applied sciences

MDPI

Article

A Comparative Study of Model Predictive Control and Deep Reinforcement Learning for Distributed Traffic Signal Control

Zhenzhen Hao *, Guorong Ma, Yongbin Cai, Xiaojiao Geng and Fuli Xiong

College of Artificial Intelligence and Robotics, Xi'an University of Architecture and Technology, Xi'an 710055, China; moama@xauat.edu.cn (G.M.); caiyongbin@xauat.edu.cn (Y.C.); xiaojiaogeng@xauat.edu.cn (X.G.); xiongfuli@xauat.edu.cn (F.X.)

* Correspondence: zhenzhenhao@xauat.edu.cn

Abstract

This paper presents a comparative study of two distributed traffic signal control approaches: a cell-transmission-model-based model predictive control (MPC) method and a deep reinforcement learning (DRL) method based on latent spatio-temporal network (LST-Net). In the MPC method, each intersection predicts local traffic evolution over the prediction horizon using the cell transmission model (CTM) and exchanges predicted boundary traffic information with neighboring controllers to enhance coordination among adjacent intersections. In the DRL method, temporal local traffic observations are processed to generate local signal decisions, while a compact latent representation of the network-level spatial traffic state is incorporated during centralized training to improve value estimation and policy learning. The controllers are evaluated on a 4×4 grid road network across multiple demand levels and traffic realizations, including variations in temporal demand profiles and spatial demand distributions, with Fixed-Time and Max-Pressure included as reference controllers. The comparison reveals clear differences in how controller performance changes as traffic demand increases. LST-Net generally provides favorable traffic efficiency under low-to-moderate demand, whereas MPC exhibits more consistent system-level performance as demand increases. The results further show that favorable in-network performance does not necessarily imply lower total time when depart delay is considered. Under unseen traffic scenarios, MPC maintains comparatively consistent performance, whereas LST-Net remains competitive and performs particularly well in some scenarios but exhibits greater variation across different traffic conditions. These findings indicate that no single control strategy is uniformly superior across all tested traffic conditions.

Keywords: traffic signal control; distributed control; model predictive control; deep reinforcement learning

Check for updates

Academic Editor: Iwona Grobelna

Received: 1 September 2026
Revised: 23 September 2026
Accepted: 29 September 2026
Published: 30 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Urban traffic congestion has become a persistent challenge in modern cities, leading to increased travel time, energy consumption, and environmental impacts. From a broader system perspective, congestion can also be understood in terms of the balance between time-varying mobility demand and the transport supply available to accommodate that demand [1]. This perspective also motivates the evaluation of traffic control strategies across different demand levels. Signalized intersections are critical locations where conflicting traffic streams compete for limited road capacity, and inefficient signal operation can further aggravate congestion. Consequently, traffic signal control plays an important role

in improving the operational efficiency of urban road networks by dynamically allocating right of way among competing traffic movements. At the network level, traffic signal control architectures can generally be categorized as centralized or distributed.

In a centralized architecture, signal decisions for multiple intersections are determined jointly using network-wide traffic information, allowing the interactions among intersections to be explicitly considered within a unified control problem. Such architectures have been investigated using network-level optimization and rolling-horizon control [2–4], as well as centralized reinforcement learning (RL) in which a single agent determines signal actions for multiple intersections [5,6]. In contrast, a distributed architecture decomposes the network-level control problem into multiple local controllers with coordination achieved through neighboring interactions or information exchange [7,8]. Such a structure can reduce the dimensionality of individual control problems and improve scalability for large traffic networks. Motivated by this advantage, the present study focuses on distributed traffic signal control and investigates two representative adaptive control approaches, namely MPC and DRL.

MPC-based distributed traffic signal control has been widely investigated to coordinate multiple intersections while retaining local decision making. Early studies formulate local signal control as a scheduling problem [7], while cell-based distributed methods decompose network-level signal timing optimization into intersection-level subproblems within a rolling-horizon framework [9]. A closely related study employs the CTM for local traffic prediction and exchanges predicted traffic and signal information with neighboring controllers for coordinating neighboring intersections [10–12]. Other distributed MPC formulations have focused on reducing communication and computational requirements through event-triggered mechanisms [13], explicitly accounting for uncertainty in traffic demand and turning ratios through stochastic prediction [14], and enabling decentralized intersection-level optimization using local traffic information [15]. More recently, distributed predictive control has also been combined with data-driven traffic prediction and multi-objective dynamic programming to improve real-time network signal optimization [16].

In parallel with MPC, RL has also become an important approach for adaptive traffic signal control [17]. For multi-intersection networks, multi-agent reinforcement learning (MARL) has been widely adopted to improve scalability by assigning local learning controllers to individual intersections and introducing coordination through reward design, shared information, or centralized training [18,19]. Representative studies have incorporated traffic-flow principles such as Max-Pressure into the learning process to improve network coordination [20], while other approaches explicitly model inter-intersection interactions through communication protocols or graph-based representations [19,21]. Large-scale decentralized learning has also been investigated to reduce the dimensionality of joint decision making in large-scale multi-intersection networks [18]. More recently, a variational autoencoder was employed to learn a compact latent representation of the global traffic state, enabling local traffic observations and compressed global information to be jointly exploited during training while retaining distributed signal decision making [22].

Although MPC-based and RL-based traffic signal control have both been extensively investigated, the two paradigms have largely developed along separate research lines, and systematic cross-paradigm comparisons remain relatively limited [23]. Some studies have directly compared the two approaches. For example, MPC and DRL have been evaluated for traffic signal control at a single intersection using the same microscopic simulation environment [24]. In parallel, another line of research has explored the integration of model-based prediction and learning to exploit their complementary characteristics. MPC and RL have been combined to improve traffic signal control under model uncertainties and

disturbances, with model-based control providing feasible or predictive control information and learning enhancing adaptation to uncertain traffic conditions [25,26]. More recently, learning has also been incorporated into distributed MPC-based traffic signal control to select among model-generated candidate signal plans [27]. These studies demonstrate growing interest in the relationship between predictive and learning-based control. However, studies that directly compare distributed MPC and distributed DRL under identical multi-intersection traffic conditions remain limited.

Motivated by this gap, this study compares two distributed traffic signal control implementations under various traffic demand levels. The MPC controller follows the coordinated CTM-based framework in [10], in which each intersection predicts local traffic evolution and exchanges predicted boundary information with neighboring controllers, with several adaptations introduced. The RL-based controller is developed from [22] by further incorporating temporal sequences of local traffic observations and latent spatial representations of the network-level traffic state to capture spatio-temporal traffic dynamics, leading to the LST-Net. Both controllers are implemented and evaluated in the Simulation of Urban MObility (SUMO) microscopic traffic simulation environment under the same road network, signal-phase structure, control interval, and simulation settings, with Fixed-Time and Max-Pressure included as reference controllers. The evaluation considers multiple random seeds and traffic realizations, as well as unseen scenarios with altered temporal and spatial demand patterns. This design allows the comparison to examine not only traffic efficiency across different demand levels, but also performance variability and consistency under changes in traffic realization and demand structure.

The remainder of this paper is organized as follows. Section 2 presents the distributed CTM-based MPC formulation and its coordination mechanisms. Section 3 introduces the LST-Net-based distributed traffic signal control method. Section 4 presents the comparative simulation studies across different demand levels, traffic realizations, and unseen traffic scenarios, followed by a discussion of the observed performance characteristics and practical limitations. Finally, Section 5 concludes the paper and discusses future research.

2. MPC-Based Distributed Traffic Signal Control

This section presents the distributed MPC method employed for traffic signal control. The road network is decomposed into a set of locally controlled intersections, with each intersection associated with an MPC controller. Instead of optimizing the entire network in a centralized manner, each controller predicts the traffic evolution within its local control region using CTM, while the interaction with adjacent intersections is represented through predicted boundary inflows and downstream receiving capacities. At each control instant, candidate signal actions are evaluated over a finite prediction horizon, and only the first control action is implemented before the optimization is repeated using newly observed traffic states.

2.1. Control Framework

Consider an urban road network consisting of a set of signalized intersections $\mathcal{I}$ as shown in Figure 1. Each intersection $\nu \in \mathcal{I}$ is controlled by an individual MPC controller and the local control region of each controller contains the incoming links of the corresponding intersection. Each link is partitioned into a sequence of connected cells. While the upstream portion of the link is represented by conventional CTM cells with aggregated traffic states, the terminal region adjacent to the signalized intersection is represented by a set of parallel movement-specific cells. These terminal cells distinguish traffic according to its intended movement, such as through/right-turn and left-turn movements.

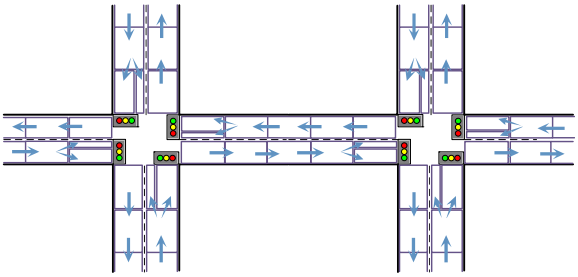


Figure 1. An illustration of connected intersections.

For the distributed MPC framework, three types of cells are considered: road cells, source cells, and sink cells. Road cells include both the conventional cells along a link and the movement-specific cells located near the stop line. Source cells represent traffic entering the local road network from upstream boundaries, whereas sink cells characterize the ability of downstream links to receive vehicles leaving the local control region. The source and sink cells mainly provide boundary information for interactions among distributed controllers. Let $\mathcal{C}_\nu^{\text{road}}$, $\mathcal{C}_\nu^{\text{src}}$, and $\mathcal{C}_\nu^{\text{snk}}$ denote the sets of road, source, and sink cells associated with intersection ν, respectively. Each local controller operates at a control interval of Δ_c. The time step of the prediction model in this study is $\Delta_m = 1$ s and is omitted in the following subsections for notational simplicity. The prediction horizon of each local controller is denoted by H.

2.2. Road-Cell Dynamics Prediction

The state of each cell $i \in \mathcal{C}_\nu^{\text{road}}$ is defined as the number of vehicles contained in the cell. At each control instant, traffic evolution is predicted over a finite horizon. Let $h = 0, 1, \ldots, H$ denote the prediction step, where $h = 0$ corresponds to the currently observed traffic state. The predicted state of cell i at prediction step h is denoted by n_i^h and its density is calculated as $\rho_i^h = n_i^h / L_i$ where L_i is the length of cell i. The dynamics of cell i follows the conservation equation of vehicles:

$$n_i^{h+1} = n_i^h + \sum_{r \in \mathcal{U}(i)} f_{ri}^h - \sum_{j \in \mathcal{D}(i)} f_{ij}^h, \tag{1}$$

where $\mathcal{U}(i)$ and $\mathcal{D}(i)$ denote the sets of upstream and downstream cells of cell i, respectively, and f_{ij}^h denotes the number of vehicles transferred from cell i to cell j during prediction step h.

The transfer flow f_{ij}^h is determined according to the sending function of cell i and the receiving function of cell j. Let v_f and w denote the free-flow speed and backward wave speed of the triangular fundamental diagram, respectively. The capacity Q_i and the jam density $\rho_{\text{jam},i}$ of cell i are usually proportional to the number of lanes contained in this cell. In this study, all cells share the same per-lane fundamental-diagram parameters. The sending function of cell i is defined as

$$S_i^h = \min\{v_f \rho_i^h, Q_i\}, \tag{2}$$

and the receiving function of cell i is given by

$$R_i^h = \min\{Q_i, w(\rho_{\text{jam},i} - \rho_i^h)\} \tag{3}$$

The transferred flow from cell i to downstream cell j is

$$f_{ij}^h = \min\left\{\beta_{ij} S_i^h, R_j^h\right\}, \tag{4}$$

where β_{ij} denotes the fraction of the sending demand of cell i directed toward downstream cell j and is usually determined by the routing probabilities. For a conventional one-to-one connection, as typically occurs between cells in the upstream portion of a link, all sending demand from cell i is directed to cell j, and thus $\beta_{ij} = 1$. For the case of the traffic flows from multiple cells merging to one cell, the available receiving capacity R_j^h for each upstream cell is further decided by the prescribed priority rule. Traffic flows usually diverge and merge near the intersection.

Let $\mathcal{M}_v \subset \mathcal{C}_v$ denote the set of movement-specific cells associated with signalized intersection v. For movement cell $m \in \mathcal{M}_v$, its outflow through the intersection is additionally governed by the traffic signal. Let ϕ^h denote the signal phase at prediction step h, and define

$$g_m^h = \begin{cases} 1, & \text{if movement cell } m \text{ is served by phase } \phi^h, \\ 0, & \text{otherwise,} \end{cases} \qquad m \in \mathcal{M}_v. \tag{5}$$

The actual transferred flow from movement cell m to downstream cell j is determined by

$$f_{mj}^h = \min\left\{ g_m^h \beta_{mj} S_m^h, R_j^h \right\}, \; m \in \mathcal{M}_v. \tag{6}$$

Therefore, no vehicle can leave the movement cell during a red or yellow phase.

2.3. Source-Cell and Sink-Cell Prediction

Since each MPC controller optimizes only the traffic dynamics associated with a local intersection, traffic interactions between neighboring control regions are represented by source and sink cells. A source cell represents traffic arriving from an upstream neighboring region and waiting to enter the first road cell of the local control region, whereas a sink cell represents the receiving condition provided by a downstream neighboring region. At each control instant, neighboring controllers exchange their predicted outflow and receiving trajectories, which are subsequently used to construct the source and sink cell conditions for the next MPC optimization.

