-future traffic estimation at the city scale.

7 CONCLUSION

Intelligent traffic signal control is crucial for the transportation system’s efficiency. In this work, we propose *FutureLight*, the first RL-TSC framework that leverages future navigation data, which is hard to incorporate due to its high volume, high dynamicity, and high cost to obtain. To convert the planned routes to future traffic data, we propose a macroscopic and signal-aware lane-level simulation framework. To support dynamic signal updates during training, we proposed several pruning strategies to further improve efficiency. Finally, by incorporating the future traffic information into state representation, reward design, and value estimation, *FutureLight* becomes a unified traffic signal control optimization framework with the higher decision quality and the higher training efficiency.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China under Grant #2024YFA1012700, NSFC #62572414, Hong Kong RGC grant #VPRDO24EG09, Guangdong-Hong Kong Technology Innovation Joint Funding Scheme #2024A0505040012, the HKUST – HKUST(GZ) Cross-campus Research Collaboration “1+1+1” Joint Funding Program (grant #C_2025_028), and JC STEM DSF Lab funded by Hong Kong Jockey Club Charities Trust.

REFERENCES

- [1] 2023. SUMO Official Website. <https://eclipse.dev/sumo/> Accessed: 2026-05-18.
- [2] Baher Abdulhai, Rob Pringle, and Grigoris J. Karakoulas. 2003. Reinforcement Learning for True Adaptive Traffic Signal Control. *Journal of Transportation Engineering* 129, 3 (2003), 278–285.
- [3] Sercan Ö Arik and Tomas Pfister. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 6679–6687.
- [4] Mohammad Aslani, Saadi Mesgari, Stefan Seipel, and Marco Wiering. 2018. Developing adaptive traffic signal control by actor–critic and direct exploration methods. *Proceedings of the Institution of Civil Engineers - Transport* 172 (03 2018), 289–298. <https://doi.org/10.1680/jtran.17.00085>
- [5] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24 (1996), 123–140.
- [6] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.
- [7] Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence (AAAI 2020 - 34th AAAI Conference on Artificial Intelligence)*. AAAI press, 3414–3421.
- [8] Mu-Yen Chen, Hsiu-Sen Chiang, and Kai-Jui Yang. 2022. Constructing Co-operative Intelligent Transport Systems for Travel Time Prediction With Deep Learning Approaches. *Trans. Intell. Transport. Sys.* 23, 9 (Sept. 2022), 16590–16599. <https://doi.org/10.1109/TITS.2022.3148269>
- [9] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.
- [10] Rui Dai, Shenkun Xu, Qian Gu, Chenguang Ji, and Kaikui Liu. 2020. Hybrid spatio-temporal graph convolutional network: Improving traffic flow prediction with navigation data. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*, 3074–3082.
- [11] Luiz Fernando Pinto De Oliveira, Leandro Tiago Manera, and Paulo Denis Garcez Da Luz. 2020. Development of a smart traffic light control system with real-time monitoring. *IEEE Internet of Things Journal* 8, 5 (2020), 3384–3393.
- [12] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.
- [13] Chunjiao Dong, Zhihua Xiong, Chunfu Shao, and Hui Zhang. 2015. A spatial-temporal-based state space approach for freeway network traffic flow modelling and prediction. *Transportmetrica A: Transport Science* 11, 7 (2015), 547–560.
- [14] Zekuan Fang, Fan Zhang, Ting Wang, Xiang Lian, and Mingsong Chen. 2022. MonitorLight: Reinforcement Learning-based Traffic Signal Control Using Mixed Pressure Monitoring. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (Atlanta, GA, USA) (CIKM '22)*. Association for Computing Machinery, New York, NY, USA, 478–487. <https://doi.org/10.1145/3511808.3557400>
- [15] Donald E. Farrar and Robert R. Glauber. 1967. Multicollinearity in regression analysis: the problem revisited. *Review of Economics and Statistics* 49, 1 (1967), 92–107.
- [16] Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. 2024. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proceedings of the VLDB Endowment* 17, 5 (2024), 1081–1090.
- [17] P B Hunt, D. I. Robertson, R. D. Bretherton, and Michael Royle. 1982. THE SCOOT ON-LINE TRAFFIC SIGNAL OPTIMISATION TECHNIQUE. *Traffic engineering and control* 23 (1982).
- [18] Weiwei Jiang and Jiayun Luo. 2022. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications* 207 (2022), 117921.
- [19] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
- [20] Peter Koonce et al. 2008. *Traffic signal timing manual*. Technical Report. United States. Federal Highway Administration.
- [21] Peter Koonce et al. 2008. *Traffic signal timing manual*. Technical Report. United States. Federal Highway Administration.
- [22] Apostolos Kotsialos, Markos Papageorgiou, Christina Diakaki, Yannis Pavlis, and Frans Middelham. 2002. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. *IEEE Transactions on intelligent transportation systems* 3, 4 (2002), 282–292.
