sustainability

MDPI

Article

HydraLight: A Global-Context Spatio-Temporal Graph Transformer Framework for Scalable Multi-Agent Traffic Signal Control

Ahmed Dabbagh [1], Guray Yilmaz [1], Esra Calik Bayazit [2] and Ozgur Koray Sahingoz [1,*]

[1] Computer Engineering Department, Biruni University, 34015 Istanbul, Turkey; ahmeddabbagh004@gmail.com (A.D.); gurayy@biruni.edu.tr (G.Y.)
[2] Computer Engineering Department, Fatih Sultan Mehmet Vakif University, 34015 Istanbul, Turkey; ecalik@fsm.edu.tr
* Correspondence: osahingoz@biruni.edu.tr

Abstract

Urban traffic congestion presents a complex challenge driven by intricate spatial dependencies and non-stationary temporal dynamics. While Multi-Agent Deep Reinforcement Learning has shown promise for Traffic Signal Control, existing approaches often struggle with partial observability and fail to coordinate effectively across large-scale, heterogeneous road networks. In this paper, we propose HydraLight (HYbrid Deep Reinforcement Learning Architecture for Traffic Lights), a novel spatio-temporal framework that integrates Graph Attention Networks and Temporal Transformers. To overcome the localized myopia of standard graph methods, HydraLight introduces a Global Pooling Context module that broadcasts macroscopic, citywide traffic summaries, enabling agents to proactively mitigate systemic gridlock. Furthermore, to facilitate robust multi-scenario training, we introduce a Unified Prioritized Experience Replay (Unified PER) module that normalizes Temporal-Difference errors, preventing task dominance across diverse topologies. Extensive experiments on the RESCO benchmark across five synthetic and real-world networks demonstrate that HydraLight consistently outperforms state-of-the-art baselines (including X-Light and CoSLight).Byreducing traffic congestion, travel delays, and idle waiting times, the proposed framework also contributes to more sustainable urban mobility through improved traffic flow efficiency, lower fuel consumption, and reduced vehicular carbon emissions. Notably, the proposed architecture excels in structurally irregular environments, achieving up to 13.07% reduction in average travel time on complex arterial networks and consistently improving queue stability and waiting-time minimization across both synthetic and real-world RESCO benchmarks compared to state-of-the-art baselines.

Keywords: trafficsignal control; multi-agent reinforcement learning; multi-scenario learning; GAT; sustainable transportation; smart cities; urban sustainability; green traffic management; intelligent transportation systems

Academic Editor: Marilisa Botte

Received: 24 April 2026
Revised: 17 May 2026
Accepted: 20 May 2026
Published: 22 May 2026

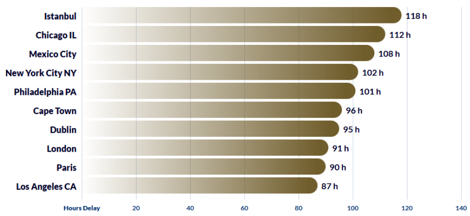
1. Introduction

The rapid acceleration of urbanization has led to a substantial increase in the number of vehicles, resulting in serious traffic congestion in major cities worldwide [1,2]. This has resulted not only in significant delays and economic losses but has also contributed to severe environmental pollution [3]. Therefore, the development of efficient Traffic Signal

Control (TSC) systems has become a key objective for urban planners and researchers to optimize traffic flow and promote sustainable transportation systems.

Figure 1 illustrates the impact and economic consequences of traffic congestion in major cities worldwide, emphasizing the importance of efficient TSC strategies. Figure 1a shows the top ten cities with the highest traffic delay times, many of which have high population density and vehicle ownership rates. These high delay times reflect inefficient traffic flow and inadequate TSC systems.

Additionally, Figure 1b shows the cost of congestion for some major cities across the US. These costs include hours lost per traveller, delay duration, and economic costs. The economic cost of congestion is huge, with billions of dollars spent each year [4]. These figures indicate that traffic congestion is not limited to the economy but also affects the environment, underscoring the need for intelligent TSC systems. Such systems help mitigate urban traffic congestion. Consequently, the development of cooperative TSC systems is critical for reducing congestion and minimizing travel delays.



(**a**) Ten Highest Traffic Delay Times By City

2025 Impact Rank (2024 Rank)	Urban Area	2025 Hours Lost (2024)	Delay Change	2025 Cost per Driver	2025 Cost per City	Downtown Speed (mph)
1 (2)	Chicago IL	112 (102)	10%	$2063	$7.5 B	9
2 (1)	New York City NY	102 (102)	0%	$1879	$9.7 B	11
3 (5)	Philadelphia PA	101 (77)	31%	$1860	$4.4 B	10
4 (3)	Los Angeles CA	87 (88)	-1%	$1602	$8.6 B	17
5 (4)	Boston MA	83 (79)	5%	$1529	$2.9 B	10
6 (6)	Miami FL	75 (74)	1%	$1381	$3.5 B	14
7 (8)	Atlanta GA	75 (65)	15%	$1381	$3.4 B	14
8 (7)	Houston TX	70 (66)	6%	$1289	$3.8 B	14
9 (9)	Washington DC	70 (62)	13%	$1289	$3.2 B	11
10 (10)	Seattle WA	68 (63)	8%	$1252	$1.9 B	14

(**b**) Congestion Costs in US Cities

Figure 1. Statistics about Traffic Congestion.

Traditionally, TSC systems have evolved from static, fixed-time control systems using various methods. For instance, the Webster method [5] uses historical data to determine fixed-time control. However, these methods are not adaptive to real-world traffic changes. Conversely, adaptive systems, such as Split Cycle Offset Optimization Technique (SCOOT) [6] and Sydney Coordinated Adaptive Traffic System (SCATS) [7], have been developed to respond to real-time traffic conditions. These systems frequently rely on simplified models and heuristic rules, which are inadequate for addressing complex, high-dimensional traffic problems and tend to perform effectively only under specific environmental conditions.

In the past few years, some machine learning algorithms, specifically Reinforcement Learning (RL), have emerged as promising solutions for TSC, enabling optimal control policies to be learned through interaction with the environment without modeling the underlying traffic dynamics [8]. Early work in RL used tabular Q-learning to control individual intersections. However, with the development of Deep Learning, Deep Reinforcement Learning techniques such as the Deep Q-Network (DQN) have shown promise in handling the high-dimensional state spaces that characterize current traffic networks [9]. Some notable works include the application of Multi-Agent Reinforcement Learning (MARL), in which an agent is specified as a controller for traffic signals at an intersection, responsible for controlling the phase at that location. These agents may be able to communicate with one another, take local information, decide on actions (for example, switching the phase or setting the duration), and receive rewards focused on reducing delay or maximizing throughput.

Managing traffic in a big city is a major challenge because what happens at one intersection directly affects the others. If a light stays red too long at one corner, it quickly causes a backup at the next. Standard AI methods often struggle with this because traffic doesn't just move in a simple, straight line; it flows through a complex, web-like network that changes constantly throughout the day. To solve this, researchers are now using a specialized technology called Graph Attention Networks (GATs) [10], which can be thought of as giving the Artificial Intelligence (AI) the ability to "look around" and prioritize which neighboring intersections are most important at any given moment. By combining this with Reinforcement Learning, the system can better understand the city's layout and adjust signal timing in real time to keep traffic moving smoothly.

Current TSC systems use advanced artificial intelligence to manage city flow, but generally fall into two categories, each with its own "blind spot" [11]. The first group, known as graph-structured models [10], views a city's road network as a web in which intersections are nodes and roads are the connecting lines. These models are excellent at understanding spatial relationships, basically, how a backup at one light immediately affects the next. However, they struggle with the "big picture" of time. They often look at traffic in very short bursts, failing to account for how congestion might slowly build up or spread across several signal cycles over time.

The second category consists of sequence-based models [12], which are specifically designed to track how traffic patterns evolve over long periods, making them highly effective in predicting long-term changes. The disadvantage here is that they often treat each intersection as if it existed in an isolated environment. By focusing so much on the timeline of a single spot, they often overlook the actual physical layout of the traffic network, unless they use highly specialized tools such as those found in X-Light models [13].

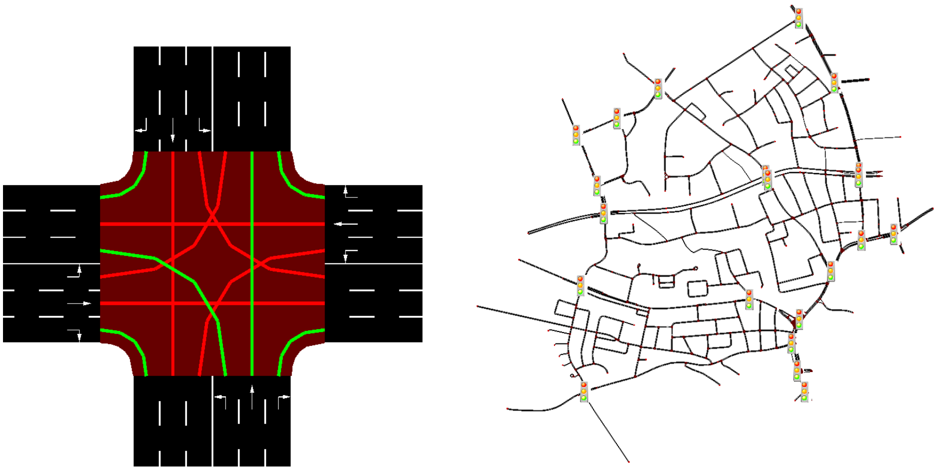
The management of complex traffic networks is often addressed using MARL. In this paradigm, each traffic signal is represented by an autonomous agent that learns control policies through interactions with the environment and receives rewards based on improvements in traffic conditions, such as reduced delays. Because every intersection is different (some are three-way, some are four-way), MARL requires a way to standardize these different "states" and "actions" so the AI can understand them uniformly. Even in the most recent MARL designs [13,14], there remains a lingering problem in how they model the shape of the city. Although they use certain manual rules to help the AI adapt quickly to new locations, these rules are often too rigid. They don't capture the fine, intricate details of how a specific road network actually functions. Achieving optimal efficiency requires future models to move beyond rigid, pre-defined rules and instead learn the structural topology of road networks directly from observed traffic data [15].

1.1. Problem Definition

Existing TSC methods often optimize either spatial or temporal structure in isolation, which creates coordination failures in dense and heterogeneous networks [9,16]. The core problem addressed in this paper is to learn a single cooperative control policy that simultaneously captures: (i) local lane-level dynamics, (ii) network topology constraints, and (iii) long-horizon temporal effects under partial observability. We formulate this as a cooperative multi-agent over a traffic graph and train one shared policy across multiple synthetic and real-world maps.