Consider a source cell $i \in \mathcal{C}_v^{\text{src}}$ connected to the first road cell j of the local control region. Let r denote the corresponding upstream cell belonging to a neighboring control region. The state of the source cell i evolves as

$$n_i^{h+1} = n_i^h + f_{ri}^h - f_{ij}^h, \; i \in \mathcal{C}_v^{\text{src}} \tag{7}$$

where f_{ri}^h represents the predicted number of vehicles arriving at source cell i during prediction step h, and f_{ij}^h is the number of the transferred vehicles from source cell i to the first road cell j and is given by

$$f_{ij}^h = \min\left\{ n_i^h + f_{ri}^h, R_j^h \right\}, \; i \in \mathcal{C}_v^{\text{src}}. \tag{8}$$

Therefore, if the predicted arrival exceeds the amount that the first road cell can receive, the unaccepted vehicles remain in source cell i and contribute to n_i^{h+1}. Note that the source cells have an infinite storage capacity of vehicles.

At each control instant, the upstream neighboring controller provides a predicted outflow trajectory for the corresponding source cell. The trajectory received for the previous control instant is advanced by Δ_c prediction steps. Specifically, the expired entries of the trajectory are discarded, and its terminal value is held constant to complete the prediction horizon. Let $\bar{f}_{ri}^h$ denote the resulting shifted outflow at prediction step h. Since the reliability of the previously predicted trajectory generally decreases with prediction

distance, a piecewise-linear non-increasing confidence function $c(h) \in [0,1]$ is introduced. The inflow used for prediction is then given by

$$f_{ri}^{h} = \begin{cases} q_{ri}^{0}, & 0 \le h < \Delta_c, \\ c(h)\bar{f}_{ri}^{h} + [1-c(h)]q_{ri}^{0}, & \Delta_c \le h < H. \end{cases} \tag{9}$$

where q_{ri}^{0} denotes the currently measured inflow rate related to source cell i.

For each sink cell $i \in \mathcal{C}_{\nu}^{\mathrm{snk}}$, the downstream neighboring controller provides a predicted receiving-capacity trajectory rather than an outflow trajectory. The previously received trajectory is advanced by Δ_c prediction steps and completed by holding its terminal value constant, following the same procedure as that used for source cells. The receiving capacity R_i^h used over the prediction horizon is obtained using the same confidence-based correction as in Equation (9). The resulting receiving capacity constrains the traffic discharged from the local road network toward sink cell i:

$$\sum_{m \in \mathcal{U}(i)} f_{mi}^{h} \le R_i^{h}, \qquad i \in \mathcal{C}_{\nu}^{\mathrm{snk}}. \tag{10}$$

Through this mechanism, the predicted outflow of an upstream controller determines the arrival condition of the corresponding source cell, whereas the predicted receiving capability of a downstream controller determines the admissible inflow to the corresponding sink cell. Hence, neighboring MPC controllers exchange only boundary outflow and receiving capacity information, thereby retaining the distributed control framework.

2.4. Model Predictive Control Formulation

At each control instant, the controller determines whether the current signal phase should be maintained or changed. Let $\phi_{\nu,t}$ denote the current green phase and $g_{\nu,t}$ its elapsed green time of intersection ν at control instant t. Given the minimum and maximum green durations $g_{\phi_{\nu,t}}^{\min}$ and $g_{\phi_{\nu,t}}^{\max}$, respectively, the admissible action set at control instant t is defined as

$$\mathcal{A}_{\nu,t} = \begin{cases} \{\text{hold}\}, & g_{\nu,t} < g_{\phi_{\nu,t}}^{\min}, \\ \{\text{switch}\}, & g_{\nu,t} + \Delta_c > g_{\phi_{\nu,t}}^{\max}, \\ \{\text{hold}, \text{switch}\}, & \text{otherwise}. \end{cases} \tag{11}$$

For a hold action, the current phase remains active before the next control instant. For a switch action, the controller enters the prescribed yellow transition and subsequently activates the next green phase in the predefined phase sequence.

The objective of MPC is designed to simultaneously alleviate queue and promote throughput of the controlled intersection. The normalized queue measure $\overline{\mathcal{Q}}_{\nu}^{h}$ is used and defined as

$$\overline{\mathcal{Q}}_{\nu}^{h} = \frac{\sum_{i \in \mathcal{C}_{\nu}^{\mathrm{road}} \setminus \mathcal{M}_{\nu}} \left[n_i^h - n_i^{\mathrm{crit}} \right]_+ + \sum_{m \in \mathcal{M}_{\nu}} n_m^h + \sum_{i \in \mathcal{C}_{\nu}^{\mathrm{src}}} n_i^h}{\sum_{i \in \mathcal{C}_{\nu}^{\mathrm{road}}} \rho_{\mathrm{jam},i} L_i} \tag{12}$$

where $n_i^{\mathrm{crit}} = L_i Q_i / v_f$ is the critical vehicle number of cell i and $[x]_+ = \max(x, 0)$. For conventional road cells, queue is characterized by the vehicle accumulation exceeding the critical vehicle number. In contrast, vehicles stored in movement cells and accumulated in source cells are directly penalized.

Let Φ_{ν} denote the set of green phases of intersection ν, and let $\mathcal{M}_{\nu}(\phi)$ denote the set of movements served by phase ϕ. The normalized throughput is then calculated as

$$\overline{\mathcal{Y}}_{\nu}^{h} = \frac{\sum_{m\in\mathcal{M}_\nu}\sum_{j\in\mathcal{D}(m)} f_{mj}^{h}}{\max_{\phi\in\Phi_\nu}\sum_{m\in\mathcal{M}_\nu(\phi)} Q_m}, \tag{13}$$

where the numerator represents the total throughput of intersection ν at prediction step h and the denominator is the maximally allowable discharge capability of intersection ν.

The optimal action of each controller is determined by

$$a_{\nu,t}^{*} = \arg\min_{a\in\mathcal{A}_{\nu,t}} J_\nu(a) = \arg\min_{a\in\mathcal{A}_{\nu,t}} \frac{1}{H}\sum_{h=1}^{H}\left[\lambda_q \overline{\mathcal{Q}}_{\nu}^{h}(a) - \lambda_y \overline{\mathcal{Y}}_{\nu}^{h-1}(a)\right] \tag{14}$$

where λ_q and λ_y are weighting coefficients of the normalized queue measure $\overline{\mathcal{Q}}_{\nu}^{h}$ and throughput measure $\overline{\mathcal{Y}}_{\nu}^{h}$, respectively.

For each admissible action $a \in \mathcal{A}_{\nu,t}$, the traffic dynamics is predicted over H prediction steps as described above. After the first control decision, the subsequent signal actions are determined by a predefined continuation policy, denoted by π^{c}, rather than being explicitly optimized. This substantially reduces the number of signal sequences to be evaluated and thereby enables real-time implementation of the MPC. The continuation policy respects the signal timing constraints and, in general, favors phases with effective traffic demand and sufficient downstream receiving capacity to avoid unnecessary service to movements.

The proposed distributed MPC controller is implemented in a receding-horizon manner, as summarized in Algorithm 1. At each control instant, each intersection independently evaluates all admissible control actions using the local traffic model. The optimal control action is selected according to the objective function of Equation (14). After the first action of the optimal predicted trajectory is applied, the optimization is repeated using updated traffic information.

Algorithm 1 Distributed MPC at intersection ν.

Require: Current cell states $\{n_i^0\}$, current signal state, received source/sink predictions, prediction horizon H
Ensure: Optimal signal action $a_{\nu,t}^{*}$

```
Update the source-cell inflows f_ri^h and sink-cell receiving capacities R_i^h, h = 0, ..., H − 1
Determine the admissible control action set A_{ν,t}
for each a ∈ A_{ν,t} do
    Set a as the first signal action
    Generate subsequent signal actions using the continuation policy π^c
    Predict {f_ij^{h−1}, n_i^h}_{h=1}^{H} using the CTM-based prediction model
    Evaluate the finite-horizon objective J_ν(a)
end for
a*_{ν,t} ← arg min_{a∈A_{ν,t}} J_ν(a)
Exchange the predicted outflow and receiving trajectories with neighboring controllers
Apply a*_{ν,t} before next control instant
```

3. DRL-Based Distributed Traffic Signal Control

Different from the model-based MPC method, this section presents a DRL-based distributed traffic signal control method. Each signalized intersection is controlled independently according to its locally observed traffic conditions, while network-level traffic information is exploited during training to improve policy learning and coordination. The adopted method is based on LST-Net, which combines global traffic-state representation, temporal feature extraction, and actor–critic reinforcement learning within a centralized-training and decentralized-execution framework.

3.1. Problem Formulation

Consider a road network consisting of a set of signalized intersections $\mathcal{I}$. Each intersection $\nu \in \mathcal{I}$ is associated with an individual DRL controller that selects its traffic signal phase at discrete control instants. The control action at intersection ν is based on traffic information available within its local control region. Let $\mathbf{o}_{\nu,t}$ denote the local observation of intersection ν at control instant t. The observation consists of the current signal state and traffic conditions on the incoming lanes of controlled intersection and is defined as

$$\mathbf{o}_{\nu,t} = [\mathbf{p}_{\nu,t}, g_{\nu,t}, \boldsymbol{\rho}_{\nu,t}, \mathbf{q}_{\nu,t}], \tag{15}$$

where $\mathbf{p}_{\nu,t}$ is the one-hot representation of the current green phase, $g_{\nu,t}$ is a binary variable indicating whether the minimum green-time requirement has been satisfied, $\boldsymbol{\rho}_{\nu,t}$ denotes the normalized vehicle density of the incoming lanes, and $\mathbf{q}_{\nu,t}$ represents the normalized queue state of these lanes.

For each intersection, the action space consists of a predefined set of non-conflicting green phases:

$$\mathcal{A}_\nu = \{1, 2, \ldots, P_\nu\} \tag{16}$$

where P_ν denotes the number of available green phases at intersection ν. The action selected by the controller is $a_{\nu,t} \in \mathcal{A}_\nu$, with each action corresponding to one target green phase. As in MPC, the same minimal and maximal green-time constraints are also enforced in DRL. Based on the local observation, the policy is learned from interactions with the traffic environment, with the objective of improving traffic efficiency through an appropriately designed reward signal.

Two local reward formulations are considered in this study to investigate the influence of different learning objectives on traffic signal control. The first reward is based on vehicle waiting time. Let $W_{\nu,t}$ denote the total accumulated waiting time of vehicles within the local control region of intersection ν at decision instant t. Since signal decisions are made at intervals of Δ_c, the waiting-time-based reward is defined as

$$r_{\nu,t}^{\mathrm{WT}} = W_{\nu,t-\Delta_c} - W_{\nu,t}. \tag{17}$$

A positive reward is therefore obtained when the local accumulated waiting time decreases between two consecutive decision instants. Although the waiting-time-based reward directly reflects the delay experienced by vehicles already present in the road network, it does not account for the vehicle delay that is unable to enter the network under congested conditions.

To capture both in-network traffic efficiency and delayed departures, a new reward is constructed using local vehicle travel time and departure delay. Let $\mathcal{L}_\nu$ denote the set of controlled incoming lanes of intersection ν, and let $N_l(\tau)$ denote the number of vehicles on lane l at simulation time τ. Since the simulation step is 1 s, the local vehicle travel time accumulated during the control interval $(t - \Delta_c, t]$ is expressed as

$$T_{\nu,t} = \sum_{\tau=t-\Delta_c+1}^{t} \sum_{l \in \mathcal{L}_\nu} N_l(\tau). \tag{18}$$

For vehicles that are scheduled to depart but have not yet entered the road network, each pending vehicle is associated with the first controlled intersection along its predefined route. Let $\mathcal{P}_\nu(\tau)$ denote the set of such pending vehicles associated with intersection ν at time τ. The corresponding departure delay accumulated during the same control interval is

$$D_{\nu,t} = \sum_{\tau=t-\Delta_c+1}^{t} |\mathcal{P}_\nu(\tau)|. \tag{19}$$

The local vehicle-travel-time-and-departure-delay-based reward is then defined as

$$r_{v,t}^{\mathrm{TD}} = (T_{v,t-\Delta_c} + D_{v,t-\Delta_c}) - (T_{v,t} + D_{v,t}). \tag{20}$$

A positive reward is obtained when the combined local vehicle travel time and departure delay decreases from the preceding control interval to the current one. Compared with the waiting-time-based reward, this formulation additionally accounts for the vehicles waiting to enter the network and therefore provides a broader measure of traffic efficiency under congested conditions.

3.2. LST-Net Architecture

To capture both temporal traffic evolution and network-level traffic interactions, LST-Net combines a compact global traffic representation with local temporal observations. During training, the value network exploits both local and global traffic information, whereas the policy network relies only on locally observable information, thereby enabling distributed signal control during execution.

3.2.1. Global Traffic-State Encoding

To provide network-level information during training, the local observations of all intersections are organized into a gridded global traffic state $\mathbf{X}_t \in \mathbb{R}^{C\times N_r\times N_c}$, with C traffic-feature channels over an $N_r \times N_c$ grid road network. The gridded representation preserves the spatial arrangement of intersections and provides structured input for extracting global traffic features.

Since directly incorporating the high-dimensional global state into the control network would substantially increase the dimensionality of the learning problem, a convolutional variational autoencoder (Conv-VAE) is employed to obtain a compact latent representation. The encoder maps the global state into the parameters of a latent Gaussian distribution:

$$(\boldsymbol{\mu}_t, \log \boldsymbol{\sigma}_t^2) = E_\psi(\mathbf{X}_t) \tag{21}$$

and the latent traffic representation is obtained as

$$\mathbf{z}_t = \boldsymbol{\mu}_t + \boldsymbol{\sigma}_t \odot \boldsymbol{\epsilon}_t, \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{22}$$

The resulting latent vector $\mathbf{z}_t$ provides a compact representation of the network-wide traffic condition.