- [23] Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. 2024. LLMLight: Large Language Models as Traffic Signal Control Agents. *arXiv:2312.16044 [cs.AI]*
- [24] Haoen Lei, Ruoxue Liu, Jiajing Chen, Haijiang Li, and Shuai Jia. 2026. Vessel traffic flow prediction through multi-scale spatiotemporal attention in dual-graph networks. *Transportation Research Part C: Emerging Technologies* 184 (2026), 105529.
- [25] Mingyuan Li, Jiahao Wang, Bo Du, Jun Shen, and Qiang Wu. 2025. FuzzyLight: A Robust Two-Stage Fuzzy Approach for Traffic Signal Control Works in Real Cities. In *KDD'25: The 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery.
- [26] Chenlei Liao, Jihao Deng, Xiqun Michael Chen, and Quan Yuan. 2026. Optimizing electric vehicle charging patterns and infrastructure for grid decarbonization. *Communications Sustainability* 1, 1 (2026), 43.
- [27] Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. 2017. Dynamic route planning with real-time traffic predictions. *Information Systems* 64 (2017), 258–265.
- [28] PR Lowrie. 1990. Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic. (1990).
- [29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.
- [30] Sajad Mousavi, Michael Schukat, Peter Corcoran, and Enda Howley. 2017. Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning. *IET Intelligent Transport Systems* 11 (04 2017). <https://doi.org/10.1049/iet-its.2017.0153>
- [31] New York City Taxi and Limousine Commission. 2025. TLC Trip Record Data. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml Accessed: 2025-10-10.
- [32] William S Noble. 2006. What is a support vector machine? *Nature biotechnology* 24, 12 (2006), 1565–1567.
- [33] Markos Papageorgiou, Christina Diakaki, Vaya Dinopoulou, Apostolos Kotsialos, and Yibing Wang. 2003. Review of road traffic control strategies. *Proc. IEEE* 91, 12 (2003), 2043–2067.
- [34] Leif E Peterson. 2009. K-nearest neighbor. *Scholarpedia* 4, 2 (2009), 1883.
- [35] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems* 31 (2018).
- [36] Jonas Ranstam and Jonathan A Cook. 2018. LASSO regression. *Journal of British Surgery* 105, 10 (2018), 1348–1348.
- [37] Irina Rish et al. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3. Citeseer, 41–46.
- [38] Qian Sun, Le Zhang, Huan Yu, Weijia Zhang, Yu Mei, and Hui Xiong. 2023. Hierarchical Reinforcement Learning for Dynamic Autonomous Vehicle Navigation at Intelligent Intersections. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Long Beach, CA, USA) (KDD '23)*. Association for Computing Machinery, New York, NY, USA, 4852–4861. <https://doi.org/10.1145/3580305.3599839>
- [39] Hind Taud and Jean-Francois Mas. 2018. Multilayer perceptron (MLP). *Geomatic approaches for modeling land change scenarios* (2018), 451–455.
- [40] Pravin Varaiya. 2013. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, 27–66.
- [41] Yibing Wang, Xianghua Yu, Jinqu Guo, Ioannis Papamichail, Markos Papageorgiou, Lihui Zhang, Simon Hu, Yongfu Li, and Jian Sun. 2022. Macroscopic traffic flow modelling of large-scale freeway networks with field data verification: State-of-the-art review, benchmarking framework, and case studies using METANET. *Transportation Research Part C: Emerging Technologies* 145 (2022), 103904.
- [42] Fo Vo Webster. 1958. *Traffic signal settings*. Technical Report.
- [43] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 1290–1298.
- [44] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. CoLight: Learning Network-level Cooperation for Traffic Signal Control (CIKM '19). Association for Computing Machinery, New York, NY, USA, 1913–1922. <https://doi.org/10.1145/3357384.3357902>
- [45] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control (KDD '18). Association for Computing Machinery, New York, NY, USA, 2496–2505. <https://doi.org/10.1145/3219819.3220096>
- [46] S Weisberg. 2005. Applied linear regression.
- [47] David H Wolpert. 1992. Stacked generalization. *Neural networks* 5, 2 (1992), 241–259.
- [48] Qiang Wu, Jianqing Wu, Jun Shen, Bo Du, Akbar Telikani, Mahdi Fahmideh, and Chao Liang. 2022. Distributed agent-based deep reinforcement learning for large scale traffic signal control. *Knowledge-Based Systems* (04 2022), 108304. <https://doi.org/10.1016/j.knsys.2022.108304>
- [49] Qiang Wu, Liang Zhang, Jun Shen, Linyuan Lü, Bo Du, and Jianqing Wu. 2021. Efficient Pressure: Improving efficiency for signalized intersections. *arXiv:2112.02336 [cs.LG]* <https://arxiv.org/abs/2112.02336> Accessed: 2026-05-18.
- [50] Ruizhong Wu, Mengxuan Zhang, Shuxin Wang, Frodo Kin Sun Chan, Yan Nei Law, and Lei Li. 2025. Continuous Lifelong Conflict-Aware AGV Routing with Kinematic Constraints. *Proceedings of the VLDB Endowment* 18, 7 (2025), 2254–2267.