To address these challenges, we introduce **HydraLight** (HYbrid Deep Reinforcement Learning Architecture for Traffic Lights), a scalable and coordinated framework designed for city-wide TSC. HydraLight bridges existing modeling gaps in spatial and temporal dynamics while ensuring robust generalization across heterogeneous traffic scenarios.

Figure 2 provides a concrete network-level view of the control setting, emphasizing that learning and action selection are performed over an entire traffic-light graph.



(**a**) Intersection geometry (**b**) Representative network-wide traffic map

Figure 2. Intersection geometry and representative network-wide traffic map used for HydraLight.

1.2. Contributions

Our primary contributions are summarized as follows:

- **All-in-One Space and Time Modeling:** This system acts like a single "brain" for the whole city, combining a digital map of road layouts (via GAT) with a memory of traffic patterns over time (via Transformer) to keep cars moving smoothly across all intersections at once.
- **Data-Driven Heterogeneous Alignment:** The system uses a topology-independent observation representation with zero-padding and shared embedding layers, enabling automatic adaptation to heterogeneous intersections with different lane counts, traffic movements, and phase configurations (e.g., three-way and four-way intersections) without manually designed rules. This ensures it works anywhere without requiring a human to set up rules for every new street or intersection manually.
- **Big-Picture Information Sharing:** This module gives each traffic light a "big-picture" summary of the entire city's traffic, represented as a broadcasted numerical vector that captures the overall mean density and the most severe bottleneck in the network. This helps agents distinguish between local queues and systemic gridlock so they can work together proactively.
- **Robust Multi-Scenario Training:** This system uses a smart learning strategy (Unified PER) that focuses on the most structurally important and surprising traffic moments for example, prioritizing unexpectedly high learning errors at a complex real-world intersection rather than just standard heavy traffic on a simple grid. This helps the AI stay steady and reliable across a wide range of unpredictable road conditions.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the formal problem formulation. Section 4 details the HydraLight methodology. Section 5 presents the experimental setup. Section 6 reports the experimental results and analysis. Section 7 discusses mechanisms and limitations. Finally, Section 8 concludes the paper and outlines future directions.

https://doi.org/10.3390/su18115252

2. Related Work

TSC systems have gradually evolved from traditional fixed-time and rule-based strategies into more intelligent, learning-driven, cooperative, and topology-aware frameworks that can better adapt to the complexity of modern urban traffic networks. In general, the existing literature can be grouped into three major categories. The first category, discussed in *Traditional Approaches*, focuses on classical adaptive and heuristic-based traffic control systems. The second category, presented in *Gap Analysis and Differentiation from State-of-the-Art*, examines the limitations of current DRL-, graph-, and Transformer-based methods in handling large-scale and heterogeneous traffic environments. Finally, *Comparison with State-of-the-Art Works* provides a systematic comparison between representative TSC approaches and the proposed HydraLight framework in terms of scalability, coordination capability, temporal modeling, and heterogeneous network support.

2.1. Traditional Approaches

Traditional adaptive systems, such as SCOOT [6] and SCATS [7], are widely deployed in real-world environments due to their robustness, interpretability, and relatively low computational requirements. SCOOT operates using a centralized architecture that continuously collects traffic data from inductive loop detectors and incrementally optimizes signal parameters—such as cycle time, green splits, and offsets—based on short-term traffic predictions. It performs frequent, small adjustments to maintain network-wide coordination, particularly in urban corridors. In contrast, SCATS (Sydney Coordinated Adaptive Traffic System) adopts a more decentralized and hierarchical control strategy, where local controllers select from a library of pre-defined signal timing plans based on real-time traffic flow, occupancy, and degree of saturation, while also enabling coordination across regions through adaptive linking mechanisms. Despite their widespread adoption and operational success, both systems fundamentally rely on heuristic rules and simplified traffic flow assumptions, which limits their scalability and adaptability in highly dynamic, non-stationary, and heterogeneous traffic environments. In particular, their ability to capture complex multi-intersection dependencies and long-term temporal dynamics remains constrained. On the other hand, pressure-based control methods, especially max-pressure control, have demonstrated strong theoretical guarantees for network stability by minimizing queue imbalances at intersections [17]. However, such approaches are inherently myopic, lacking mechanisms to model long-term temporal dependencies and flexible inter-agent coordination, which are increasingly critical for next-generation TSC systems.

The use of Deep Reinforcement Learning (DRL) signaled an important shift toward learning control policies directly from high-dimensional traffic observations. The initial DRL-based TSC models have shown that phase selection strategies can be optimized to improve performance over traditional fixed-time and actuated control methods in simulated scenarios [18]. A conceptual advancement toward TSC was proposed by the FRAP model, which treated the phase selection process as a competition among conflicting traffic movements [19]. These approaches are beneficial for enhancing sample efficiency and generalization capability by leveraging invariance structures. However, they are mostly focused on single-intersection decision structures or weakly coupled multiple-intersection settings.

Beyond single-intersection DRL, a series of works extended the scope to network-wide coordination and multi-scenario generalization. PressLight [20] integrated the max-pressure signal directly into a DRL reward structure, enabling effective network-wide queue management without requiring explicit neighbor communication. CoLight [10] modeled the road network as a graph and applied GAT to let each intersection aggregate weighted information from its physical neighbors, yielding cooperative phase decisions. DynST-

GAT [21] extended this by constructing a dynamic spatio-temporal graph attention network that simultaneously captures spatial topology and short-horizon temporal dependencies, making it more responsive to abrupt and transient demand shifts. AttendLight [22] introduced a phase-level attention mechanism that explicitly handles heterogeneous intersection geometries, providing a flexible representation for intersections with varying numbers of lanes and movement phases.

On the generalization front, GeneraLight [23] proposed clustering-based meta-learning over phase structures to share policies across diverse intersection layouts, demonstrating zero-shot transfer to previously unseen environments. MetaLight [24] framed multi-scenario TSC as a meta-reinforcement learning problem, enabling rapid policy adaptation to new traffic contexts with minimal additional training. X-Light [13] extended the transfer learning paradigm by combining Transformer-based temporal attention with large-scale cross-city pretraining, allowing agents to capture long-range temporal patterns and generalize across heterogeneous urban networks. CoSLight [25] took a different route by jointly optimizing the control policy and a dynamic communication graph, allowing agents to adaptively select communication partners based on the current traffic state rather than relying on a fixed physical topology.

2.2. Gap Analysis and Differentiation from the State of the Art

While numerous approaches have applied DRL to TSC, the closest competitors to HydraLight fall into three distinct paradigms. We explicitly compare our framework against these leading methods to define the novelty boundary:

1. Graph-Based Spatial Cooperation (e.g., CoLight) is one of the cooperative TSC approaches; the traffic network is considered as a graph, with intersections being the nodes and the roads being the edges. Interactions among intersections are considered, and intersections can make decisions based on information from neighboring intersections. Spatial dependencies are modeled using graph neural networks, such as GAT. This allows the intersections' agent to make decisions based not only on the current state of the traffic but also based on the state of the surrounding network.

Nevertheless, CoLight faces the challenge of efficiently disseminating congestion information across the network by relying solely on message passing through neighbors. This challenge is referred to as the local myopia problem. This is mainly because the model lacks a global view. This means that an intersection in the middle of the grid cannot respond proactively to congestion on the outskirts before it reaches its neighbors. To address local myopia, the HydraLight model proposes using the Global Pooling Context, which sends overall citywide traffic summary information to all agents. This allows each agent to have a global view and distinguish between congestion and gridlock.

2. Transformer-Based Transfer Learning (e.g., X-Light) in cooperative TSC leverages Transformer architectures to model long-range spatial and temporal dependencies across traffic networks. Unlike traditional graph-based methods that rely primarily on local neighborhood aggregation, Transformer-based models use self-attention mechanisms to capture global interactions between intersections, enabling each agent to consider distant but relevant traffic patterns [26].

Through transfer learning, a model trained in one traffic environment (e.g., a specific city or network topology) can be adapted to a new environment with little to no additional training data. For example, X-Light-style approaches pretrain a Transformer on large-scale traffic scenarios to learn generalizable representations of traffic dynamics, and then directly apply it to new cities or intersection layouts zero-shot. This significantly reduces training time, improves sample efficiency, and enhances scalability across heterogeneous urban networks.

https://doi.org/10.3390/su18115252

While powerful in modeling temporal dynamics, these methods often under-utilize the explicit graph structure of the road network. They treat intersections as sequences or as a fully connected set, potentially overlooking hard connectivity constraints such as upstream and downstream flow relationships. Furthermore, they typically rely on standard PPO or DQN training strategies, which assume all training samples are equally important. To address these limitations, HydraLight synergizes the explicit spatial structure of GAT with the temporal memory capabilities of Transformers. In addition, it replaces the conventional training loop with Unified PER, which normalizes learning priorities according to scenario difficulty and prevents simpler transfer-learning scenarios from dominating gradient updates.

3. Dynamic coordination mechanisms such as CoSLight aim to improve cooperation within the multi-agent traffic control system by simultaneously optimizing the control policy and the communication structure. With this approach, the agent not only determines how to adjust the traffic lights optimally but also which agent to communicate with. This provides the advantage of adaptive information sharing based on the current traffic conditions.

Recent studies in intelligent transportation systems have increasingly focused not only on traffic efficiency but also on traffic safety, collision-risk prediction, and trajectory-aware traffic modeling [27,28]. In particular, recent works have investigated trajectory analysis and traffic-flow stability under dynamic and adversarial environments, emphasizing the importance of robust temporal reasoning and global coordination mechanisms in connected transportation systems. Furthermore, emerging research on cyber-physical traffic stability and trajectory-sensitive control has shown that localized decision-making alone is insufficient for maintaining stable large-scale traffic behavior under heterogeneous and non-stationary conditions. However, despite these advances, existing TSC frameworks still suffer from three major limitations: (i) inadequate integration of long-term temporal dependencies with explicit graph-based spatial coordination, (ii) limited capability to capture network-wide congestion propagation and systemic gridlock formation, and (iii) insufficient robustness and balanced learning across heterogeneous traffic scenarios. HydraLight addresses these gaps through a unified spatio-temporal architecture combining Graph Attention Networks, Temporal Transformers, Global Context Pooling, and Unified Prioritized Experience Replay within a scalable multi-agent reinforcement learning framework.