The Conv-VAE is trained by jointly minimizing the reconstruction error of the global traffic state and the KL divergence:

$$L_{\mathrm{VAE}} = \mathbb{E}\Big[\big\|\mathbf{X}_t - \hat{\mathbf{X}}_t\big\|_2^2 + 0.5\sum_{d=1}^{d_z}\Big(\mu_{t,d}^2 + \sigma_{t,d}^2 - 1 - \log\sigma_{t,d}^2\Big)\Big], \tag{23}$$

where $\hat{\mathbf{X}}_t$ denotes the reconstructed global traffic state, d_z is the dimension of the latent representation, and $\mu_{t,d}$ and $\sigma_{t,d}^2$ denote the mean and variance of the dth latent variable, respectively.

3.2.2. Temporal Traffic Representation

For each intersection v, a sliding window of length W is used to construct the local temporal observation sequence:

$$\mathbf{O}_{v,t} = [\mathbf{o}_{v,t-(W-1)\Delta_c}, \ldots, \mathbf{o}_{v,t-\Delta_c}, \mathbf{o}_{v,t}]. \tag{24}$$

For value estimation during training, each local observation is further augmented with the global latent representation as $\widetilde{\mathbf{o}}_{\nu,t} = [\mathbf{o}_{\nu,t}; \mathbf{z}_t]$, and the corresponding enhanced temporal sequence $\widetilde{\mathbf{O}}_{\nu,t}$ is constructed in the same manner. Thus, the policy network uses only local temporal observations, whereas the value network additionally exploits network-level traffic information.

Both temporal sequences are processed by the GRU–self-attention encoder. The GRU first extracts temporal traffic features:

$$\mathbf{H}^{\mathrm{G}}_{\nu,t} = \mathrm{GRU}(\mathbf{O}_{\nu,t}) \tag{25}$$

and self-attention is subsequently applied to capture dependencies among different time steps:

$$\mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V}. \tag{26}$$

A temporal pooling operation then converts the resulting sequence into a fixed-dimensional representation. In this way, the encoder combines the sequential modeling capability of the GRU with the ability of self-attention to emphasize temporally relevant traffic states.

The policy and value networks employ temporal encoders with the same general architecture but independent parameters. The policy network receives the local temporal sequence and outputs the action distribution $\pi_{\theta_\nu}(a_{\nu,t} \mid \mathbf{O}_{\nu,t})$, where θ_ν denotes the policy-network parameters. In contrast, the value network receives the enhanced temporal sequence containing both local observations and the global latent traffic representation and outputs the state-value estimate $V_{\omega_\nu}(\widetilde{\mathbf{O}}_{\nu,t})$, where ω_ν denotes the value-network parameters. This asymmetric information structure follows the centralized-training and decentralized-execution principle.

3.3. PPO-Based Training

The policy and value networks are trained using proximal policy optimization (PPO). For each intersection ν, transitions collected through interactions with the traffic environment are used to update the policy and estimate the value function. Since control actions are generated every Δ_c, the temporal-difference error is defined as

$$\delta_{\nu,t} = r_{\nu,t} + \gamma V_{\omega_\nu}(\widetilde{\mathbf{O}}_{\nu,t+\Delta_c}) - V_{\omega_\nu}(\widetilde{\mathbf{O}}_{\nu,t}) \tag{27}$$

where γ is the discount factor. The generalized advantage estimate (GAE) is then calculated as

$$\hat{A}_{\nu,t} = \textstyle\sum_{\ell=0}^{L_t-1} (\gamma\lambda)^{\ell} \delta_{\nu,t+\ell\Delta_c}, \tag{28}$$

where L_t denotes the number of remaining control steps from the current time t to the end of the episode, and λ is the GAE smoothing parameter.

The policy network is optimized using the clipped loss:

$$L_\pi = -\mathbb{E}\big[\min\big(\chi_{\nu,t}\hat{A}_{\nu,t}, \mathrm{clip}(\chi_{\nu,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{\nu,t}\big)\big], \tag{29}$$

where ϵ is the clipping parameter and $\chi_{\nu,t} = \pi_{\theta_\nu}(a_{\nu,t} \mid \mathbf{O}_{\nu,t}) / \pi_{\theta_\nu^{\mathrm{old}}}(a_{\nu,t} \mid \mathbf{O}_{\nu,t})$ is the probability ratio between the current and previous policies. An entropy bonus is additionally included during training to encourage policy exploration.

The value network is trained by minimizing the mean squared error between the estimated value and the temporal-difference target:

$$L_V = \mathbb{E}\Big[\big(V_{\omega_\nu}(\widetilde{\mathbf{O}}_{\nu,t}) - y_{\nu,t}\big)^2\Big], \tag{30}$$

where $y_{v,t} = r_{v,t} + \gamma V_{\omega_v}(\widetilde{\mathbf{O}}_{v,t+\Delta_c})$. The policy and value networks are optimized with independent parameters. Meanwhile, the Conv-VAE is updated using the auxiliary loss defined previously, so that global traffic-state compression, policy learning, and value estimation are jointly improved during training.

The overall training of LST-Net is summarized in Algorithm 2. During training, local observations from all intersections are first organized into the global gridded traffic state and compressed by the Conv-VAE. Each local policy selects a signal action using its local temporal observations, whereas the corresponding value network additionally exploits the global latent representation. The collected transitions are subsequently used to update the policy and value networks using PPO and to train the Conv-VAE. Once training is completed, each intersection can independently determine its signal phase without accessing the global network state, preserving the distributed control structure of LST-Net.

Algorithm 2 Training of LST-Net.

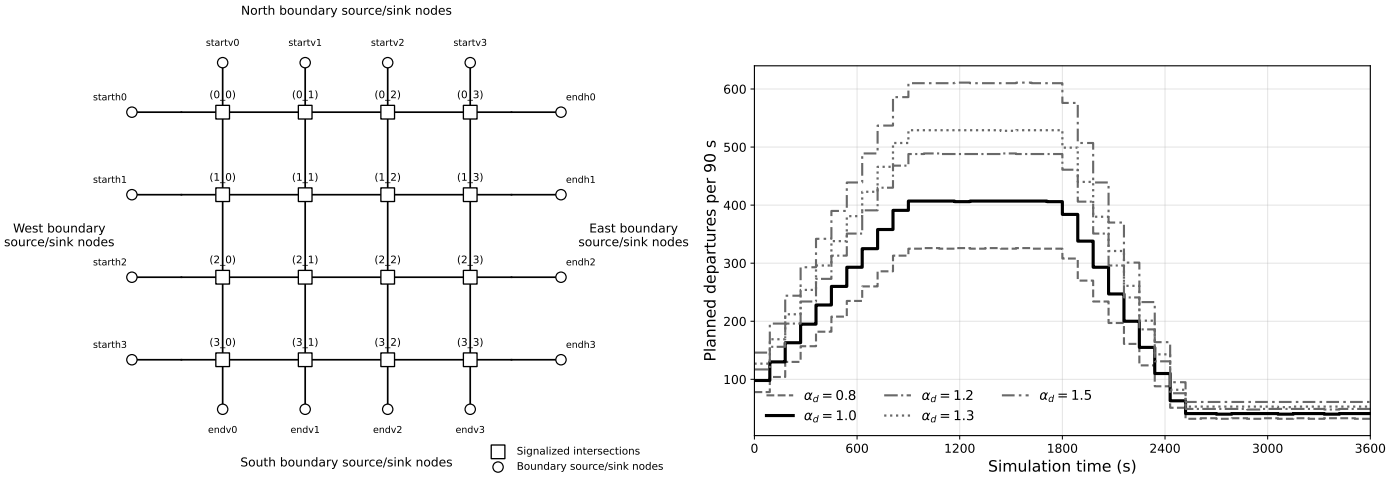
Initialize the policy networks, value networks, and Conv-VAE
for each training episode **do**
  for each decision instant t **do**
    Obtain local observations $\{\mathbf{o}_{v,t}\}_{v\in\mathcal{I}}$
    Construct $\mathbf{X}_t$ and obtain the latent representation $\mathbf{z}_t$
    Construct $\mathbf{O}_{v,t}$ and $\widetilde{\mathbf{O}}_{v,t}$ for each intersection
    Select $a_{v,t} \sim \pi_{\theta_v}(\cdot \mid \mathbf{O}_{v,t})$
    Apply the signal actions for the next control interval Δ_c
    Observe $r_{v,t}$ and the subsequent traffic state, and store the transitions
  end for
  Update the policy networks using L_π
  Update the value networks using L_V
  Update the Conv-VAE using L_{VAE}
end for

4. Simulation Studies

4.1. Simulation Setup

The comparative experiments are conducted on a 4×4 grid road network consisting of 16 signalized intersections, as illustrated in Figure 2a. The intersections are indexed from (0_0) to (3_3), while the external boundary nodes are denoted by `starth0-starth3`, `endh0-endh3`, `startv0-startv3`, and `endv0-endv3`. For each pair of adjacent nodes, two directed roads are defined, corresponding to the two opposite travel directions. Each road segment between two adjacent intersections is 400 m long and contains three lanes, including one exclusive left-turn lane, one through lane, and one shared through/right-turn lane. For the CTM-based MPC, each road is equally divided into four 100 m sections. The first three sections are represented by conventional CTM cells, whereas the terminal section adjacent to the downstream intersection is represented by two movement-specific cells, including one left-turn movement cell and one through/right-turn movement cell. The microscopic traffic evolution is simulated using SUMO, while the CTM is used only by the MPC controller for traffic prediction.

Both the MPC and LST-Net controllers make signal decisions every $\Delta_c = 5$ s, so that the two approaches are evaluated under the same control frequency. The SUMO simulation step and the CTM prediction step are both 1 s. Each simulation lasts 3600 s. The same signal-phase structure is adopted for all control methods. Each intersection employs four green phases: north–south through/right-turn, north–south protected left-turn, east–west through/right-turn, and east–west protected left-turn. For MPC and LST-Net, a minimum green time of 6 s, a maximum green time of 61 s, and a yellow transition time of 4 s are imposed for signal operation.



(**a**) (**b**)

Figure 2. Simulation network and traffic-demand profiles used in the comparative experiments: (**a**) simulation network; (**b**) time-varying traffic-demand profiles.

Five different traffic demand levels are considered, with demand scaling factors $\alpha_d \in \{0.8, 1.0, 1.2, 1.3, 1.5\}$. The 1.0-demand case is used as the reference demand profile. It follows a time-varying trapezoidal pattern and contains 8788 routed vehicles in the reference case. The corresponding temporal profiles are shown in Figure 2b, where the vertical axis represents the total number of planned vehicle departures within each 90 s interval. Specifically, the demand first increases during the initial 900 s, remains at its peak level for the following 900 s, decreases over the next 720 s, and subsequently enters a 1080 s recovery period. For the other demand levels, the same temporal demand pattern and routing structure are retained while the traffic demand is scaled by α_d. Consequently, the experiments evaluate controller performance from relatively uncongested conditions to increasingly saturated network states.

Five traffic signal control strategies are compared under the same network and traffic-demand conditions. Fixed-Time serves as the conventional baseline and operates according to a predefined signal timing plan. Max-Pressure is included as an established adaptive baseline and determines the signal allocation from movement-level traffic pressures without requiring an explicit traffic prediction model or offline policy training. MPC denotes the distributed model-based controller described in Section 2. Two variants of LST-Net are considered according to the reward formulation adopted during training. LST-Net-WT employs the waiting-time-based reward in Equation (17), whereas LST-Net-TD employs the local vehicle-travel-time-and-departure-delay-based reward in Equation (20).

For the Max-Pressure controller, let l and m denote two consecutive directed roads, and (l, m) denote the corresponding traffic movement through an intersection. The variable $x_{(l,m)}$ represents the stopped-vehicle queue associated with this movement, while $\beta_{(m,p)}$ denotes the turning ratio from road m to a downstream road p. The pressure of movement (l, m) is defined as

$$w_{(l,m)} = x_{(l,m)} - \sum_{p \in \mathcal{D}(m)} \beta_{(m,p)} x_{(m,p)}. \tag{31}$$

For each signal phase ϕ, the corresponding phase pressure is calculated as

$$P_\phi = \sum_{(l,m) \in \mathcal{M}_\phi} c_{\phi,(l,m)} w_{(l,m)}. \tag{32}$$

where $\mathcal{M}_\phi$ denotes the set of movements served by phase ϕ, and $c_{\phi,(l,m)}$ is the nominal service rate of movement (l, m) under phase ϕ. The movement service rates are constructed

from the per-lane saturation flow s_0 and the turning ratios. At the beginning of each signal cycle, the green duration of each phase is determined by

$$\begin{aligned} \mathbf{g}^* &= \arg\max_{\mathbf{g}} \textstyle\sum_{\phi\in\Phi} P_\phi g_\phi, \\ \text{s.t.} \quad & \textstyle\sum_{\phi\in\Phi} g_\phi = C - |\Phi| t_y, \\ & g_\phi^{\min} \le g_\phi \le g_\phi^{\max}, \qquad \phi \in \Phi. \end{aligned} \tag{33}$$

where C is the signal cycle length and $t_y = 4$ s is the yellow-transition duration. The main controller parameters are summarized in Table 1.

Table 1. Main parameters of the Max-Pressure, MPC and LST-Net controllers.