- [51] Bingyu Xu, Yaowei Wang, Zhaozhi Wang, Huizhu Jia, and Zongqing Lu. 2021. Hierarchically and Cooperatively Learning Traffic Signal Control. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 1 (May 2021), 669–677. <https://doi.org/10.1609/aaai.v35i1.16147>
- [52] Yehong Xu, Lei Li, Mengxuan Zhang, Zizhuo Xu, and Xiaofang Zhou. 2023. Global routing optimization in road networks. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2524–2537.
- [53] Yehong Xu, Lei Li, Mengxuan Zhang, Zizhuo Xu, and Xiaofang Zhou. 2024. Global Optimal Travel Planning for Massive Travel Queries in Road Networks. *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [54] Zizhuo Xu, Lei Li, Mengxuan Zhang, Yehong Xu, and Xiaofang Zhou. 2024. Managing the Future: Route Planning Influence Evaluation in Transportation Systems. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 4558–4572.
- [55] Zizhuo Xu, Lei Li, Mengxuan Zhang, Yehong Xu, and Xiaofang Zhou. 2025. Ultra-Flexible, Explainable, and Scalable Traffic Prediction with Dynamic Future Routes. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*.
- [56] Zizhuo Xu, Doucheng Pan, Lei Li, Mengxuan Zhang, Yehong Xu, and Xiaofang Zhou. 2025. RouteSys: A Future Route Data Management System (Demo). In *Proceedings of the 33rd ACM International Conference on Advances in Geographic Information Systems*.
- [57] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.
- [58] Haitao Yuan, Guoliang Li, Zhifeng Bao, and Ling Feng. 2021. An effective joint prediction model for travel demands and traffic flows. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 348–359.
- [59] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference*. 3620–3624.
- [60] Liang Zhang, Qiang Wu, Jun Shen, Linyuan Lü, Bo Du, and Jianqing Wu. 2022. Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 26645–26654.
- [61] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. 2019. Learning Phase Competition for Traffic Signal Control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (Beijing, China) (CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1963–1972. <https://doi.org/10.1145/3357384.3357900>