2.3. Comparison with State-of-the-Art Works

The main disadvantage of this approach is that it introduces significant complexity to the model by requiring an additional policy for neighbor selection. It is not very efficient in terms of samples and is more complicated to train. It mainly focuses on learning which agent to communicate with, rather than on learning long-term temporal dependencies. Additionally, there is no mechanism to address extreme reward-scaling issues between synthetic grid worlds and real-world traffic maps. On the other hand, HydraLight uses a fixed yet globally informed structure that includes physical neighbors and a Global Context module. This makes the model simpler and more efficient while achieving 13% better performance on arterial networks by focusing on temporal memory with the Temporal Transformer and on fairness with Unified PER, rather than on neighbor selection. The comparison of the proposed system is depicted in Table 1.

https://doi.org/10.3390/su18115252

Table 1. Comparison of representative TSC approaches and HydraLight.

Method	Spatial Modeling	Temporal Modeling (Transformer)	Network-Wide Coordination	Heterogeneous Intersection Support	Generalization Focus	Scenario
SCOOT [6]	✗	✗	Partial	✗	✗	Single
SCATS [7]	✗	✗	Partial	✗	✗	Single
Max-Pressure [17]	✗	✗	✓ (local)	Partial	✗	Single
FRAP [19]	✗	✗	Partial	Partial	Partial	Single
PressLight [20]	✗	✗	✓	Partial	Partial	Single
CoLight [10]	✓ (GAT)	✗	✓	Partial	✗	Single
GeneraLight [23]	✗	✗	Partial	✗	✓	Single
DynSTGAT [21]	✓ (Spatio-Temporal GAT)	Partial	✓	Partial	✗	Single
AttendLight [22]	✗	Partial	Limited	✓	Partial	Single
X-Light [13]	Partial	✓	✓	Partial	✓	Multi
MetaLight [24]	Partial	✗	Partial	Partial	✓	Multi
CoSLight [25]	Partial	✗	✓	Partial	Partial	Multi
HydraLight (Ours)	✓ (GAT)	✓ (Transformer)	✓	✓	✓	Multi

3. Problem Formulation

Network-wide Traffic Signal Control (TSC) is defined as the problem of jointly optimizing the signal phases and timings of all intersections within a traffic network to improve global traffic efficiency. Formally, it can be modeled as a sequential decision-making problem on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in V$ represents a signalized intersection and each edge $e \in \mathcal{E}$ represents a road segment connecting intersections.

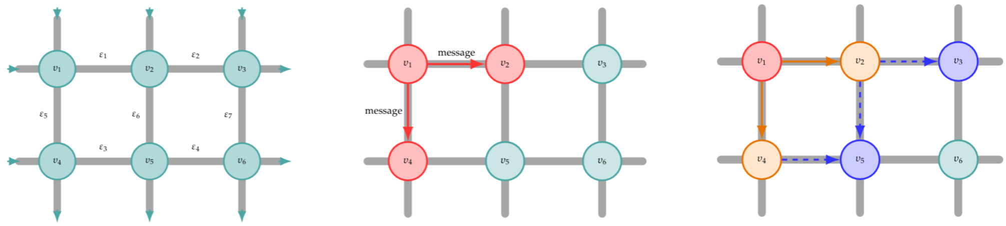
At each time step t, a control policy determines the action $a_t^{(i)}$ (e.g., phase selection or duration) for every intersection based on partial observations of the traffic state. The objective is to optimize network-level performance metrics such as minimizing total delay, queue length, or average travel time. The core challenge of network-wide TSC lies in simultaneously capturing spatial dependencies among intersections, temporal dynamics of traffic flow, and coordinated decision-making under partial observability and non-stationary traffic conditions.

3.1. Network Topology and Adjacency

Topology of a network is defined as a way of describing how the traffic network looks like, by showing the nodes and links that connect the network. The adjacency of nodes is determined by their connectivity to one another, and thus how they can affect each other. The traffic network is defined as a graph where

- **Nodes ($\mathcal{V}$):** represent the set of N signalized intersections.
- **Edges ($\mathcal{E}$):** represent the road links connecting them. An edge $e_{ji} \in \mathcal{E}$ exists if and only if a road segment connects the outflow of intersection j directly to the inflow of intersection i. This adjacency defines the set of communicable neighbors $\mathcal{N}_i = j \mid e_{ji} \in \mathcal{E}$ for the Graph Attention mechanism.

For a visual example of this graph abstraction (nodes, edges, and flow directions), including one-hop and two-hop GNN communication, see Figure 3. In particular, Figure 3a shows the 2×3 network topology used for notation, Figure 3b illustrates one-hop message passing, and Figure 3c illustrates two-hop information propagation.



(**a**) 2×3 traffic graph topology (**b**) 1st GAT layer: one-hop communication from v_1 (**c**) 2nd GAT layer: two-hop communication from v_1

Figure 3. Extended graph explanation for the traffic network and GNN communication. (**a**) A 2×3 intersection graph with node set $\mathcal{V} = v_i$ and edge set $\mathcal{E} = \varepsilon_j$. (**b**) In the first GAT layer, v_1 exchanges messages with one-hop neighbors. (**c**) In the second GAT layer, information propagates to two-hop neighbors through intermediate nodes, illustrating multi-hop coordination between agents.

To formalize the components of the traffic control system, we first introduce the following fundamental definitions:

An intersection $i \in \mathcal{V}$ is a junction where multiple roads connect, governed by a traffic light. A standard intersection, as shown in Figure 2a, typically consists of entrance arms (e.g., North, South, East, West), each accommodating specific traffic movements such as left-turn, through, and right-turn lanes.

A traffic phase is a set of permissible traffic movements that can proceed simultaneously without spatial conflict.

The multi-hop communication mechanism of the Graph Attention Network enables the extraction of hierarchical spatial dependencies between intersections. In the first GAT layer, one-hop message passing captures direct local interactions such as immediate queue spillback and neighboring traffic pressure. In the second GAT layer, information propagates beyond adjacent nodes, allowing intersections to indirectly perceive upstream and downstream congestion trends through intermediate intersections. This hierarchical aggregation process transforms raw topological connectivity into learned spatial traffic representations, enabling the model to capture both localized and network-wide traffic propagation dynamics. Consequently, the generated node embeddings encode not only the current state of a single intersection but also the broader spatial context of the surrounding urban traffic network.

3.2. Partially Observable Markov Decision Process Formulation

While intersections utilize local sensors and cannot observe the exact global state of the entire road network (e.g., exact vehicle positions on distant links), traffic signal control is fundamentally a sequential decision-making problem. Assuming there are N signalized intersections in a scenario, each intersection acts as an independent agent operating under partial observability. Therefore, we formulate the learning objective for each agent as a Partially Observable Markov Decision Process defined by the tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. To achieve cooperative network-wide control, HydraLight employs a single centralized Spatio-Temporal Q-Network with parameter sharing across all intersections. Although each intersection acts as an agent with its own local observation, all actions are jointly generated by the same centralized controller rather than by independently trained policies. The Partially Observable Markov Decision Process components mapping to our traffic environment are defined as follows:

- **States ($\mathcal{S}$):** The true global state of the traffic network $s_t \in \mathcal{S}$ at time step t.
- **Observations ($\mathcal{O}$):** The local, partial observation $o_t^{(i)} \in \mathcal{O}$ gathered by intersection i's sensors prior to action selection.
- **Actions ($\mathcal{A}$):** The available phase selections $a_t^{(i)} \in 1, \ldots, |\Phi_i|$ for the intersections.

- **Transition Model ($\mathcal{P}$):** The underlying traffic dynamics of the simulator moving from state to state based on the chosen signal phases.
- **Rewards ($\mathcal{R}$):** The local immediate reward $r_t^{(i)}$ for intersection i, structured as a combination of delay and throughput to ensure stability.
- **Discount Factor (γ):** A discount factor $\gamma \in [0, 1)$ (specifically, $\gamma = 0.95$) used to balance immediate and future rewards.

System State ($\mathcal{S}$) and Observation Space ($\mathcal{O}$): At any time step t, the global traffic environment is in a true state $s_t \in \mathcal{S}$, encapsulating the dynamic information of the entire network. Each intersection i only captures a local, partial observation $o_t^{(i)} \in \mathcal{O}$ generated by its specific observation mapping function $\mathcal{Z}^{(i)}(s_t)$, which translates the global state to a local view before action selection. Rather than acting independently, the local observations from all N intersections are aggregated into a joint observation $\mathbf{o}_t = \langle o_t^{(1)}, \ldots, o_t^{(N)} \rangle$. A single centralized Q-network processes this joint observation simultaneously. By leveraging the GAT layer, the centralized controller facilitates spatial information sharing across the traffic graph topology, ultimately outputting Q-values for all N intersections in a single forward pass, from which the joint action $\mathbf{a}_t$ is derived. This design corresponds to centralized coordinated control with shared policy execution, where the network jointly reasons over the complete traffic graph while still preserving intersection-level observations.

Action Space ($\mathcal{A}$): We adopt a discrete **Phase Selection** control scheme, as illustrated in Figure 4. Rather than individual agents acting independently, the centralized controller selects an action $a_t^{(i)} \in 1, \ldots, |\Phi_i|$ for each intersection i for the next control interval Δt, corresponding to the phase to be activated. As shown in Figure 4b, the available phases Φ_i are constructed by grouping non-conflicting traffic movements (Figure 4a).

- Transition Constraint: If $a_t^{(i)} \neq a_{t-1}^{(i)}$, a yellow transition phase of fixed duration t_{yellow} is enforced before the new phase begins, ensuring safety compliance.
- Heterogeneity: The size of the action space $|\Phi_i|$ varies by intersection geometry (e.g., 3-way vs. 4-way). We handle this via action masking in the final policy head.

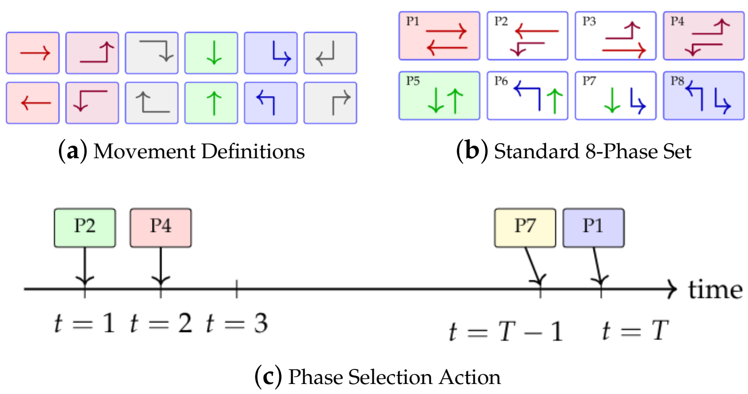


Figure 4. Intersection and Action Space Definition. (**a**) The 12 vehicular movements. (**b**) The set of 8 admissible signal phases Φ_i used in our action space, where non-conflicting movements are grouped. (**c**) The agent selects a phase $p \in \Phi_i$ at each step t; if $p \neq p_{t-1}$, a yellow transition period is enforced [25].