Method	Parameter	Value
Max-Pressure	Cycle length C	90 s
Max-Pressure	Through/right-turn green bounds $(g_{\mathrm{TR}}^{\min}, g_{\mathrm{TR}}^{\max})$	$(12, 45)$ s
Max-Pressure	Protected-left-turn green bounds $(g_{\mathrm{L}}^{\min}, g_{\mathrm{L}}^{\max})$	$(6, 12)$ s
Max-Pressure	Per-lane saturation flow s_0	1800 veh/h/lane
MPC	Prediction horizon H	175
MPC	Queue weight λ_q	0.58
MPC	Throughput weight λ_y	0.42
MPC	CTM capacity	1800 veh/h/lane
MPC	CTM free flow speed v_f	46.55 km/h
MPC	CTM jam density	130.10 veh/km/lane
MPC	CTM shockwave speed w	19.69 km/h
MPC	Confidence time points for $c(h)$	0, 15, 60, 120, 175 s
MPC	Confidence levels for $c(h)$	1, 1, 0.7, 0.4, 0.25
LST-Net	Temporal window W	3
LST-Net	Latent dimension d_z	16
LST-Net	Actor learning rate	1×10^{-4}
LST-Net	Critic learning rate	2×10^{-4}
LST-Net	Discount factor γ	0.99
LST-Net	GAE parameter λ	0.95
LST-Net	PPO clipping parameter ϵ	0.1
LST-Net	PPO epochs	10
LST-Net	Training episodes	80

4.2. Control Performance Across Demand Levels Under Matched Traffic Realizations

To reduce the influence of stochastic traffic dynamics and learning, five random seeds, $\{42, 43, 101, 102, 103\}$, are considered. The same set of seeds is used for all traffic-demand levels and control methods so that the compared controllers are evaluated under matched stochastic traffic realizations. For LST-Net, each model is trained for 80 episodes under a given random seed, with each episode lasting 3600 s. The performance metrics obtained from the final training episode are used to represent the trained policy for that seed. For MPC, Fixed-Time, and Max-Pressure control, the corresponding performance metrics are obtained directly from the simulation run associated with each random seed.

The overall control performance is evaluated using five traffic-efficiency measures: trip duration, waiting time, average speed, departure delay, and total time. The first four measures are obtained directly from the `vehicleTripStatistics` reported by SUMO. Specifically, duration represents the average trip duration, waiting time measures the average time that vehicles spend involuntarily standing, speed denotes the average trip speed, and departure delay measures the average delay between the intended and actual vehicle departure caused by insufficient road space. To account for both vehicles traveling

within the network and those waiting to enter, total time is calculated as the sum of `totalTravelTime` and `totalDepartDelay` reported by SUMO.

Table 2 summarizes the mean and standard deviation of these metrics over the five random seeds. Fixed-Time control is used as the reference method for calculating the improvement percentages. For duration, waiting time, departure delay, and total time, lower values indicate better performance, and the improvement is calculated as $(x_{\mathrm{FT}} - x)/x_{\mathrm{FT}} \times 100\%$. For average speed, higher values indicate better performance, and the corresponding improvement is calculated as $(v - v_{\mathrm{FT}})/v_{\mathrm{FT}} \times 100\%$. Therefore, a positive improvement denotes better performance than Fixed-Time control, whereas a negative value indicates performance degradation.

Table 2. Overall control performance under different traffic-demand levels. Values are reported as mean ± standard deviation over five random seeds, with the improvement relative to Fixed-Time shown in parentheses. Positive improvement indicates better performance.

α_d	Method	Duration (s)	Waiting Time (s)	Speed (m/s)	Depart Delay (s)	Total Time (10^6 s)
0.8	Fixed-Time	267.26 ± 0.95 (0.00%)	110.00 ± 0.83 (0.00%)	8.22 ± 0.02 (0.00%)	0.34 ± 0.01 (0.00%)	1.881 ± 0.007 (0.00%)
	Max-Pressure	268.29 ± 1.23 (−0.39%)	110.15 ± 0.99 (−0.13%)	7.97 ± 0.03 (−3.07%)	0.37 ± 0.01 (−9.41%)	1.889 ± 0.009 (−0.40%)
	LST-Net-TD	**196.81 ± 3.84 (26.36%)**	**41.46 ± 2.54 (62.31%)**	**10.68 ± 0.10 (29.92%)**	**0.31 ± 0.00 (9.41%)**	**1.386 ± 0.027 (26.34%)**
	LST-Net-WT	204.43 ± 3.62 (23.51%)	42.60 ± 1.66 (61.27%)	10.48 ± 0.11 (27.46%)	0.31 ± 0.01 (8.24%)	1.439 ± 0.025 (23.49%)
	MPC	252.37 ± 3.86 (5.57%)	94.40 ± 2.84 (14.18%)	8.35 ± 0.10 (1.51%)	0.36 ± 0.01 (−5.88%)	1.777 ± 0.027 (5.56%)
1.0	Fixed-Time	341.11 ± 1.05 (0.00%)	163.99 ± 0.86 (0.00%)	7.19 ± 0.01 (0.00%)	0.62 ± 0.02 (0.00%)	3.003 ± 0.009 (0.00%)
	Max-Pressure	336.14 ± 2.84 (1.46%)	158.34 ± 1.85 (3.45%)	6.87 ± 0.04 (−4.39%)	0.58 ± 0.02 (5.81%)	2.959 ± 0.025 (1.47%)
	LST-Net-TD	**285.18 ± 6.30 (16.40%)**	**101.89 ± 3.72 (37.86%)**	**8.68 ± 0.10 (20.78%)**	1.51 ± 1.85 (−142.77%)	2.519 ± 0.069 (16.11%)
	LST-Net-WT	326.82 ± 17.89 (4.19%)	112.20 ± 25.05 (31.58%)	8.20 ± 0.12 (14.11%)	2.76 ± 1.02 (−343.73%)	2.897 ± 0.152 (3.53%)
	MPC	285.94 ± 4.68 (16.17%)	120.22 ± 3.52 (26.69%)	7.65 ± 0.10 (6.43%)	**0.53 ± 0.02 (14.79%)**	**2.517 ± 0.041 (16.17%)**
1.2	Fixed-Time	463.93 ± 2.06 (0.00%)	255.83 ± 2.04 (0.00%)	6.17 ± 0.01 (0.00%)	**4.10 ± 0.46 (0.00%)**	4.935 ± 0.023 (0.00%)
	Max-Pressure	443.12 ± 9.91 (4.49%)	238.51 ± 7.54 (6.77%)	5.90 ± 0.09 (−4.41%)	12.14 ± 1.63 (−196.00%)	4.801 ± 0.114 (2.73%)
	LST-Net-TD	406.96 ± 12.62 (12.28%)	**187.41 ± 9.35 (26.74%)**	**7.20 ± 0.15 (16.76%)**	25.40 ± 11.23 (−519.72%)	4.559 ± 0.183 (7.62%)
	LST-Net-WT	521.30 ± 47.10 (−12.37%)	240.17 ± 62.97 (6.12%)	6.24 ± 0.14 (1.10%)	42.44 ± 10.43 (−935.53%)	5.996 ± 0.493 (−21.48%)
	MPC	**378.93 ± 4.60 (18.32%)**	191.50 ± 3.85 (25.15%)	6.65 ± 0.07 (7.71%)	5.42 ± 4.50 (−32.26%)	**4.053 ± 0.090 (17.88%)**
1.3	Fixed-Time	502.86 ± 2.52 (0.00%)	287.50 ± 1.47 (0.00%)	5.88 ± 0.03 (0.00%)	26.14 ± 1.82 (0.00%)	6.043 ± 0.023 (0.00%)
	Max-Pressure	481.93 ± 9.46 (4.16%)	269.68 ± 7.56 (6.20%)	5.59 ± 0.08 (−4.86%)	25.37 ± 1.80 (2.93%)	5.795 ± 0.104 (4.10%)
	LST-Net-TD	469.02 ± 15.65 (6.73%)	236.48 ± 17.51 (17.75%)	**6.81 ± 0.12 (15.78%)**	60.09 ± 12.71 (−129.90%)	6.077 ± 0.243 (−0.56%)
	LST-Net-WT	561.92 ± 54.27 (−11.75%)	269.35 ± 92.44 (6.31%)	5.88 ± 0.08 (0.00%)	63.72 ± 19.77 (−143.80%)	7.303 ± 0.686 (−20.85%)
	MPC	**420.88 ± 4.12 (16.30%)**	**222.55 ± 2.82 (22.59%)**	6.25 ± 0.07 (6.26%)	**8.36 ± 2.45 (68.02%)**	**4.904 ± 0.046 (18.86%)**
1.5	Fixed-Time	582.66 ± 3.75 (0.00%)	355.25 ± 3.32 (0.00%)	5.31 ± 0.03 (0.00%)	75.24 ± 5.35 (0.00%)	8.982 ± 0.044 (0.00%)
	Max-Pressure	554.27 ± 9.98 (4.87%)	328.57 ± 8.62 (7.51%)	4.99 ± 0.07 (−6.10%)	76.01 ± 3.73 (−1.03%)	8.325 ± 0.186 (7.31%)
	LST-Net-TD	579.08 ± 19.94 (0.62%)	341.18 ± 31.08 (3.96%)	**6.18 ± 0.22 (16.43%)**	125.89 ± 10.49 (−67.33%)	9.876 ± 0.458 (−9.96%)
	LST-Net-WT	639.71 ± 90.96 (−9.79%)	321.43 ± 127.82 (9.52%)	5.14 ± 0.19 (−3.20%)	139.69 ± 29.97 (−85.66%)	10.677 ± 1.183 (−18.88%)
	MPC	**539.60 ± 5.00 (7.39%)**	**320.89 ± 3.66 (9.67%)**	5.39 ± 0.07 (1.51%)	**25.43 ± 9.78 (66.20%)**	**7.452 ± 0.144 (17.04%)**

At the relatively low demand level of $\alpha_d = 0.8$, both LST-Net variants substantially outperform Fixed-Time control, with LST-Net-TD achieving the best overall performance. Compared with Fixed-Time, LST-Net-TD reduces the average trip duration and waiting time by 26.36% and 62.31%, respectively, while increasing the average speed by 29.92%. Its total time is reduced by 26.34%, compared with reductions of 23.49% for LST-Net-WT and 5.56% for MPC. In contrast, Max-Pressure produces performance close to Fixed-Time at this demand level, with its total time increasing slightly by 0.40%. This indicates that under relatively uncongested conditions, the learned policies can rapidly adapt signal service to the prevailing traffic demand.

When the demand scale increases to $\alpha_d = 1.0$, the performance difference between LST-Net-TD and MPC becomes substantially smaller in terms of the system-level total time.

LST-Net-TD and MPC achieve total-time reductions of 16.11% and 16.17%, respectively, and their mean total times are almost identical. Nevertheless, LST-Net-TD still produces a considerably lower waiting time and a higher average speed, with improvements of 37.86% and 20.78%, respectively. In contrast, MPC maintains a much smaller departure delay, improving this metric by 14.79%, whereas the departure delay of LST-Net-TD is higher than that of Fixed-Time control. Max-Pressure provides more modest improvements at this demand level, reducing trip duration, waiting time, departure delay, and total time by 1.46%, 3.45%, 5.81%, and 1.47%, respectively. These results reveal that improved mobility for vehicles already traveling within the network does not necessarily imply a corresponding reduction in the delay experienced by vehicles waiting to enter the network.

At $\alpha_d = 1.2$, the relative advantage begins to shift toward MPC. MPC achieves the lowest trip duration and total time, corresponding to improvements of 18.32% and 17.88% over Fixed-Time control. LST-Net-TD remains effective in reducing in-network congestion, achieving the lowest waiting time and the highest average speed, with improvements of 26.74% and 16.76%, respectively. However, its departure delay increases markedly, resulting in a smaller total-time improvement of 7.62%. Max-Pressure reduces trip duration and waiting time by 4.49% and 6.77%, respectively, and yields a 2.73% reduction in total time. In contrast, LST-Net-WT exhibits a 21.48% increase in total time relative to Fixed-Time control. The results therefore show that as the network approaches congested operation, performance evaluated solely from in-network indicators such as waiting time or speed can differ substantially from system-level performance when delayed vehicle departures are also considered.

The difference becomes more pronounced under high traffic demand. At $\alpha_d = 1.3$, MPC reduces the trip duration, waiting time, departure delay, and total time by 16.30%, 22.59%, 68.02%, and 18.86%, respectively. By comparison, although LST-Net-TD still increases the average speed by 15.78% and reduces waiting time by 17.75%, its total time becomes slightly higher than that of Fixed-Time control. Max-Pressure achieves smaller but positive reductions in trip duration, waiting time, and total time of 4.16%, 6.20%, and 4.10%, respectively. At $\alpha_d = 1.5$, MPC continues to achieve a 17.04% reduction in total time and a 66.20% reduction in departure delay. Max-Pressure maintains a 7.31% reduction in total time, although its departure delay is slightly higher than that of Fixed-Time control. In contrast, the total time of LST-Net-TD and LST-Net-WT increases by 9.96% and 18.88%, respectively, relative to Fixed-Time control.