Reward Function ($\mathcal{R}$) and Policy (π): The overarching objective is to minimize network-wide travel delay. To evaluate performance at each node, we define the local immediate reward $r_t^{(i)}$ for intersection i as a combination of delay and throughput to ensure stability:

$$r_t^{(i)} = V_t^{(i)} - W_t^{(i)} - Q_t^{(i)} \tag{1}$$

Equation (1) defines the local reward at intersection i, where $r_t^{(i)}$ is the reward at time step t, $V_t^{(i)}$ is mean speed, $W_t^{(i)}$ is waiting time, and $Q_t^{(i)}$ is queue length, and it is used as the training signal stored in the replay buffer for Q-network optimization. To ensure stability across scenarios with different traffic densities, eachmetric is dynamically normalized using an online adaptive standardization technique detailed in Section 4.5 yielding the normalized from in Equation (16).

The centralized controller defines an implicit joint policy $\pi(\mathbf{a}_t \mid \mathbf{o}_t)$ derived from a shared Q-network $Q_\theta(\mathbf{o}_t)$. During training, actions are selected via an ε-greedy strategy; at deployment, the policy is deterministic:

$$a_t^{(i)} = \arg\max_a Q_\theta^{(i)}(\mathbf{o}_t, a) \quad (2)$$

Equation (2) defines the greedy action selection, where $a_t^{(i)}$ is the selected phase index for intersection i at time t, $Q_\theta^{(i)}(\mathbf{o}_t, a)$ is the predicted action-value under parameters θ, and $\mathbf{o}_t$ is the joint observation, and it is important for deployment-time phase selection across the entire traffic network. The overarching objective is to maximize the global expected discounted return across the network. Since the Q-network outputs individual action values for each intersection i, we first define the local discounted return as:

$$G_t^{(i)} = \sum_{k=0}^{\infty} \gamma^k r_{t+k}^{(i)} \quad (3)$$

Equation (3) defines the local discounted return, where $G_t^{(i)}$ is the return for intersection i from time t, $r_{t+k}^{(i)}$ is the reward k steps ahead, $\gamma \in [0, 1)$ is the discount factor (set to 0.95), and k is the future-step index, and it is used in Equation (4) to form the network-wide optimization objective.

$$G_t = \sum_{i=1}^{N} G_t^{(i)} \quad (4)$$

Equation (4) defines the network-wide objective, where G_t is the global return, $G_t^{(i)}$ is the local return of intersection i, and N is the number of signalized intersections, and it severs as the global learning objective that the shared Q-network is trained to maximize.

The reward function was deliberately formulated as a simple unweighted linear combination in order to preserve interpretability, reduce unnecessary complexity, and support stable optimization across different traffic environments. Because the three traffic-related indicators, namely average speed, waiting time, and queue length, inherently exist on different numerical ranges, assigning fixed manual weights could introduce imbalance in the reward structure and negatively affect the stability of the learning process under varying traffic conditions. To address this issue, HydraLight applies an adaptive online normalization strategy to dynamically standardize each metric during training. In this way, all reward components can contribute more consistently and fairly to the optimization process, while still maintaining the intuitive objective of improving traffic flow efficiency and reducing congestion-related delays.

4. Methodology

The methodology section is crucial to this research for the practical implementation of the theoretical advancements made by HydraLight in an effective learning framework. Due to the complexity of the problem of cooperative traffic signal control which encompasses issues of high dimensionality, partial observation, and strong coupling of spatial and temporal dependencies, it is imperative to adopt a meticulous design that ensures scalability

https://doi.org/10.3390/su18115252

and generalization capabilities in order to effectively solve the problem over heterogeneous traffic networks. With that in mind, the methodology chapter is devised to provide a holistic approach toward solving the problem at hand using graph-based spatial inference combined with sequence-based temporal modeling under a multi-agent reinforcement learning framework. Furthermore, it proposes new techniques such as global context propagation and scenario-balanced training to overcome some of the most common pitfalls of conventional solutions, including local myopia and task domination.

4.1. Framework Overview

We propose HydraLight (HYbrid Deep Reinforcement Learning Architecture for Traffic Lights), a cooperative multi-agent framework designed for scalable network-wide signal control. The high-level architecture of the system is illustrated in Figure 5. Unlike fully centralized methods that suffer from the curse of dimensionality, or independent methods that lack coordination [29], HydraLight trains a single Spatio-Temporal Q-Network using experience from all intersections. This enables efficient knowledge transfer and stable learning across heterogeneous topologies [13].

During execution, all intersections share a single centralized policy network with shared parameters. Each traffic light contributes its local observations to the shared controller, which jointly computes coordinated actions for the entire network in a single forward pass. It constructs a decision based on its local observation history, messages aggregated from direct neighbors, and a broadcast summary of the city-wide traffic state. The network integrates two distinct inductive biases to handle the specific complexities of traffic dynamics:

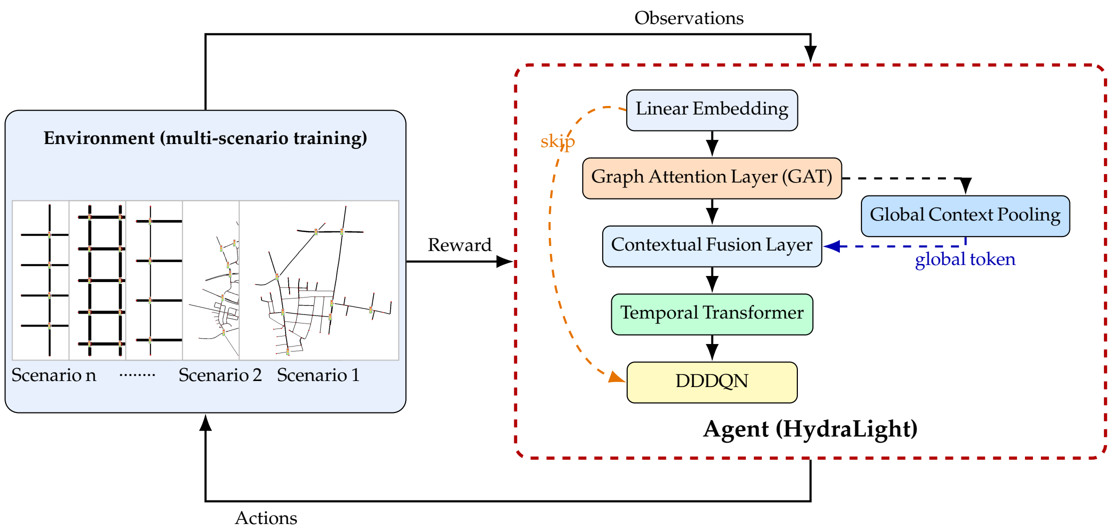


Figure 5. HydraLight control loop and internal agent architecture. The left panel shows multi-scenario training with five traffic maps and a HydraLight agent pipeline, with interaction signals restricted to Observations, Reward, and Actions. The right panel preserves the detailed module-level architecture view.

1. Spatial Dependency: A GAT [30] explicitly models the non-linear propagation of congestion between connected intersections, dynamically weighting neighbor importance as seen in works like CoLight [10] and DynSTGAT [21].
2. Temporal Dependency: A Temporal Transformer processes a history of observations to mitigate the partial observability inherent in snapshot-based traffic states, following recent trends in meta-learning for TSC [13].

HydraLight utilizes Double Dueling Deep Q-Network (DDDQN) [31,32], stabilized by a novel Unified PER mechanism to facilitate balanced multi-scenario training. This architecture is further enhanced by a novel Unified Prioritized Experience Replay mechanism, which normalizes Temporal-Difference errors across heterogeneous traffic scenarios, thereby mitigating task dominance and ensuring balanced, difficulty-aware sampling during multi-scenario training.

The multi-scenario training environment was constructed using five heterogeneous traffic networks from the RESCO benchmark, incorporating both synthetic and real-world urban layouts to improve the generalization capability of the proposed framework. The synthetic scenarios consist of Grid 4×4, Grid 5×5, and Arterial 4×4 networks, whereas the real-world environments include Ingolstadt21 and Cologne8. Together, these scenarios represent a diverse set of traffic topologies, ranging from structured grid-based intersections and arterial road corridors to more irregular and complex urban road configurations. Depending on the scenario, the number of controlled intersections varies between 8 and 21. To further increase environmental diversity, traffic demand patterns were generated under both low-density and highly congested conditions using randomized vehicle injection rates and dynamically changing route distributions. Moreover, the training process intentionally included challenging saturation-level traffic situations with highly uneven flow distributions in order to evaluate the robustness, scalability, and adaptability of HydraLight under dynamic and non-stationary traffic conditions.

4.2. Observation Space and Temporal Windowing

In a traffic signal control setting for multiple agents, an efficient observation space needs to accommodate the different types of intersections and help learn policies effectively. Due to partial observability issues, using only immediate information from the local environment is inadequate to understand the changing trends in the traffic situation. For this purpose, we have developed a single, topology-independent observation representation scheme that incorporates lane-level and phase-level information and augmented it using observation windows over time.

For traffic signal control by multiple agents, an observation space which allows learning a robust policy needs to account for different intersection topologies and provide heterogeneous observations. Because of the inherent limitations in terms of partial observability, it is necessary to look beyond immediate local observations in order to understand the changing traffic trends. To achieve this, we create a common representation based on features observed from lanes and traffic phases along with time windowed representations of past observations. To enable topology-agnostic learning, we construct a unified feature vector for each intersection. For intersection i at time t, the instantaneous observation vector $o_{t,\text{raw}}^{(i)}$ is the direct input representation before the first linear embedding layer in Figure 5. It consists of lane-level and phase-level features as follows:

1. Queue Length (lane-level, $o_{\text{queue}} \in \mathbb{R}^{|L_i|}$): The number of halting vehicles on each incoming lane $l \in L_i$. We normalize this value by the lane's physical capacity, mapping it to the range $[0, 1]$.
2. Lane Occupancy (lane-level, $o_{\text{occ}} \in \mathbb{R}^{|L_i|}$): The fraction of each incoming lane currently covered by vehicles.
3. Traffic Flow (lane-level, $o_{\text{flow}} \in \mathbb{R}^{|L_i|}$): The estimated flow rate on incoming lanes based on vehicle density and mean speed, normalized by a theoretical maximum flow $q_{\max}$.
4. Current Phase (phase-level, $o_{\text{phase}} \in 0, 1^{|\Phi_i|}$): A one-hot encoded vector representing the currently active green phase.

5. Elapsed Time (phase-level, $o_{\text{duration}} \in \mathbb{R}^1$): The time t_{elapsed} since the last phase change, normalized by $T_{\text{hor}} = 100\text{s}$ and clipped to unity: $\min(t_{\text{elapsed}}/T_{\text{hor}}, 1)$.
6. Last Action (phase-level, $o_{\text{last}} \in 0, 1^{|\Phi_i|}$): A one-hot encoding of the action applied at step $t-1$, providing explicit auto-regressive memory.

Thus, lane-level features are o_{queue}, o_{occ}, o_{flow} and phase-level features are o_{phase}, o_{duration}, o_{last}. For a 4-arm intersection with 8 incoming lanes and 4 candidate phases, the unpadded observation is depicted in Equation (5):

$$o_{t,\text{raw}}^{(i)} = [\underbrace{q_1, \ldots, q_8}_{\text{queue}}, \underbrace{u_1, \ldots, u_8}_{\text{occupancy}}, \underbrace{f_1, \ldots, f_8}_{\text{flow}}, \underbrace{p_1, \ldots, p_4}_{\text{current phase one-hot}}, \underbrace{\tau}_{\text{elapsed time}}, \underbrace{a_1, \ldots, a_4}_{\text{last action one-hot}}], \quad (5)$$

Equation (5) illustrates the raw observation vector for a standard 4-arm instersection with total dimension $8+8+8+4+1+4 = 33$, and it is used as the direct input to the Linear Embeddign Layer (Figure 5) before spatial and temporal processing. Crucially, $o_{t,\text{raw}}^{(i)}$ contains *only* local data: information from neighboring intersections $j \in \mathcal{N}_i$ is *not* concatenated to this vector and is instead aggregated later via the GAT mechanism described in Section 4.3.1. To support cross-scenario learning under heterogeneous state spaces (e.g., different lane and phase counts), we then standardize this same input pipeline by zero-padding $o_t^{(i)}$ to a fixed size of 64 and passing it through a shared linear projection layer before the GAT and Temporal Transformer modules, yielding a unified embedding space regardless of topology.

To handle heterogeneous intersections with varying numbers of lanes and traffic signal phases, lane-level and phase-level observations are first transformed into a fixed-dimensional representation using zero-padding. This design enables the model to standardize input dimensions across different intersection layouts before projecting them into a shared latent embedding space. As a result, the proposed Graph Transformer architecture can process diverse urban traffic configurations in a unified and scalable manner without requiring intersection-specific model modifications.

The temporal history length H was determined empirically by considering the trade-off between capturing long-term traffic dependencies and maintaining computational efficiency. During the preliminary analysis phase, several different window sizes ($H = 4, 6, 8, 10$, and 12) were evaluated on both synthetic and real-world traffic scenarios. The experimental observations showed that smaller temporal windows were generally insufficient for modeling delayed congestion propagation and evolving traffic-wave patterns across intersections. In contrast, larger window sizes increased computational complexity and memory overhead while contributing only marginal performance improvements due to the inclusion of redundant temporal information. Based on these findings, $H = 10$ was selected as the most balanced configuration, as it consistently achieved stable convergence behavior and strong generalization performance across heterogeneous traffic environments.

4.3. Spatio-Temporal Q-Network Architecture

A proper control scheme in complicated traffic settings should jointly consider the relationships between different intersections as well as the evolving traffic processes. A single representation is not sufficient for addressing the interconnected nature of traffic flows, as locally taken decisions can affect other parts of the system throughout time. For that reason, we introduce a Spatio-Temporal Q-Network architecture, which combines graph-based modeling for the spatial component along with sequence learning for the temporal one.

In complicated traffic conditions, efficient traffic control demands joint modeling of the spatial interaction between different intersections and temporal dynamics of the traffic itself. The separate consideration cannot account for the inherently connected structure of the traffic, where the impact of local decisions can extend through time throughout the traffic system. To cope with this challenge, we propose a unified Spatio-Temporal Q-Network architecture for modeling spatial interactions based on graphs and modeling temporal dynamics based on sequences. The core agent architecture, shown in Figure 5, processes a sequence of joint observations $X \in \mathbb{R}^{B \times N \times T \times D_{\max}}$, where T is the history length.

This representation enables the model to jointly capture spatial interactions among N intersections and temporal evolution across T time steps within a unified learning framework. Specifically, the architecture decomposes the learning process into three tightly coupled stages: (i) an embedding and spatial aggregation module based on Graph Attention Networks (GAT) to model non-Euclidean traffic dependencies, and (ii) a global context integration mechanism to propagate network-wide congestion signals. The resulting spatio-temporal representation is subsequently processed by a Double Dueling Q-Network to estimate action values, enabling coordinated decision-making across all agents while maintaining scalability and generalization across heterogeneous traffic scenarios.

4.3.1. Embedding and Spatial Fusion (GAT)

At each time step t, the local observation $x_{i,t}$ is projected to a hidden dimension $h_{i,t}^{(0)}$. To capture the non-Euclidean spatial structure of traffic propagation, we applied GAT [30]. Traffic congestion is inherently relational; a queue at intersection j will physically spill over to the upstream intersection i if left unaddressed. The GAT mechanism explicitly models this by computing an attention score $e_{ij,t}$ between connected intersections:

$$e_{ij,t} = \text{LeakyReLU}\left(\mathbf{a}_1^{\top} \mathbf{W} h_{i,t} + \mathbf{a}_2^{\top} \mathbf{W} h_{j,t}\right). \tag{6}$$

Equation (6) defines the pairwise attention score between connected intersections, where $e_{ij,t}$ is the attention from node i to node j at time t, $h_{i,t}$ and $h_{j,t}$ are node embeddings, $\mathbf{W}$ is a learnable linear projection, $\mathbf{a}_1, \mathbf{a}_2$ are learnable attention vectors, and LeakyReLU is the activation function, and it is used to compute the softmax-normalized attention coefficients that weight neighbor contributions during GAT aggregation. The LeakyReLU activation function is employed to ensure continuous gradient flow for negative attention scores, preventing dead neurons and allowing the model to learn meaningful distinctions even among less relevant neighboring intersections. By dynamically weighting neighbor information, the GAT allows an agent to shift its focus from local optimization to cooperative relief. As visualized in Figure 3b, the first GAT layer aggregates one-hop messages from direct neighbors, while Figure 3c shows how stacking layers enables two-hop communication through intermediate intersections. If a neighbor j reports high congestion (large hidden state magnitude), agent i assigns it a higher attention weight, effectively "anticipating" the incoming spillover and adjusting its phase duration to prevent gridlock before it reaches the local stop-line.

4.3.2. Global Context Pooling

While GAT layers effectively model first-order spatial dependencies, they are inherently limited by their local receptive fields, leading to "local myopia" in large-scale networks. To enable city-wide coordination, we define a global latent representation that serves as a macroscopic state summary. As illustrated in Figure 6, node-level embeddings are first pooled into a single global context token and then broadcast back to all intersections to support coordinated decision making.

Let $H_t^{\text{GAT}} = h_{1,t}^{\text{GAT}}, h_{2,t}^{\text{GAT}}, \ldots, h_{N,t}^{\text{GAT}} \in \mathbb{R}^{N \times d}$ be the set of hidden features for all N intersections at time t following spatial aggregation. We define the global context token $z_{\text{global},t}$ through a permutation-invariant aggregation function $F_{\text{pool}} : \mathbb{R}^{N \times d} \to \mathbb{R}^{2d}$:

$$\mu_{\text{global},t} = \frac{1}{N}\sum_{i=1}^{N} h_{i,t}^{\text{GAT}} \quad (7)$$

$$m_{\text{global},t} = \max_{i\in\{1,\ldots,N\}} h_{i,t}^{\text{GAT}} \quad (8)$$

$$z_{\text{global},t} = \left[\mu_{\text{global},t} \parallel m_{\text{global},t}\right] \quad (9)$$

$$\hat{h}_{i,t} = \sigma\Big(\mathbf{W}_{\text{fuse}}[h_{i,t}^{\text{GAT}} \parallel z_{\text{global},t}] + \mathbf{b}_{\text{fuse}}\Big) \quad (10)$$

Equation (9) defines the global context token, where $z_{\text{global},t}$ is the pooled global representation at time t, $h_{i,t}^{\text{GAT}}$ is the GAT feature of node i, N is the number of intersections, $\parallel$ denotes concatenation, and the max term is the element-wise maximum over nodes, and it is used in Equation (10) to broadcast city-wide traffic context to each intersection agent and this defines local-global feature fusion, where $\hat{h}_{i,t}$ is the fused embedding for intersection i, $h_{i,t}^{\text{GAT}}$ is the local spatial embedding, $z_{\text{global},t}$ is the broadcast token, $\mathbf{W}_{\text{fuse}}$ and $\mathbf{b}_{\text{fuse}}$ are learnable parameters, and $\sigma(\cdot)$ is the nonlinear activation, and it is important for conditioning each agent's policy on both its immediate neighborhood and the network-wide congestion state before temporal processing by the Transformer Encoder.

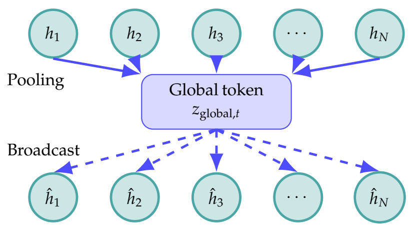


Figure 6. Global context pooling token in HydraLight. Node embeddings are pooled into $z_{\text{global},t}$ and then broadcast to all intersections.

The proposed global pooling operator is permutation-invariant with respect to node ordering. This ensures that the resulting global representation depends on the distribution of traffic states across the network, rather than on the arbitrary indexing of intersections. Let $H_t^{\text{GAT}} \in \mathbb{R}^{N\times d}$ denote the node embeddings at time t, and let $P \in \mathcal{P}_N$ be any $N \times N$ permutation matrix applied along the node dimension. The global pooling function satisfies

$$F_{\text{pool}}\Big(PH_t^{\text{GAT}}\Big) = F_{\text{pool}}\Big(H_t^{\text{GAT}}\Big), \qquad \forall P \in \mathcal{P}_N. \quad (11)$$

This property is important for scalable traffic signal control, since intersections in different city topologies may be indexed in arbitrary orders. In our formulation, mean pooling captures the overall operating regime of the traffic network, whereas max pooling highlights extreme congestion patterns associated with critical bottlenecks. Consequently, the resulting global token $z_{\text{global},t}$ encodes both holistic network-level conditions and worst-case traffic awareness.