An interesting observation is that LST-Net-TD maintains the highest average speed at all five demand levels, including the heavily congested cases. However, this advantage does not translate into lower total time at the highest demand levels because the associated departure delay increases substantially. Conversely, Max-Pressure exhibits a lower average speed than Fixed-Time at all five demand levels, while its total-time improvement increases from -0.40% at $\alpha_d = 0.8$ to 7.31% at $\alpha_d = 1.5$. This result emphasizes the importance of accounting for vehicles that have not yet entered the road network when evaluating traffic signal control under oversaturated conditions. Waiting time and average speed mainly characterize vehicles already operating within the network, whereas total departure delay captures the additional cost imposed on traffic demand waiting for network entry. Consequently, the combined total-time measure provides a more comprehensive indication of overall traffic-system efficiency.

Since the LST-Net results in Table 2 are obtained from the final episode of the 80-episode training process, the corresponding training trajectories are further examined to characterize the learning behavior over the adopted training budget. Figure 3 presents representative results for the reference demand level $\alpha_d = 1.0$ and the highest demand level $\alpha_d = 1.5$. At the reference demand level $\alpha_d = 1.0$, both LST-Net variants exhibit substantial improvements

during the early stage of training. The five-seed mean return changes rapidly during approximately the first 20–30 episodes and subsequently varies within a relatively stable range. Although individual seeds occasionally exhibit noticeable episodic fluctuations, the major improvement in the learned policies is obtained during the earlier part of the 80-episode training process.

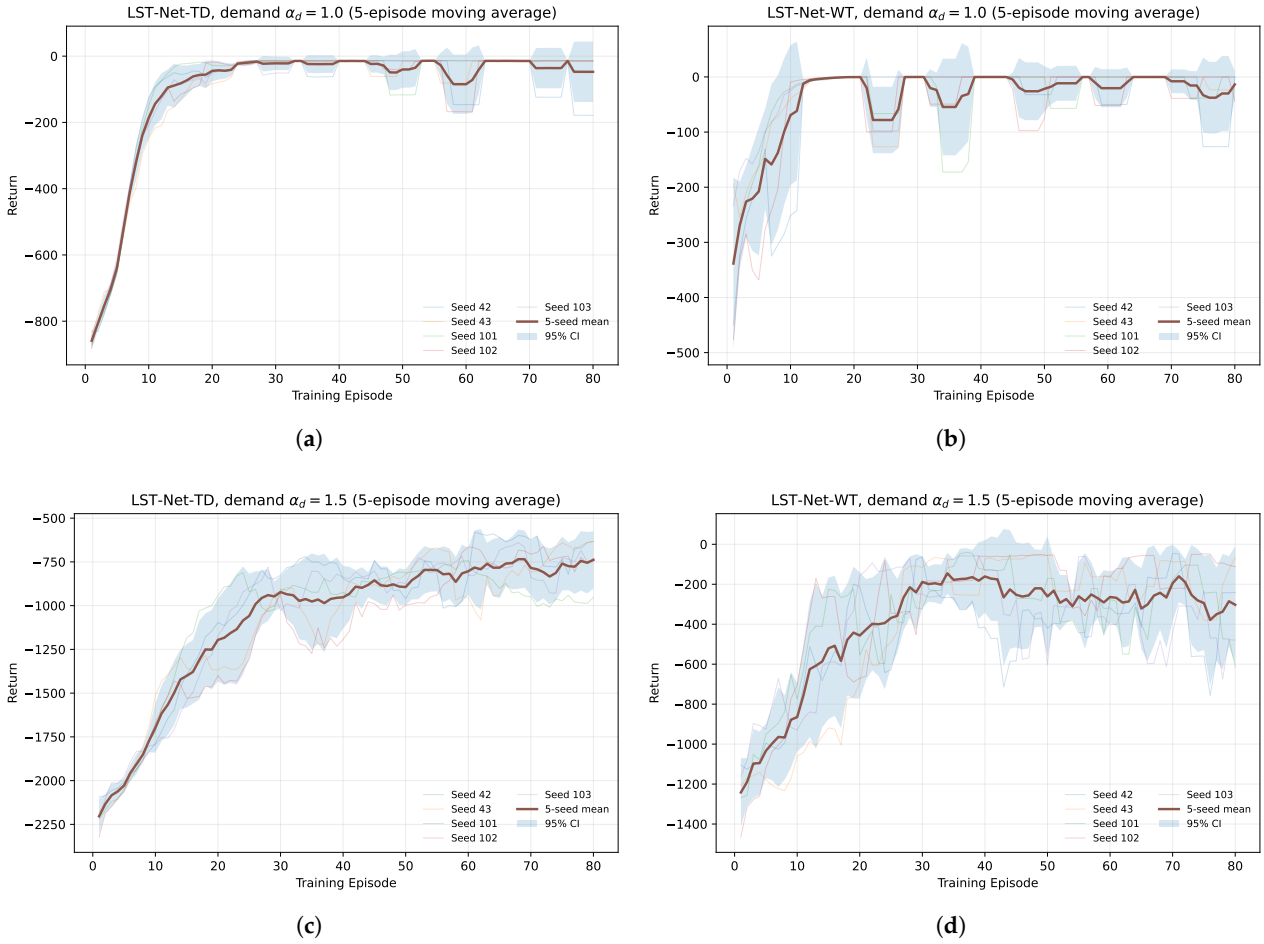


Figure 3. Training trajectories of LST-Net under the reference and highest demand levels: (**a**) LST-Net-TD at $\alpha_d = 1.0$; (**b**) LST-Net-WT at $\alpha_d = 1.0$; (**c**) LST-Net-TD at $\alpha_d = 1.5$; (**d**) LST-Net-WT at $\alpha_d = 1.5$. The individual curves correspond to the five random seeds, while the mean trajectory and its 95% confidence interval summarize the across-seed training behavior. A five-episode moving average is used for visualization. Since LST-Net-TD and LST-Net-WT employ different reward formulations, their absolute return values are not directly comparable.

At the highest demand level $\alpha_d = 1.5$, the training trajectories exhibit substantially larger fluctuations and more pronounced differences among the random seeds. LST-Net-TD shows an overall improvement over the training process, whereas LST-Net-WT exhibits more frequent performance variations after its initial improvement. Similar increases in training variability are also observed at the intermediate high-demand levels. These observations indicate that policy learning becomes more sensitive to stochastic traffic evolution as the network operates closer to heavily congested conditions. Additional training experiments with an extended number of episodes were also conducted for the high-demand cases, but increasing the training horizon did not yield consistent improvements in the resulting control performance. Therefore, the larger fluctuations observed under high demand should not be interpreted simply as a consequence of an insufficient number of

training episodes. The 80-episode setting is consequently retained as a common training budget to ensure a consistent comparison across demand levels and reward formulations.

4.3. Representative Traffic Evolution and Signal-Allocation Behavior

The above analyses characterize the network-level performance and learning behavior under the matched traffic realizations. To further examine the signal-allocation behavior underlying the network-level performance, the realized phase sequences are analyzed using a representative matched traffic realization corresponding to random seed 42. Three consecutive edges, (2_0)–(2_1), (2_1)–(2_2), and (2_2)–(2_3), are selected along the same corridor. In the edge notation, (i)–(j) denotes the directed edge from node i to node j, and hence represents traffic traveling from the first node to the second. For each selected edge, the traffic-light switching records are used to summarize the realized green-time allocation at its downstream intersection.

Table 3 illustrates the mean continuous green duration, the number of continuous green intervals, and the total green duration over the 3600 s simulation for the through/right-turn phase. Several observations can be obtained from Table 3. First, the most consistent distinction among the controllers appears in the duration and frequency of continuous through/right-turn service. Across all three selected edges and all five demand levels, MPC exhibits the longest mean continuous through/right-turn green duration and, simultaneously, the smallest number of green intervals. For example, on edge (2_1)–(2_2), the mean continuous green duration under MPC increases from 17.79 s at $\alpha_d = 0.8$ to 46.48 s at $\alpha_d = 1.5$, while the corresponding number of green intervals decreases from 47 to 29. In comparison, LST-Net-WT generally maintains continuous through/right-turn greens of approximately 9–11 s on this edge and activates the phase much more frequently.

Table 3. Through/right-turn phase allocation for the selected edges under different traffic-demand levels using random seed 42. Each entry is reported as mean continuous green duration/number of green intervals/total green duration, where the two duration quantities are measured in seconds.

Edge	Method	$\alpha_d = 0.8$	$\alpha_d = 1.0$	$\alpha_d = 1.2$	$\alpha_d = 1.3$	$\alpha_d = 1.5$
(2_0)-(2_1)	LST-Net-WT	10.53/106/1116	11.26/96/1081	9.40/125/1175	9.28/119/1104	9.58/123/1178
	LST-Net-TD	11.90/105/1250	20.47/66/1351	22.72/61/1386	16.99/86/1461	14.59/78/1138
	MPC	14.66/53/777	22.14/43/952	29.18/39/1138	33.11/35/1159	40.68/34/1383
(2_1)-(2_2)	LST-Net-WT	10.44/99/1034	10.10/106/1071	9.38/117/1097	10.10/111/1121	10.09/121/1221
	LST-Net-TD	13.77/74/1019	14.97/68/1018	22.36/59/1319	17.91/68/1218	14.80/71/1051
	MPC	17.79/47/836	23.41/41/960	29.31/36/1055	35.52/33/1172	46.48/29/1348
(2_2)-(2_3)	LST-Net-WT	9.24/142/1312	9.90/127/1257	10.35/108/1118	14.79/103/1523	11.54/111/1281
	LST-Net-TD	11.65/123/1433	12.88/109/1404	13.19/112/1477	13.27/99/1314	13.08/96/1256
	MPC	18.10/52/941	20.33/46/935	23.74/47/1116	27.91/44/1228	39.18/39/1528

The two LST-Net variants also exhibit different phase-allocation patterns. In most of the examined cases, LST-Net-TD produces longer continuous through/right-turn service and fewer phase activations than LST-Net-WT. This difference is particularly evident on edges (2_0)–(2_1) and (2_1)–(2_2) when the demand scale is between 1.0 and 1.3. For instance, at $\alpha_d = 1.2$, LST-Net-TD provides a mean continuous through/right-turn green of 22.72 s with 61 intervals on edge (2_0)–(2_1), compared with 9.40 s and 125 intervals for LST-Net-WT. This suggests that the alternative reward formulation affects not only the resulting traffic performance but also the temporal organization of the learned signal actions.

In contrast to the relatively clear differences in continuous green duration and activation frequency, the total green duration does not exhibit a uniform ordering among the three controllers. For example, at $\alpha_d = 0.8$ on edge (2_0)–(2_1), MPC activates the

through/right-turn phase only 53 times with a mean duration of 14.66 s, but its cumulative through/right-turn green time is 777 s, which is lower than those of both LST-Net variants. At $\alpha_d = 1.5$, however, the cumulative through/right-turn green time under MPC increases to 1383 s on the same edge. These results indicate that the distinction among the controllers is related not only to the amount of green time allocated but also to how that green time is temporally distributed.

Another notable feature is the demand-dependent evolution of the MPC phase allocation. For all three selected edges, the mean continuous through/right-turn green duration under MPC increases monotonically as the demand scale increases. At the same time, the number of through/right-turn green intervals generally decreases. This representative result indicates that, as traffic demand increases, the MPC controller tends to realize the required through/right-turn service through increasingly longer continuous green periods rather than through more frequent phase activations.

Time-series trajectories are further examined to illustrate how the different phase-allocation patterns are accompanied by different edge-level traffic evolution. Figures 4–6 show the traffic evolution and signal-phase allocation on edges (2_0)–(2_1), (2_1)–(2_2), and (2_2)–(2_3), respectively, under the demand level $\alpha_d = 1.3$ and random seed 42. The edge traffic state is aggregated over 300 s and the signal panels show the realized through/right-turn green intervals.

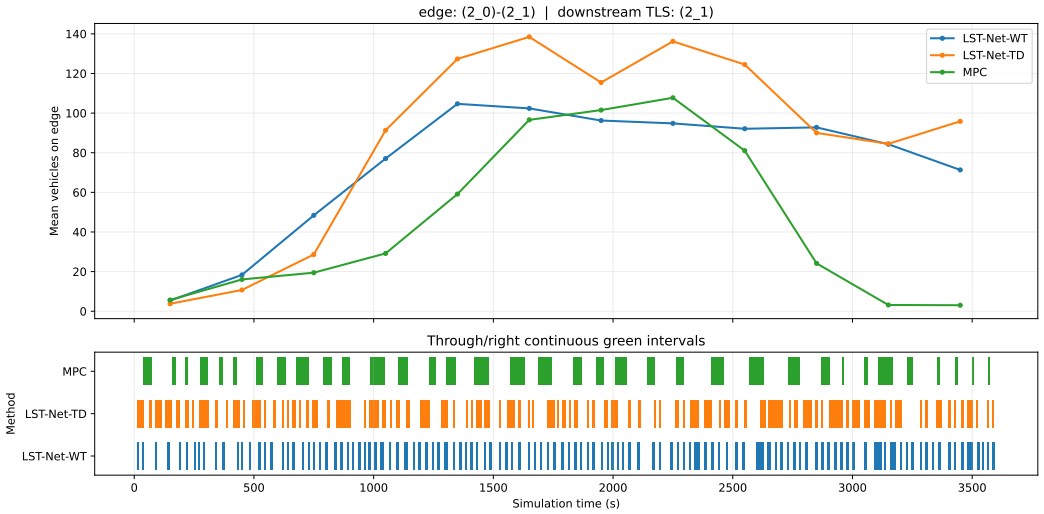


Figure 4. Representative traffic evolution and signal-phase allocation on edge (2_0)–(2_1) under the demand level $\alpha_d = 1.3$ and random seed 42.