To enable effective network-wide coordination, it is essential to construct a global representation that captures both the overall traffic state and the presence of critical congestion points. A single aggregation strategy is often insufficient, as it may either smooth out important local extremes or fail to reflect the general operating regime of the system. Therefore, we design a composite global context representation that combines complementary statistical properties of the node embeddings. This formulation allows the model to simultaneously reason about average traffic conditions and extreme congestion patterns, providing a more informative and balanced global signal. The mathematical rationale behind this design is detailed as follows:

- The Mean Component (μ_{global}): By computing the empirical mean of the hidden states, the network captures ambient traffic density across the topology, signaling whether the system is in a free-flow or saturated regime. Together with the Max Component, it forms the global token $z_{\text{global},t}$ (Equation (9)) that is fused with local features in Equation (10).
- The Max Component (m_{global}): The element-wise maximum serves as a bottleneck detector. Because the hidden states h_i encode local queue severity, m_{global} highlights the most critical congestion point in the city, even if it is topologically distant from a specific agent. Together with the Mean Component, it forms the global token $z_{\text{global},t}$ that is broadcast to all agents via Equation (10).

This formulation ensures that the local policy π_i is conditioned not only on the immediate neighborhood $\mathcal{N}_i$ but also on the systemic "pressure signal" of the entire network. This allows for proactive inflow gating in which peripheral agents reduce green time for incoming traffic during citywide saturation to prevent the propagation of gridlock.

4.4. Temporal Sequence and Transformer Processing

Traffic dynamics are inherently non-Markovian when observed through static snapshots. Because a single frame of queue length cannot distinguish between a queue that is dissipating and one that is rapidly accumulating, the environment acts as a Partially Observable Markov Decision Process that requires historical sequence modeling [33]. To resolve this partial observability, we do not rely on a single observation. Instead, we form a temporal window (sequence) of the last H observations. The raw input to the network for agent i at time t is the matrix $X_t^{(i)}$:

$$X_t^{(i)} = [o_{t-H+1}^{(i)}, \dots, o_{t-1}^{(i)}, o_t^{(i)}] \in \mathbb{R}^{H \times D_{obs}} \tag{12}$$

Equation (12) defines the temporal observation window, where $X_t^{(i)}$ is the stacked history for intersection i at time t, $o_\tau^{(i)}$ is the local observation at time τ, H is the history length, and D_{obs} is the observation dimension, and it servers as the input sequence to the Temporal Transformer Encoder for modeling long-horizon traffic dynamics and resolving partial observability. where H is the history length (set to 10 frames).

In our implementation, the temporal history length is fixed to $H = 10$ observations. Since each observation corresponds to one traffic control interval ($\Delta t = 10$ s), the effective temporal receptive field of the Transformer spans approximately 100 s of recent traffic evolution. This backtracking window was intentionally selected to capture short- and medium-term traffic dynamics such as platoon arrivals, queue propagation, congestion spillback, and signal-cycle interactions while maintaining computational efficiency for large-scale deployment.

After these observations undergo spatial and global context fusion and reward shaping, the resulting features form a historical sequence that is processed by a Temporal Transformer Encoder [34]. The Multi-Head Self-Attention mechanism enables the agent to attend to relevant historical events (e.g., a platoon arrival three cycles ago) regardless of their temporal distance. By capturing these long-term dependencies, the network can implicitly infer traffic flow derivatives (acceleration and deceleration trends) and predict future arrival rates more accurately than recurrent models (RNNs), which often suffer from vanishing gradients over long horizons.

Crucially, we implement a Direct Observation Skip Connection in the final decision head. While the Transformer output h_i^{final} effectively encodes historical context, deep sequence models can suffer from feature over-smoothing, potentially diluting the urgency of the current state. To mitigate this, we concatenate h_i^{final} directly with the initial linear embedding of the observation at the current time step, denoted $e_t^{(i)}$. This ensures the policy remains highly sensitive to immediate, localized traffic signals while still being modulated by the broader spatio-temporal context:

$$z_{\text{final}} = [h_i^{\text{final}} \parallel x_{i,t}^{\text{embed}}]. \tag{13}$$

Equation (13) defines the final representation, where z_{final} is the feature passed to the value heads, h_i^{final} is the Transformer output for agent i, and $x_{i,t}^{\text{embed}}$ is the current-step embedded observation, and it is used as the input to the Dueling Q-Network (Equation (14)) to compute action values while preserving sensitivity to the current traffic state. In the last stage, the action values are computed using the Double Dueling architecture [32], which separates the estimation of the state-value V and the action-advantage A:

$$Q(s,a) = V(z_{\text{final}}) + \left(A(z_{\text{final}}, a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(z_{\text{final}}, a') \right). \tag{14}$$

Equation (14) defines the dueling decomposition, where $Q(s,a)$ is the action-value for state s and action a, $V(z_{\text{final}})$ is the state-value stream, $A(z_{\text{final}}, a)$ is the advantage stream, and $|\mathcal{A}|$ is the number of available actions, and it is used for both ε-greedy action selection during training and computing TD errors for the Unified PER sampling priorities (Equation (18)).

4.5. Reward Function: Adaptive Flow Optimization

Defining a stationary reward function across diverse scenarios is critical. A raw reward metric (e.g., queue length) might range from 0 to 10 in light traffic but surge to over 100 in a congested grid. Such drastic scale variations can destabilize gradient descent, causing the agent to prioritize scenarios with naturally larger reward magnitudes [35].

To address this, we employ *Online Adaptive Standardization* to dynamically normalize rewards. For any raw metric $x \in Q_t, V_t, W_t$ (Queue, Speed, Waiting Time), we maintain running estimates of the mean μ_t and standard deviation σ_t using an Exponential Moving Average (EMA) with decay $\beta = 0.05$. The bounded Z-score $\hat{x}_t$ is computed as:

$$\hat{x}_t = \tanh\left(\frac{x_t - \mu_t}{\sigma_t + \epsilon} \right). \tag{15}$$

Equation (15) defines bounded online standardization, where $\hat{x}_t$ is the normalized metric at time t, x_t is the raw metric value, μ_t and σ_t are running mean and standard deviation, ϵ is a numerical-stability constant, and $\tanh(\cdot)$ bounds the output to $[-1, 1]$, and it is applied independently to each raw traffic metric (Q_t, V_t, W_t) to produce the normalized terms $\hat{Q}_t$, $\hat{V}_t$, $\hat{W}_t$ used in Equation (16). The final reward $r_t^{(i)}$ maximizes throughput (Speed) while minimizing delay (Wait, Queue):

$$r_t^{(i)} = \hat{V}_t - \hat{W}_t - \hat{Q}_t. \tag{16}$$

Equation (16) defines the normalized local reward, where $r_t^{(i)}$ is the reward at intersection i, $\hat{V}_t$ is normalized speed, $\hat{W}_t$ is normalized waiting time, and $\hat{Q}_t$ is normalized queue length, and it is stored in the shared replay buffer as the reward signal for the Bellman updated during Q-network training (Algorithm 1).

Algorithm 1 HYDRA-MT Training with Unified PER.

Require: Set of scenarios $\mathcal{T}$, history length K, buffer capacity C
Ensure: Optimized policy π_θ

```
Initialize unified buffer D, parameters θ, target θ′ ← θ
for episode e = 1 to E do
    Sample scenario τ from T uniformly and reset environment
    H ← ∅                                              ▷ Initialize history buffer
    for each time step t do
        Get raw observations o_t and update history H_t
        Construct state s_t = (o_k, a_{k−1}, r_{k−1})_{k=t−K+1}^{t}
        Select action a_t using ϵ-greedy on Q(s_t; θ)
        Execute a_t, observe r_t and s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in D
        if size(D) > batch_size then
            Sample minibatch B via Unified PER (Eq. (18))
            y = r + γQ(s′, arg max_{a′} Q(s′, a′; θ); θ′)
            Update θ by minimizing L(θ) using weight w_i
            θ′ ← τ_soft θ + (1 − τ_soft)θ′                 ▷ Soft update
        end if
    end for
end for
```

4.6. Multi-Scenario Training Orchestration

To train a robust, generalizable policy, we employ a Round-Robin Scenario Sampling strategy combined with Unified PER. This design ensures that the learning process is not biased toward any single traffic topology or demand pattern, which is critical in heterogeneous urban environments. In the round-robin scheme, the agent sequentially interacts with multiple traffic scenarios, enabling balanced exposure to diverse spatial structures and temporal dynamics. This prevents overfitting to simpler or more frequently encountered environments. Complementing this, Unified PER mechanism normalizes Temporal-Difference (TD) errors at the scenario level, allowing sampling priorities to reflect relative learning difficulty rather than absolute reward magnitudes. As a result, transitions from complex and underrepresented scenarios are not overshadowed by those from simpler environments with inherently larger TD errors. Together, these mechanisms promote stable convergence, improve sample efficiency, and enhance the transferability of the learned policy across unseen traffic networks.

4.6.1. Data Collection Schedule

Data collection strategy plays an important role in multi-scenario reinforcement learning and impacts training stability, diversity of samples, and generalization capability. An ill-designed data collection method can easily introduce biases into the training process, which will make the learned model biased towards certain traffic flows or topological structures. As a result, this may reduce the transferability of the learning agent and limit its performance in other situations. To address this issue, we need to use a carefully designed method for data collection, in which different traffic scenarios are explored to train a robust controller. We train the agent concurrently on a set of M diverse traffic scenarios $\mathcal{M} = m_1, \ldots, m_M$ (e.g., Synthetic Grids, Ingolstadt, Cologne). The training loop proceeds as follows:

1. Episode Initialization: At the start of each training epoch, we reset all M environments.
2. Step-wise Interleaving: In every collection step, the agent interacts with *one* active scenario m_k, selected in a round-robin fashion. This prevents the replay buffer from becoming temporally correlated to a single map's dynamics.

3. Global Buffer Storage: Transitions $(s_t, a_t, r_t, s_{t+1})_m$ from *all* maps are pushed into a single, global Replay Buffer $\mathcal{D}$.

4.6.2. Unified Prioritized Experience Replay (Unified PER)

To ensure balanced learning across heterogeneous traffic scenarios, we propose Unified Prioritized Experience Replay (Unified PER), an extension of standard PER tailored for multi-scenario training. Unlike conventional approaches that prioritize samples based on absolute Temporal-Difference (TD) error, Unified PER normalizes TD errors at the scenario level, allowing priorities to reflect relative learning difficulty rather than magnitude. This prevents simpler scenarios with inherently larger rewards from dominating the replay process and ensures that complex or underrepresented environments receive sufficient training attention. As a result, Unified PER improves training stability, sample efficiency, and generalization across diverse traffic networks.

Specifically, we use a single shared SumTree buffer D as a Difficulty-Aware replay memory that stores transitions from all training maps concurrently and prioritizes them relative to scenario-specific difficulty. Standard PER prioritizes transitions by the absolute Temporal-Difference (TD) error magnitude ($|\delta|$). In multi-scenario training, a simple grid with heavy traffic naturally produces larger absolute rewards and, therefore, larger TD-errors (e.g., $\delta \approx 10$), whereas a complex, optimized city map may produce smaller absolute values (e.g., $\delta \approx 1$). Consequently, standard PER tends to oversample simpler high-magnitude scenarios, starving complex maps of gradient updates and leading to task dominance.

Unified PER intervenes during the *batch sampling* phase. Crucially, unlike standard PER, which prioritizes samples based on the raw TD-error, Unified PER prioritizes samples based on their *deviation from the average difficulty of their specific scenario*. This makes prioritization comparable across heterogeneous maps. To achieve this, we maintain a running estimate of the average absolute TD-error for each map m, denoted by $\bar{\mu}_m$, which is updated using EMA:

$$\bar{\mu}_{m,k} = \beta \cdot \bar{\mu}_{m,k-1} + (1-\beta) \cdot \frac{1}{|B_m|} \sum_{j \in B_m} |\delta_{j,m}|. \tag{17}$$

Equation (17) defines the EMA estimate of scenario-specific TD-error scale, where $\bar{\mu}_{m,k}$ is the running mean absolute TD error for map m at update k, β is the EMA coefficient, B_m is the sampled mini-batch subset from map m, and $\delta_{j,m}$ is the TD error of sample j, and it is used as the per-scenario normalization baseline $\bar{\mu}_m$ in Equation (18) to compute scenario-relative sampling priorities.

The sampling priority $p_{i,m}$ for transition i originating from map m is then computed using the *normalized* TD-error:

$$p_{i,m} = \left(\frac{|\delta_{i,m}|}{\bar{\mu}_m + \epsilon} \right)^{\alpha}. \tag{18}$$

Equation (18) defines the Unified PER sampling priority, where $p_{i,m}$ is the priority of sample i from map m, $\delta_{i,m}$ is its TD error, $\bar{\mu}_m$ is the map-specific TD-error scale, ϵ is a stability constant, and α controls prioritization strength, and it is used by the SumTree replay buffer to sample training mini-batches in a difficulty-aware manner across all scenarios(Algorithm 1, line 12). Here, ϵ is a small constant added for numerical stability, and α controls the strength of prioritization, as in standard PER. However, due to the normalization by $\bar{\mu}_m$, α now governs prioritization with respect to a map-specific difficulty baseline rather than absolute error magnitudes.

This formulation drives sampling according to *relative surprise* rather than absolute TD-error magnitude. A transition is prioritized if it is unexpected relative to its map's typical error scale, preventing simpler scenarios with naturally larger TD errors from dominating the replay buffer and starving more complex environments of gradient updates. Training proceeds by minimizing the importance-weighted mean squared error of the Bellman error using Double Dueling DQN targets. Specifically, Double Q-learning is employed for stable target estimation, while the dueling value–advantage decomposition improves value function representation [31]. Together, this ensures balanced gradient updates and robust feature learning across heterogeneous network topologies. The complete training procedure is detailed in Algorithm 1.

4.7. Hyperparameter Settings

The choice of hyperparameters is a crucial consideration that has a direct impact on the stability of the system and its convergence properties, among others [36,37]. This is especially true when considering multi-agent and multi-scenario reinforcement learning frameworks because an inappropriate setup might yield unstable training processes, convergence problems, or an environment-biased learning procedure. Hyperparameters such as the learning rate, discount factor, replay buffer size, and priority coefficients are some important parameters that affect the balance between exploration and exploitation, as well as the efficient transfer of knowledge across various scenarios. Furthermore, architectural hyperparameters, like the number of GAT layers and attention heads, as well as the depth of Transformers, determine the ability of the system to learn complex spatial-temporal dependencies [38].

The hyperparameters used in this study are listed in Table 2. Careful selection and tuning of these hyperparameters are critical to the stability, convergence speed, and overall performance of the proposed HydraLight framework. Given the complexity of multi-agent reinforcement learning under heterogeneous and multi-scenario environments, parameters such as the learning rate, discount factor, replay buffer size, and prioritization coefficients directly influence the balance between exploration and exploitation, as well as the efficiency of knowledge transfer across scenarios.

In particular, the discount factor γ governs the trade-off between short-term responsiveness and long-term traffic optimization, while the learning rate controls the stability of gradient updates in the presence of non-stationary traffic dynamics. The configuration of the replay buffer and batch size affects sample diversity and training robustness, especially when combined with the Unified PER mechanism. Furthermore, architectural hyperparameters, including the number of GAT layers, attention heads, Transformer depth, and embedding dimensions, determine the model's capacity to capture complex spatial and temporal dependencies without overfitting. The chosen hyperparameter settings reflect a balance between computational efficiency and representational power, ensuring stable training across diverse traffic scenarios while maintaining strong generalization performance.

The hyperparameter values presented in Table 2 were determined through a combination of established reinforcement learning practices, preliminary ablation studies, and empirical stability evaluations conducted across multiple traffic scenarios. Parameters associated with temporal modeling capability, such as the history length H and Transformer embedding dimensions, were tuned by analyzing convergence behavior and the model's ability to capture long-term traffic dynamics. Similarly, optimization-related parameters, including the learning rate, replay buffer size, batch size, and discount factor, were selected based on their impact on training stability, convergence consistency, and sample efficiency. Furthermore, the architectural settings of the Graph Attention Network, such as the number of attention heads and hidden feature dimensions, were configured to achieve a balance between expressive spatial representation learning and computational efficiency. The final hyperparameter configuration was validated using both synthetic and real-world

RESCO environments to promote robust cross-scenario generalization and reduce the risk of overfitting to a specific traffic topology.

Table 2. Hyperparameter Settings.

Category	Parameter	Value
General	Total episodes	300
	Discount Factor (γ)	0.95
	Batch Size	128 (2^7)
	Replay Buffer Size	524,288 (2^{19})
	Min Replay Size	3600 steps
	Soft Update (τ)	0.001
	Learning Rate	1×10^{-3}
Prioritized Replay	Prioritization (α)	0.6
	Bias Correction (β)	0.3 (annealed to 1.0)
Network Architecture	GAT layers	2
	GAT Heads	4
	GAT Within-Dimension	128
	Transformer Layers	2
	Transformer Heads	4
	Transformer Dimension	128
	Sequence Length (T)	10
	DQN layers	2
	DQN Dimension	128

4.8. Computational Complexity and Scalability Analysis

The computational cost of HydraLight mainly comes from its GAT and Temporal Transformer components. In the spatial modeling stage, the GAT module computes attention only between physically connected neighboring intersections rather than across the entire traffic network, allowing the complexity to scale approximately with the number of road connections. Similarly, the Temporal Transformer processes a bounded historical sequence of traffic observations. Since the temporal window length is intentionally kept small ($H = 10$ in this study), the temporal attention overhead remains limited and does not significantly increase with city size.

Overall, the inference complexity of HydraLight scales nearly linearly with the number of intersections due to localized graph communication and parameter sharing among agents, making the framework more practical for large-scale urban traffic networks compared to fully connected global attention mechanisms. Nevertheless, Transformer-based architectures naturally introduce additional computational overhead compared to lightweight rule-based traffic controllers. Therefore, for extremely large-scale deployments involving thousands of intersections, further optimization strategies such as sparse attention, hierarchical graph partitioning, or distributed edge inference may be beneficial to satisfy strict real-time operational requirements.

5. Experiments

In order to prove the efficacy, accuracy, and generality of TSC algorithms, an effective and systematic experimental design is necessary [39]. Otherwise, results of the experiments could suffer from bias and be difficult to reproduce or compare with one another [40]. To resolve this issue, we employ a unified and standardized approach for testing all TSC algorithms. This allows us to make sure that our evaluations are transparent, unbiased, and scientifically sound. More specifically, we ensure that simulations are conducted under the same parameters and criteria for all tested algorithms, but keep the initial training configuration of each benchmark intact. By doing this, we enable a fair comparison with other state-of-the-art methods, which will make it possible to demonstrate the true advantages of our algorithms. The key principles of our testing procedure are listed below:

All experiments are conducted using the SUMO (Simulation of Urban Mobility) simulator within the RESCO benchmark framework to ensure consistency and realism [41]. To maintain fairness across all methods, we enforce standardized control settings, including a control interval of $\Delta t = 10$ s and a mandatory yellow transition duration of $t_{yellow} = 5$ s. This yellow phase complies with urban traffic safety regulations and accounts for the unavoidable "lost time" during signal transitions. Each training and evaluation episode spans 3600 simulation seconds, corresponding to one hour of traffic flow.

For performance evaluation, we adopt Average Trip Time as the primary metric, defined as the mean duration (in seconds) a vehicle spends traveling from entry to exit within the network, thereby providing a comprehensive measure of traffic efficiency. To ensure a fair comparison and eliminate biases due to implementation differences, the performance results of all baseline methods (e.g., CoLight, X-Light, CoSLight, and DuaLight) are sourced directly from their original publications. This guarantees that each baseline reflects its best-reported performance under identical RESCO configurations and demand settings.