It can be observed that, for most of the simulation period, the mean numbers of vehicles on edges (2_0)–(2_1) and (2_1)–(2_2) are lower under MPC than under the two LST-Net controllers. In contrast, on edge (2_2)–(2_3), MPC generally results in a slightly higher vehicle accumulation. Nevertheless, the traffic accumulation on edge (2_2)–(2_3) remains relatively low for all three methods: the mean number of vehicles is approximately 20–25 under MPC and approximately 10–15 under LST-Net-TD and LST-Net-WT for most of the simulation. Compared with edge (2_2)–(2_3), considerably larger vehicle accumulations are observed on edges (2_0)–(2_1) and (2_1)–(2_2), indicating more severe traffic congestion along these two road segments.

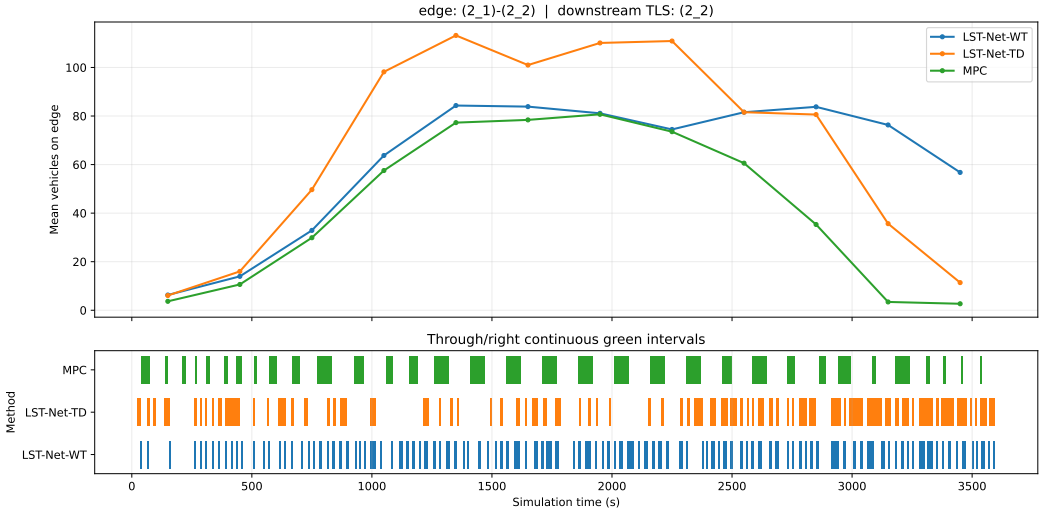


Figure 5. Representative traffic evolution and signal-phase allocation on edge (2_1)–(2_2) under the demand level $\alpha_d = 1.3$ and random seed 42.

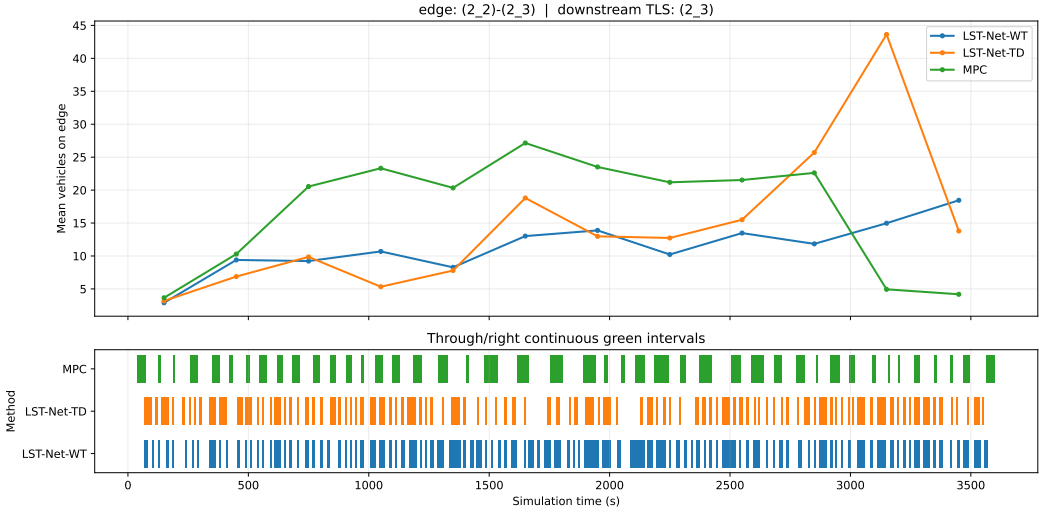


Figure 6. Representative traffic evolution and signal-phase allocation on edge (2_2)–(2_3) under the demand level $\alpha_d = 1.3$ and random seed 42.

The differences among the controllers become particularly evident during the recovery period near the end of the simulation. Under MPC, the vehicle accumulations on all three selected edges decrease to relatively low levels, indicating that the corridor successfully recovers from the preceding congested condition. For edge (2_0)–(2_1), however, the vehicle accumulation remains relatively high at the end of the simulation under both LST-Net-TD and LST-Net-WT. On edge (2_1)–(2_2), LST-Net-TD is also able to reduce the vehicle accumulation to a relatively low level, whereas a substantial number of vehicles remain under LST-Net-WT. These observations are consistent with the superior system-level performance of MPC under the $\alpha_d = 1.3$ demand level reported in Section 4.2, particularly in terms of total time and departure delay.

The signal-phase trajectories further illustrate the differences in phase-allocation behavior. During approximately 1200–2500 s, substantial vehicle accumulation is observed along the corridor. For the congested edge (2_1)–(2_2) whose downstream edge (2_2)–(2_3) contains a relatively low number of vehicles, MPC generally maintains relatively long continuous through/right-turn green intervals during this period. In contrast, the through/right-turn green durations for the upstream edge (2_0)–(2_1) vary more substantially under MPC, with both relatively long and short green intervals appearing during the same congested

period. This behavior coincides with the higher and more variable traffic accumulation on the downstream edge (2_1)–(2_2). No similarly consistent relationship can be identified from the signal trajectories of the two LST-Net controllers. Nevertheless, their continuous through/right-turn green intervals during the congestion period are generally shorter than those under MPC, which is consistent with the phase-allocation statistics reported in Table 3.

4.4. Robustness Under Distinct Traffic Realizations

To further evaluate robustness beyond the matched-realization setting, all controllers are tested on five distinct traffic realizations generated using random seeds 301–305 under the reference demand level $\alpha_d = 1.0$. Each realization preserves the aggregate number of scheduled departures in every 90 s interval of the original demand profile and therefore maintains the same overall rising–plateau–falling temporal pattern as Figure 2b. However, the demand assigned to individual origin edges is varied across realizations, where the origin edges include both boundary-connected and internal edges within the network. The corresponding vehicle-level departure times and route assignments are also regenerated using the respective random seeds. Figure 7 illustrates the resulting origin-level redistribution of traffic demand among traces 301–305. Thus, the five traces exhibit distinct spatial demand distributions and microscopic traffic realizations, providing a more stringent test of controller robustness than the matched-realization experiments in Section 4.2.

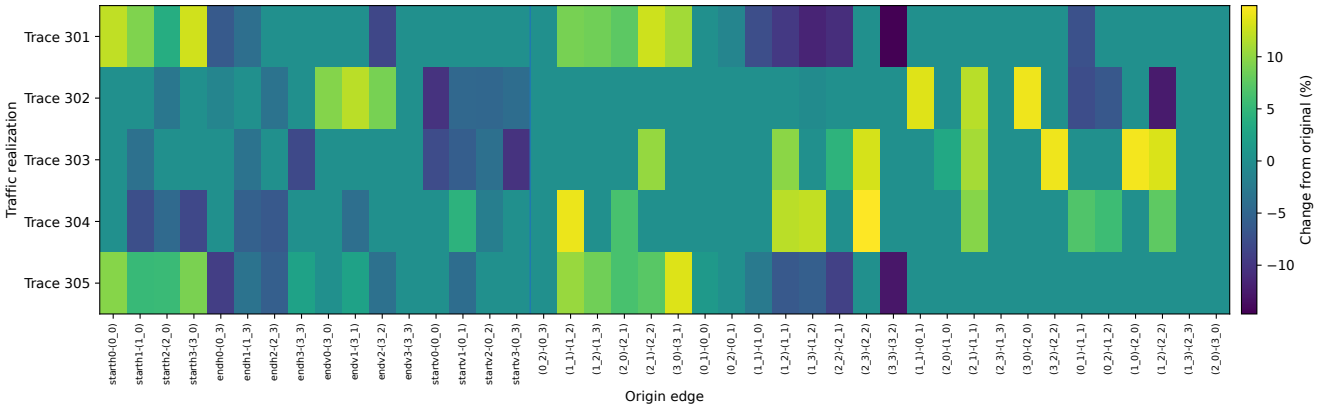


Figure 7. Origin-level demand variations of the distinct traffic realizations 301–305 relative to the original $\alpha_d = 1.0$ realization. Each column represents one origin edge, and the color indicates the percentage change in the total number of vehicles departing from that origin over the 3600 s simulation. Although the aggregate number of scheduled departures in each 90 s interval is preserved exactly, the spatial allocation of demand differs among the five realizations.

Moreover, for LST-Net, the policy parameters obtained from the original training experiments described in Section 4.2 are kept fixed throughout this evaluation, and no retraining or fine-tuning is performed on traces 301–305. The five trained policies obtained using training seeds $\{42, 43, 101, 102, 103\}$ under the original training demand setting are evaluated on traces 301–305, respectively. These new traffic realizations are not used for policy training, and no additional retraining or fine-tuning is performed before evaluation. Although the policy parameters remain unchanged, the signal-control actions are still generated online according to the traffic states observed in each new realization.

Table 4 reports the total-time performance of the five controllers for each of the distinct traffic realizations. For a given trace, all controllers are evaluated using the same route realization and the same SUMO random seed. The trace-wise results show that the relative performance of the controllers varies with the detailed traffic realization. MPC achieves the lowest total time for traces 301, 303, and 305, whereas LST-Net-TD achieves the lowest

values for traces 302 and 304. Averaged over the five realizations, MPC yields the lowest total time of 3.173×10^6 s, followed by LST-Net-TD with 3.414×10^6 s. Max-Pressure remains close to Fixed-Time, while LST-Net-WT produces a higher mean total time than Fixed-Time. Consistent with the observation in Section 4.2, LST-Net-WT achieves the lowest mean in-network waiting time (150.69 s), but this is accompanied by a substantially larger mean departure delay (7.74 s). Consequently, its advantage in in-network waiting does not translate into improved overall system efficiency once the delay incurred before network entry is also considered.

Table 4. Total-time performance under five distinct traffic realizations at $\alpha_d = 1.0$.

Method	Trace 301	Trace 302	Trace 303	Trace 304	Trace 305	Mean $\pm$ Std.
Fixed-Time	3.555	3.586	3.457	3.485	3.732	3.563 ± 0.108
LST-Net-TD	3.459	**3.150**	3.160	**3.301**	3.999	3.414 ± 0.351
LST-Net-WT	3.670	3.449	3.707	3.806	3.863	3.699 ± 0.160
MPC	**3.069**	3.247	**3.078**	3.334	**3.137**	$\mathbf{3.173 \pm 0.114}$
Max-Pressure	3.539	3.442	3.488	3.578	3.629	3.535 ± 0.073

Note: Total time is reported in 10^6 s. The standard deviation is calculated across traces 301–305.

The across-realization variability further reveals clear differences in robustness among the controllers. Max-Pressure exhibits the smallest standard deviation of the total-time metric (0.073×10^6 s), followed by Fixed-Time (0.108×10^6 s) and MPC (0.115×10^6 s). Although Max-Pressure is the least sensitive to the change in traffic realization, its mean performance remains close to that of Fixed-Time. MPC, in contrast, combines the lowest mean total time with relatively small variability, indicating that its performance advantage is maintained consistently across the five realizations. LST-Net-TD exhibits the largest standard deviation (0.351×10^6 s), mainly because its performance varies substantially across traces and deteriorates noticeably on trace 305. LST-Net-WT also shows larger variability than MPC and Fixed-Time. These results suggest that the learning-based controllers, particularly LST-Net-TD, are more sensitive to changes in the detailed spatial and microscopic traffic realization, whereas MPC provides a more consistent system-level response.

Since all controllers are evaluated on the same five traffic realizations, paired statistical comparisons are further conducted using Fixed-Time as the reference. This matched design allows the performance difference between two controllers to be evaluated within the same traffic realization, thereby reducing the influence of between-realization variability. The paired difference is defined as the total time of Fixed-Time minus that of the compared controller, such that a positive value indicates an improvement over Fixed-Time. The resulting paired differences, 95% confidence intervals, p-values, and paired Cohen's d are summarized in Table 5. Given the modest number of matched realizations, the statistical results are interpreted jointly with the trace-wise performance, confidence intervals, and effect sizes to provide a more comprehensive assessment of performance consistency.

The paired comparison confirms that the improvement achieved by MPC is both large and consistent across the five traffic realizations. Relative to Fixed-Time, MPC reduces the total-time metric by an average of 0.390×10^6 s, with a 95% confidence interval of $[0.183, 0.597] \times 10^6$ s and a paired t-test p-value of 0.006. The corresponding paired Cohen's d of 2.34 further indicates that the reduction is large relative to the across-realization variability of the paired differences. LST-Net-TD also achieves a positive mean reduction of 0.149×10^6 s, but its confidence interval crosses zero because of the larger realization-to-realization variation, particularly the performance degradation on trace 305. The paired differences for Max-Pressure and LST-Net-WT likewise do not provide clear evidence of a consistent improvement over Fixed-Time across these five realizations.

Table 5. Paired statistical comparison of total time relative to Fixed-Time across traces 301–305.

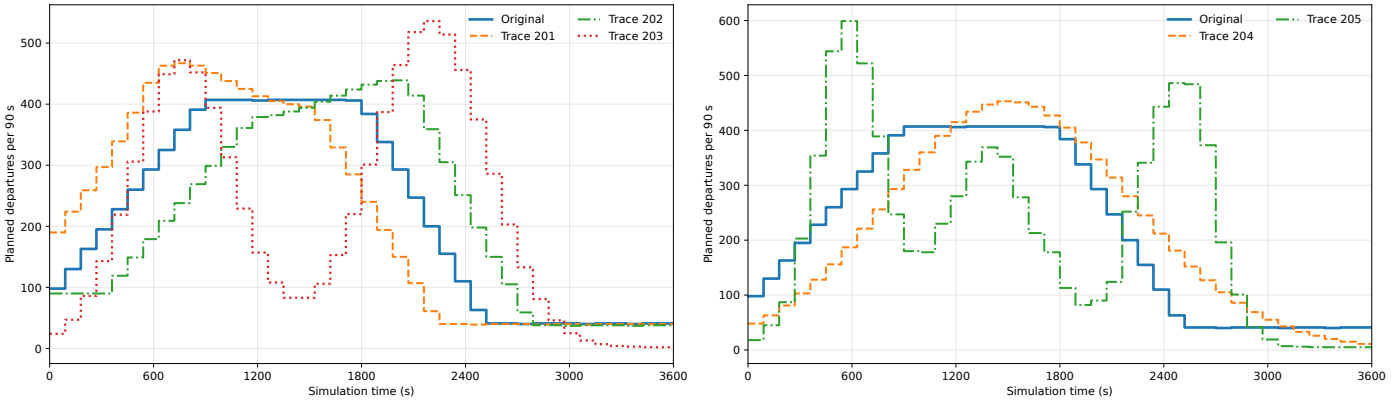
Method	Mean Paired Difference	95% CI	*p*-Value	Cohen's *d*
MPC	0.390	[0.183, 0.597]	**0.006**	2.34
LST-Net-TD	0.149	[−0.180, 0.479]	0.277	0.56
Max-Pressure	0.028	[−0.092, 0.148]	0.555	0.29
LST-Net-WT	−0.136	[−0.353, 0.081]	0.157	−0.78

Note: Mean paired differences and confidence intervals are reported in 10^6 s. Positive values indicate lower total time than Fixed-Time. The *p*-values are obtained from two-sided paired *t*-tests.

4.5. Generalization to Unseen Traffic Scenarios

The preceding subsection evaluates controller robustness under distinct microscopic traffic realizations while preserving the aggregate temporal demand profile. To further examine generalization beyond this setting, five unseen traffic scenarios, denoted by traces 201–205, are constructed at the reference demand level $\alpha_d = 1.0$. Unlike traces 301–305, these scenarios modify not only the microscopic realization but also the temporal demand pattern, spatial allocation of origin demand, and route-choice distribution, while keeping the total demand fixed at 8788 vehicles. For LST-Net, the policies trained under the original $\alpha_d = 1.0$ demand setting using training seeds $\{42, 43, 101, 102, 103\}$ are directly evaluated on traces 201–205, respectively. The policy parameters are kept fixed throughout the evaluation, and no retraining or fine-tuning is performed for the unseen scenarios.

In addition to the temporal and spatial changes summarized in Table 6, the route-choice probabilities among the valid route templates associated with each origin are perturbed relative to those in the original traffic realization. This changes the resulting route composition and, consequently, the turning ratios observed within the network. Consequently, traces 201–205 represent broader traffic-demand distribution shifts than traces 301–305, rather than different random realizations of an otherwise unchanged demand pattern. The resulting temporal demand profiles are illustrated in Figure 8. The unseen scenarios retain the same total number of vehicles but redistribute the demand over time, thereby producing temporal loading conditions that are not represented by a simple scaling of the original demand profile.



(**a**) (**b**)

Figure 8. Temporal demand profiles of the five unseen traffic scenarios compared with the original $\alpha_d = 1.0$ demand profile: (**a**) the original demand profile and traces 201–203; (**b**) the original demand profile and traces 204–205. All scenarios contain the same total number of 8788 vehicles but exhibit different temporal demand patterns.

Table 6. Characteristics of the five unseen traffic scenarios.

Trace	Temporal Demand Pattern	Spatial Demand Pattern
201	Earlier peak	Increased eastbound demand
202	Later peak	Increased north–south demand
203	Double peak	Central internal-demand hotspot
204	Broad peak	East–west directional reversal
205	Irregular demand pulses	Heterogeneous origin demand

Table 7 compares the total-time performance of the five controllers under the unseen traffic scenarios. The relative ranking varies across scenarios, reflecting the substantially different temporal and spatial loading patterns. Max-Pressure achieves the lowest total time for traces 201 and 205, LST-Net-TD performs best for traces 202 and 203, and MPC achieves the lowest value for trace 204. Nevertheless, MPC consistently reduces the total time relative to Fixed-Time in all five unseen scenarios, with reductions ranging from approximately 2.1% to 9.1%. Its mean total time is also the lowest, at 4.677×10^6 s, compared with 4.906×10^6 s for Fixed-Time.

Table 7. Total-time performance under the five unseen traffic scenarios.

Method	Trace 201	Trace 202	Trace 203	Trace 204	Trace 205	Mean $\pm$ Std.
Fixed-Time	5.773	4.724	3.906	5.414	4.713	4.906 ± 0.721
Max-Pressure	**5.091**	4.636	4.166	5.458	**4.225**	4.715 ± 0.557
MPC	5.426	4.605	3.823	**4.923**	4.608	$\mathbf{4.677 \pm 0.583}$
LST-Net-TD	6.498	**4.302**	**3.815**	5.301	4.569	4.897 ± 1.044
LST-Net-WT	5.467	5.300	4.846	6.198	5.372	5.437 ± 0.488

Note: Total time is reported in 10^6 s. The standard deviation describes the dispersion across the five structurally different scenarios.

The two learning-based controllers exhibit greater scenario dependence. LST-Net-TD provides substantial improvement in trace 202 and remains competitive in traces 203–205, but its total time increases markedly in trace 201, resulting in the largest across-scenario variation among the evaluated controllers. LST-Net-WT generalizes less consistently, outperforming Fixed-Time only in trace 201 and yielding higher total time in traces 202–205. These results indicate that the frozen LST-Net policies can retain effective control behavior under some unseen demand structures, but their performance is more sensitive to the specific form of the distribution shift.

Table 8 reports additional performance metrics to further explain the differences in total-time performance. Duration, waiting time, and speed characterize vehicle operation after entering the network, whereas depart delay measures the delay incurred before a vehicle is successfully inserted into the network. MPC achieves the lowest mean trip duration while maintaining a substantially lower depart delay than Fixed-Time, which contributes to its consistent system-level improvement across the unseen scenarios. Max-Pressure achieves the lowest mean depart delay and also reduces the mean waiting time, although its mean speed is lower than that of Fixed-Time.

LST-Net-TD achieves the highest mean speed and the lowest mean in-network waiting time, but these advantages are accompanied by considerably larger scenario-to-scenario variation, particularly in trip duration and waiting time. For LST-Net-WT, although the mean in-network waiting time remains slightly below that of Fixed-Time, the larger trip duration and depart delay lead to a higher overall total time. This is consistent with the behavior discussed in Section 4.2, where reducing in-network waiting alone does not necessarily improve system-level efficiency when entry delay is also taken into account.

Table 8. Additional performance metrics across the five unseen scenarios.

Method	Duration (s)	Waiting Time (s)	Speed (m/s)	Depart Delay (s)
Fixed-Time	548.35 ± 82.17	316.22 ± 70.95	6.08 ± 0.38	9.18 ± 8.06
Max-Pressure	532.75 ± 62.04	293.36 ± 53.46	5.79 ± 0.29	**3.79 ± 2.36**
MPC	**525.75 ± 66.38**	302.87 ± 57.54	6.05 ± 0.31	4.81 ± 3.02
LST-Net-TD	549.89 ± 114.33	**289.37 ± 109.34**	**6.67 ± 0.61**	6.15 ± 3.45
LST-Net-WT	596.21 ± 63.50	312.04 ± 88.94	6.26 ± 0.33	14.82 ± 12.10

Note: Values are reported as mean ± standard deviation across traces 201–205. Duration, waiting time, and speed characterize in-network vehicle operation, whereas depart delay represents the delay before network entry. Lower values are better for duration, waiting time, and depart delay, while higher values are better for speed.

4.6. Discussion

The comparative results show that the relative performance of the considered controllers changes with traffic demand. Under low-to-moderate demand, LST-Net generally achieves favorable traffic efficiency, whereas MPC exhibits more consistent performance as demand increases and the network becomes heavily congested. This observation should not be interpreted as an inherent superiority of one control paradigm over the other, since the relative performance also depends on the specific implementations considered in this study, including the state representation and policy architecture of LST-Net, as well as the traffic model, prediction horizon, control formulation, and parameter settings of MPC. Nevertheless, the convergence analysis reveals a notable difference on the learning side: compared with the lower-demand cases, LST-Net exhibits larger training-return fluctuations across random seeds under high demand, while extending the number of training episodes does not consistently improve the resulting control performance. This increased training variability may partly contribute to the weaker and less consistent performance of the learned policies under heavily congested conditions.

A possible reason for this increased learning difficulty is that heavily congested traffic produces stronger interactions among queues, signal decisions, and downstream traffic states. As demand approaches or exceeds network capacity, queue spillback and competition among movements become more pronounced, so that a local signal decision can affect not only the current intersection but also the traffic states subsequently encountered by neighboring controllers. This makes coordinated control more difficult to learn from repeated interactions with the environment. Similar difficulties have been reported for reinforcement-learning-based traffic signal control under oversaturated conditions, where large differences in competing traffic movements and rapidly evolving congestion states complicate the use of short-term control decisions to represent the overall traffic condition [28]. Together with the convergence behavior observed in this study, these results suggest that the deterioration of LST-Net at high demand is unlikely to be explained simply by insufficient training duration; rather, the increased complexity and interaction of congested traffic may make stable policy learning and generalization more difficult.

Beyond the increased training sensitivity under high demand, another important observation concerns the distinction between in-network traffic efficiency and network-entry performance. In several experiments, LST-Net maintains relatively high speeds or low waiting times for vehicles that have entered the network, while simultaneously producing a larger depart delay. This indicates that favorable traffic conditions within the network can coexist with increased delay before vehicles are successfully inserted into the network. A related distinction between maintaining favorable internal traffic conditions and accounting for delays outside a controlled region has also been recognized in network-level traffic management studies [29]. In the present study, this effect is particularly evident for LST-Net-WT, for which a relatively low in-network waiting time does not consistently translate into a lower total time once depart delay is taken into account. This finding

highlights the importance of considering both in-network mobility and network-entry delay when evaluating control performance under high traffic demand.

From this system-level perspective, another pattern observed in the results is the comparatively consistent total-time performance of MPC as traffic demand increases. A possible explanation lies in its explicit use of traffic-state prediction. Rather than selecting the control action solely according to the current traffic state, MPC evaluates the expected evolution of traffic over a finite prediction horizon and repeatedly updates the control decision as new state information becomes available. This look-ahead mechanism can be particularly useful under heavy demand, where a signal decision at the current step may influence subsequent queue accumulation, downstream congestion, and spillback. Previous studies on predictive traffic-signal control have similarly shown that considering future traffic evolution can improve control performance, especially when congestion propagation and spillback are relevant; however, the benefit also depends on the prediction horizon and the accuracy of the traffic model [30]. Therefore, the relatively consistent performance of MPC observed here may be partly attributed to its ability to anticipate short-term traffic evolution, although this advantage remains dependent on the adopted model, prediction horizon, and traffic-state information.

This comparatively consistent behavior of MPC across demand levels raises a further question: whether similar patterns persist when the traffic realization or demand structure differs from that used in the main comparison. The results under traces 301–305 show that, even when the aggregate temporal demand profile is preserved, changes in spatial loading, microscopic departure realization, and route composition can still lead to noticeable differences in controller performance. These differences become more pronounced in traces 201–205, where the temporal demand profile, spatial loading, and route-choice distribution are altered simultaneously. Under these changes, MPC continues to provide relatively consistent system-level performance, whereas the frozen LST-Net policies exhibit stronger scenario dependence. In particular, LST-Net-TD remains competitive and performs very well in several unseen scenarios, but its performance varies more substantially across scenarios, suggesting that the impact of distribution shift on policy performance is non-negligible. This behavior is consistent with previous studies identifying generalization to previously unexperienced traffic-demand distributions as an important challenge for reinforcement-learning-based traffic signal control [31]. These findings therefore suggest that the learned policy retains useful control capability beyond the original training conditions, but its effectiveness remains sensitive to the characteristics of unseen traffic conditions.

The dependence of controller performance on traffic conditions is also reflected in the behavior of Max-Pressure. Compared with MPC and LST-Net, Max-Pressure determines signal actions directly from local traffic-pressure information without explicitly predicting future traffic evolution or learning a control policy from training data. Its theoretical foundation is primarily associated with queue balancing, network stability, and throughput under specified traffic assumptions [8]. In the present experiments, Max-Pressure exhibits relatively small variation across traces 301–305 and achieves the lowest total time in some of the unseen scenarios, while in other cases, its performance remains close to or below that of the other adaptive controllers. This variation indicates that responsiveness to local queue imbalance can provide effective control under certain traffic conditions, but does not necessarily lead to the lowest system-level travel cost in every scenario. This variation further suggests that the relative effectiveness of different adaptive control mechanisms depends on how their decision mechanisms interact with the prevailing traffic conditions, and that no single control strategy is uniformly superior across all scenarios.

Several limitations should nevertheless be considered when interpreting the present results. The experiments cover multiple demand levels, traffic realizations, spatial and

temporal demand variations, and unseen traffic scenarios, but they are conducted within a controlled microscopic simulation environment. Therefore, the results primarily characterize the comparative behavior of the considered controllers under the examined traffic conditions rather than their readiness for direct field deployment. In practice, controller performance can also be affected by site-specific demand patterns, heterogeneous traffic composition, pedestrian interactions, incidents, imperfect sensing, communication delays, actuator constraints, and discrepancies between simulated and real traffic dynamics. These factors may influence different control approaches in different ways and remain important considerations for real-world traffic-signal control. In particular, previous studies have noted the limited availability of field-oriented testbeds and the persistent simulation-to-reality gap in reinforcement-learning-based traffic signal control [17,32].

Existing field studies further illustrate that transferring an advanced traffic-signal controller from simulation to practice involves additional system-level considerations beyond the control algorithm itself. For example, field implementations of MPC have required dedicated sensing, real-time computation, actuation, and monitoring infrastructure [33], whereas practical implementations of Max-Pressure have adopted more readily measurable traffic variables, such as travel time, in place of difficult-to-observe queue states [34]. These examples also indicate that practical challenges are method-dependent: model-based approaches require sufficiently accurate models and state information, while learning-based approaches additionally face issues related to training-data coverage and policy transfer beyond the simulated environment. Future work should therefore examine the considered controllers under field-calibrated traffic conditions, imperfect sensing and communication, and progressively more realistic hardware- or field-in-the-loop settings before conclusions regarding large-scale real-world deployment can be drawn.

5. Conclusions

This study presents a comparative evaluation of two implemented distributed traffic signal control approaches, namely a CTM-based MPC controller and the LST-Net learning-based controller, on a 4×4 grid road network under multiple traffic demand levels. Fixed-Time and Max-Pressure are included as reference controllers to provide both non-adaptive and adaptive benchmarks. In the MPC approach, each intersection predicts local traffic evolution over a finite horizon using the cell transmission model and exchanges predicted boundary traffic information with neighboring controllers to support coordinated signal control. In the LST-Net approach, temporal local traffic observations are used for local signal decision making, while a compact latent representation of the network-level traffic state is incorporated during centralized training to improve value estimation and policy learning. All controllers are evaluated under the same network configuration, signal-phase structure, control interval, and traffic-demand scenarios, with multiple random seeds, distinct traffic realizations, and unseen traffic scenarios also considered in the evaluation.

The results show that the relative performance of the two implemented controllers depends strongly on the traffic demand level. Under low-to-moderate demand, LST-Net generally provides favorable traffic efficiency, whereas the implemented MPC exhibits more consistent system-level performance as demand increases. The comparison between LST-Net-TD and LST-Net-WT further shows that the reward formulation substantially affects both traffic performance and the learned signal-allocation behavior. In particular, favorable in-network performance, such as lower waiting time or higher speed, does not necessarily translate into lower total time when depart delay is also considered. Across the distinct and unseen traffic scenarios, MPC maintains comparatively consistent total-time performance, while LST-Net-TD remains competitive in several scenarios but exhibits greater scenario dependence. Max-Pressure also performs effectively under some traffic

conditions, further indicating that no single control strategy is uniformly superior across all tested scenarios.

The conclusions of this study are limited to the simulated traffic conditions considered here and should not be interpreted as evidence of direct field readiness. Future work will extend the evaluation to field-calibrated traffic conditions and examine model mismatch, imperfect sensing and communication, and other operational disturbances that may affect different control approaches in different ways. Further investigation of hybrid control architectures that combine model-based prediction with learned traffic representations or policies may also provide a useful direction for improving both adaptability and robustness in distributed traffic signal control.

Author Contributions: Conceptualization, Z.H., X.G. and F.X.; methodology, Z.H., X.G. and F.X.; software, Z.H., G.M. and Y.C.; validation, Z.H., G.M. and Y.C.; formal analysis, Z.H.; investigation, Z.H., G.M. and Y.C.; data curation, Z.H., G.M. and Y.C.; writing—original draft preparation, Z.H.; writing—review and editing, Z.H.; visualization, Z.H.; supervision, Z.H., X.G. and F.X.; project administration, Z.H.; funding acquisition, Z.H., X.G. and F.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China under Grant 62303369, the Key Research and Development Program in Shaanxi Province of China (No. 2026SF-YBXM-776), Natural Science Basic Research Program of Shaanxi Province (No. 2026JC-YBMS-0715), the Scientific Research Plan Projects of Shaanxi Education Department under Grant 24JS025, and the Independent Research and Development project of State Key Laboratory of Green Building (No. LSZZ-Y202614).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Arman, A.; Badii, C.; Bellini, P.; Bilotta, S.; Nesi, P.; Paolucci, M. Analyzing Demand with Respect to Offer of Mobility. *Appl. Sci.* **2022**, *12*, 8982. [CrossRef]
2. Aboudolas, K.; Papageorgiou, M.; Kosmatopoulos, E. Store-and-Forward Based Methods for the Signal Control Problem in Large-Scale Congested Urban Road Networks. *Transp. Res. Part C Emerg. Technol.* **2009**, *17*, 163–174. [CrossRef]
3. Aboudolas, K.; Papageorgiou, M.; Kouvelas, A.; Kosmatopoulos, E. A Rolling-Horizon Quadratic-Programming Approach to the Signal Control Problem in Large-Scale Congested Urban Road Networks. *Transp. Res. Part C Emerg. Technol.* **2010**, *18*, 680–694. [CrossRef]
4. Le, T.; Vu, H.L.; Nazarathy, Y.; Vo, Q.B.; Hoogendoorn, S. Linear-Quadratic Model Predictive Control for Urban Traffic Networks. *Transp. Res. Part C Emerg. Technol.* **2013**, *36*, 498–512. [CrossRef]
5. Prashanth, L.A.; Bhatnagar, S. Reinforcement Learning With Function Approximation for Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 412–421. [CrossRef]
6. Fan, C.; Wang, F.; Zhou, B.; Ma, D. A Centralized Reinforcement Learning-Based Method for Traffic Signal Optimization Using an Adaptive Sequential Decision. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 13201–13216. [CrossRef]
7. Xie, X.F.; Smith, S.F.; Lu, L.; Barlow, G.J. Schedule-Driven Intersection Control. *Transp. Res. Part C Emerg. Technol.* **2012**, *24*, 168–189. [CrossRef]
8. Varaiya, P. Max Pressure Control of a Network of Signalized Intersections. *Transp. Res. Part C Emerg. Technol.* **2013**, *36*, 177–195. [CrossRef]
9. Mehrabipour, M.; Hajbabaie, A. A Cell-Based Distributed-Coordinated Approach for Network-Level Signal Timing Optimization. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 599–616. [CrossRef]
10. Hao, Z.; Boel, R.; Li, Z. Model Based Urban Traffic Control, Part II: Coordinated Model Predictive Controllers. *Transp. Res. Part C Emerg. Technol.* **2018**, *97*, 23–44. [CrossRef]
11. Hao, Z.; Boel, R. Convergence Analysis on Control for Traffic Signals in Urban Road Network. *Transp. Res. Part B Methodol.* **2022**, *165*, 35–62. [CrossRef]

12. Hao, Z.; Boel, R.K. Coordinated Model Predictive Controllers With Stabilizing Reference for Traffic Signals in Urban Road Network. *IEEE Access* **2024**, *12*, 121185–121197. [CrossRef]
13. Wu, N.; Li, D.; Xi, Y.; De Schutter, B. Distributed Event-Triggered Model Predictive Control for Urban Traffic Lights. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4975–4985. [CrossRef]
14. Pham, V.H.; Ahn, H.S. Distributed Stochastic MPC Traffic Signal Control for Urban Networks. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 8079–8096. [CrossRef]
15. Abbracciavento, F.; Zinnari, F.; Formentin, S.; Bianchessi, A.G.; Savaresi, S.M. Multi-Intersection Traffic Signal Control: A Decentralized MPC-Based Approach. *IFAC J. Syst. Control* **2023**, *23*, 100214. [CrossRef]
16. Luo, L.; Tan, C.; Tang, K.; Zhu, H. A Distributed Model Predictive Approach for Network Traffic Signal Control Using Multi-Objective Dynamic Programming. *Comput.-Aided Civ. Infrastruct. Eng.* **2025**, *40*, 3953–3978. [CrossRef]
17. Noaeen, M.; Naik, A.; Goodman, L.; Crebo, J.; Abrar, T.; Shakeri Hossein Abad, Z.; Bazzan, A.L.C.; Far, B.H. Reinforcement Learning in Urban Network Traffic Signal Control: A Systematic Literature Review. *Expert Syst. Appl.* **2022**, *199*, 116830. [CrossRef]
18. Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; Li, Z. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. In *Proceedings of the AAAI Conference on Artificial Intelligence*; Association for the Advancement of Artificial Intelligence: Washington, DC, USA, 2020; Volume 34, pp. 3414–3421. [CrossRef]
19. Li, Z.; Yu, H.; Zhang, G.; Dong, S.; Xu, C.Z. Network-Wide Traffic Signal Control Optimization Using a Multi-Agent Deep Reinforcement Learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103059. [CrossRef]
20. Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Gayah, V.; Xu, K.; Li, Z. 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*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1290–1298. [CrossRef]
21. Wei, H.; Xu, N.; Zhang, H.; Zheng, G.; Zang, X.; Chen, C.; Zhang, W.; Zhu, Y.; Xu, K.; Li, Z. CoLight: Learning Network-Level Cooperation for Traffic Signal Control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1913–1922. [CrossRef]
22. Lee, S.; Lyu, H.; Gong, Y.; Sun, Y.; Deng, C. MacLight: Multi-Scene Aggregation Convolutional Learning for Traffic Signal Control. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*; International Foundation for Autonomous Agents and Multiagent Systems: Richland, SC, USA, 2025; pp. 1263–1271. [CrossRef]
23. Wang, X.; Abdulhai, B.; Sanner, S. A Critical Review of Traffic Signal Control and a Novel Unified View of Reinforcement Learning and Model Predictive Control Approaches for Adaptive Traffic Signal Control. In *Handbook on Artificial Intelligence and Transport*; Dia, H., Ed.; Edward Elgar Publishing: Cheltenham, UK, 2023; Chapter 17; pp. 482–532. [CrossRef]
24. Imran, M.; Izzo, R.; Tortorelli, A.; Liberati, F. Comparison of Traffic Control with Model Predictive Control and Deep Reinforcement Learning. In *Proceedings of the 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*; IEEE: New York, NY, USA, 2023; pp. 989–994. [CrossRef]
25. Remmerswaal, W.; Sun, D.; Jamshidnejad, A.; De Schutter, B. Combined MPC and Reinforcement Learning for Traffic Signal Control in Urban Traffic Networks. In *Proceedings of the Proceedings of the 2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*; IEEE: New York, NY, USA, 2022; pp. 432–439. [CrossRef]
26. Guo, G.; Wang, Y. An Integrated MPC and Deep Reinforcement Learning Approach to Trams-Priority Active Signal Control. *Control Eng. Pract.* **2021**, *110*, 104758. [CrossRef]
27. Luo, L.; Tan, C.; Gao, Z.; Zhu, H.; D'Ariano, A.; Tang, K. SelectLight: Learning to Select Signal Plans Generated by Distributed Model Predictive Control for Urban Traffic Networks. *arXiv* **2026**, arXiv:2608.12256. [CrossRef]
28. Lee, H.; Han, Y.; Kim, Y. Reinforcement Learning for Traffic Signal Control: Incorporating a Virtual Mesoscopic Model for Depicting Oversaturated Traffic Conditions. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107005. [CrossRef]
29. Ingole, D.; Mariotte, G.; Leclercq, L. Perimeter Gating Control and Citywide Dynamic User Equilibrium: A Macroscopic Modeling Framework. *Transp. Res. Part C Emerg. Technol.* **2020**, *111*, 22–49. [CrossRef]
30. Poelman, M.C.; Hegyi, A.; Verbraeck, A.; van Lint, J.W.C. Structure-Free Model-Based Predictive Signal Control: A Sensitivity Analysis on a Corridor with Spillback. *Transp. Res. Part C Emerg. Technol.* **2023**, *153*, 104174. [CrossRef]
31. Yoon, J.; Ahn, K.; Park, J.; Yeo, H. Transferable Traffic Signal Control: Reinforcement Learning with Graph Centric State Representation. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103321. [CrossRef]
32. Müller, A.; Sabatelli, M. Bridging the Reality Gap of Reinforcement Learning Based Traffic Signal Control Using Domain Randomization and Meta Learning. In *Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*; IEEE: New York, NY, USA, 2023; pp. 5271–5278. [CrossRef]

33. Wang, Q.; Severino, J.; Sorensen, H.; Sanyal, J.; Ugirumurera, J.; Wang, C.; Berres, A.; Jones, W.; Kohls, A.; Venkatadurga, R.P.R. Deploying a Model Predictive Traffic Signal Control Algorithm—A Field Deployment Experiment Case Study. In *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*; IEEE: New York, NY, USA, 2022; pp. 3564–3570. [CrossRef]
34. Mercader, P.; Uwayid, W.; Haddad, J. Max-Pressure Traffic Controller Based on Travel Times: An Experimental Analysis. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 275–290. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.