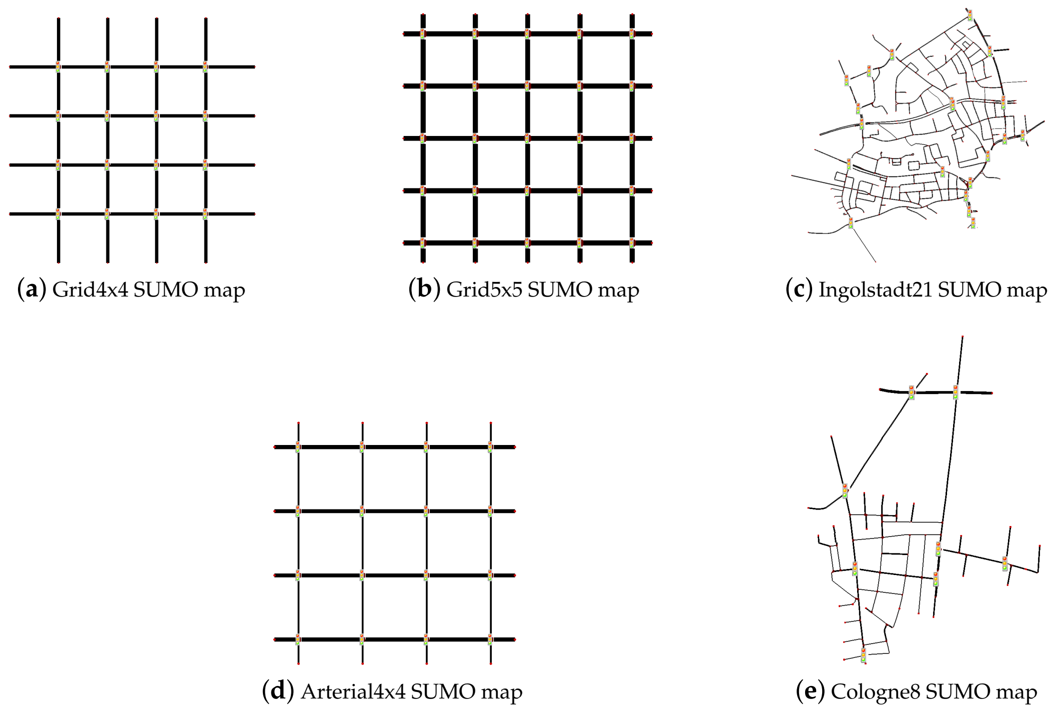
Furthermore, each method is evaluated at or near its optimal convergence point to ensure meaningful comparison. For example, X-Light relies on an on-policy PPO framework requiring up to 1000 training episodes, whereas DuaLight achieves convergence in approximately 200 episodes through scenario-specific transfer learning [13,42]. In contrast, HydraLight demonstrates strong sample efficiency by converging to superior performance within a fixed training budget of 300 episodes.

Finally, to ensure robustness and reproducibility of the results, we decouple training randomness from evaluation. While randomized seeds are used during training to enhance model robustness, the final performance metrics are computed over a fixed set of 10 independent testing seeds (indices 0–9), following established evaluation practices in recent studies such as DuaLight [42] and CoSLight [25].

To ensure a comprehensive evaluation, we consider a diverse set of *evaluation scenarios (datasets)* and assess HydraLight alongside established baselines across five different traffic networks (visualized in Figure 7). These scenarios range from synthetic grid environments designed to test specific traffic flow dynamics to complex, real-world city topologies, as detailed in Table 3.

Table 3. Evaluation Traffic Network Scenarios.

Scenario	Intersections	Description
Grid 4×4	16	A synthetic network (4×4) with uniform traffic flow, serving as a baseline for MARL stability and coordination.
Grid 5×5	25	A larger synthetic grid (5×5) with increased complexity and higher gridlock potential due to more agents and extended spatial dependencies.
Arterial 4×4	16	A heterogeneous variant of the 4×4 grid simulating arterial roads, where dominant flows must be prioritized over minor cross traffic.
Ingolstadt21	21	A real-world network from Ingolstadt, Germany, featuring irregular topology, unequal node spacing, and asymmetric traffic patterns.
Cologne8	8	A dense real-world subnet from Cologne, Germany, with tightly coupled intersections prone to rapid congestion propagation.



(**a**) Grid4x4 SUMO map (**b**) Grid5x5 SUMO map (**c**) Ingolstadt21 SUMO map

(**d**) Arterial4x4 SUMO map (**e**) Cologne8 SUMO map

Figure 7. Representative traffic-network maps used in evaluation.

In order to systematically assess the efficacy of our proposed HydraLight framework, we perform a comprehensive evaluation against a range of *baseline methods and algorithmic variants* that capture the diversity of approaches in TSC. The selected baselines represent the most widely adopted paradigms in the field, spanning from classical control strategies and heuristic-based adaptive systems to more recent learning-based techniques.

These baseline techniques may be classified into three broad categories: (i) classical and heuristic controllers, (ii) queue imbalance minimization-based algorithms, and (iii) learning-based algorithms. Traditional methods like SCOOT and SCATS use pre-defined control rules and simple traffic models to ensure stability at the cost of adaptability to changing conditions. Queue imbalance minimization-based algorithms, such as MaxPressure, make use of theoretical foundations to efficiently derive local control actions for each intersection; however, these techniques lack global coordination and temporal insight. Learning-based algorithms, such as CoLight, X-Light, and CoSLight, employ neural networks to infer optimal control rules through learning, incorporating spatial, temporal, and communication-based coordination approaches.

Beyond these baselines, we further examine some variations in algorithms to assess the role of different components in our approach. Some of these variations include HydraLight that does not contain any of the following modules: Global Context, Temporal Transformer, and Unified PER respectively. Through such algorithmic variations, we can better study how each component affects the performance, scalability, and generality in heterogeneous traffic conditions of the proposed approach compared with other solutions. Such comparative analyses not only reveal the shortcomings of current state-of-the-art models but also establish the place of HydraLight within the realm of existing TSC methods.

Within this context, *heuristic methods* represent one of the fundamental approaches in traffic signal control, relying on predefined rules and logical formulations rather than learning from data [43]. These methods typically utilize local traffic information, such as queue lengths and flow rates, to make control decisions at individual intersections. While heuristic algorithms offer advantages in terms of low computational complexity and high interpretability, their rule-based nature limits their ability to adapt to complex, dynamic, and large-scale traffic environments. As a result, their performance is generally inferior to more advanced data-driven approaches, as illustrated in Table 4.

Table 4. Heuristic Methods.

Methods	Explanation
Fixed-Time Control (FTC) [44]	Executes a pre-determined, cyclic signal schedule.
MaxPressure [17]	A greedy strategy that minimizes the pressure difference between upstream and downstream queues.

Within this category, *single-scenario reinforcement learning (no transfer)* refers to approaches where the agent learns traffic signal control policies through interaction with a single, fixed environment, without leveraging knowledge transfer across different scenarios. In this setting, the agent is both trained and evaluated on the same traffic network, allowing it to specialize and exploit the specific topology and traffic patterns of that environment. While such methods can achieve strong performance under these controlled conditions, their lack of exposure to diverse scenarios limits their ability to generalize to unseen or structurally different traffic networks, as illustrated in Table 5.

Table 5. Single-Scenario RL Methods.

Methods	Explanations
IPPO (Independent PPO) [45,46]	Agents learn independently using PPO.
rMAPPO [47]	A centralized training with decentralized execution extension of PPO using recurrent neural networks.
CoLight [10]	Uses GAT for neighbor communication.
MPLight [48]	Combines the FRAP model with MaxPressure reward principles.

In contrast, *meta-learning and generalizable reinforcement learning* approaches are specifically designed to enable scenario transfer across diverse traffic environments. Their primary objective is to learn scenario-invariant representations or initialization policies that can rapidly adapt to new and previously unseen traffic conditions with minimal additional training. By leveraging techniques such as meta-learning, transfer learning, and shared representation learning, these models improve scalability and adaptability across varying network topologies and traffic patterns. Unlike single-scenario methods, they explicitly account for differences in traffic dynamics and intersection configurations, making them more suitable for large-scale, real-world deployments, as illustrated in Table 6.

To improve methodological transparency, the performance results of several baseline methods, including CoLight, X-Light, and CoSLight, were directly adopted from their original publications in cases where official implementations or fully reproducible training configurations were not publicly available. These baseline studies were generally evaluated on RESCO-compatible traffic environments consisting of both synthetic grid-based networks and selected real-world urban scenarios, using commonly reported evaluation metrics such as Average Travel Time, queue length, traffic delay, and throughput. Nevertheless, certain experimental factors, including simulator versions, random traffic generation seeds, hardware environments, reward formulations, and training schedules, may differ across studies and inevitably introduce some level of variability into cross-publication comparisons. Therefore, the reported performance differences should be interpreted as approximate comparative references rather than perfectly identical experimental replications conducted under fully unified conditions.

https://doi.org/10.3390/su18115252

Table 6. Meta-Learning and Generalizable RL Methods.

Methods	Explanations
MetaLight [24]	Uses model-agnostic meta-learning to adapt a base policy to new tasks.
MetaGAT [49]	Combines meta-learning with GAT for spatial generalization.
GESA [14]	A scenario-agnostic approach for diverse intersection structures.
CoSLight [25]	Co-optimizes collaborator selection and decision-making.
DuaLight [42]	Separates scenario-specific and scenario-shared knowledge.
X-Light [13]	Uses a Transformer-on-Transformer architecture for cross-city transfer.
HydraLight (Ours)	A unified spatio-temporal framework with a Global Context Pooling module for network-level awareness. Training is stabilized with Unified PER.

In order to make a fair comparison among the baselines which were able to be fine-tuned, a standardized experimental setup was adopted for all the relevant methods when their implementation or codebase could be found to conduct the experiment. The training process would share identical settings in terms of SUMO simulation setting, control interval, duration of each episode, random seed, replay buffer size, optimizer, and training budget compared to HydraLight. Moreover, all the experiments were performed on an identical hardware platform to make sure of fair comparisons regarding the running time.

6. Experimental Results and Analysis

In this section, we present an extensive analysis of our HydraLight architecture framework, based on effectiveness and robustness criteria for different traffic setups. We examine the performance of the model compared to the state-of-the-art baselines using experiments conducted under the same conditions, with an emphasis on the capability of the model to work across different network configurations. Besides evaluating overall performance criteria, we also consider the effects of different elements of the model's architecture on achieving higher performance levels in different traffic dynamics.

6.1. Quantitative Comparison

In order to objectively measure the performance of HydraLight compared with heuristic algorithms, pressure-based strategies, and deep learning algorithms, a series of simulations were conducted based on the RESCO benchmark with consistent simulation settings in which all methods' evaluation was measured by Average Trip Time. Results demonstrate that HydraLight outperforms all baselines in both synthetic and real-world traffic networks. Specifically, heuristic solutions like Fixed-Time Control and MaxPressure present decent yet limited effectiveness, since they can barely cope with complicated situations because of a lack of flexibility. Although Deep Reinforcement Learning (DRL)-based algorithms including CoLight, X-Light, and CoSLight learn effective policies based on data inputs, they suffer from deficiencies in spatial modeling or temporal modeling separately. In contrast, HydraLight shows better performance, since it considers spatiotemporal dependency by combining Graph Attention Network and Temporal Transformer together, and it learns the overall context of traffic network by introducing Global Context module. Moreover, HydraLight is able to guarantee balanced learning for all scenarios through Unified PER. Based on the above experimental analysis, HydraLight is also able to realize a reduction of up to 13% in average travel time in arterial networks. Table 7 presents the average trip time across all scenarios. The proposed HydraLight framework achieves state-of-the-art performance, outperforming the strongest baselines (X-Light and CoSLight) in 4 out of 5 maps.

Table 7. Average trip time (seconds) on the evaluated traffic networks.

Methods	Avg. Trip Time (Seconds)				
	Grid4 × 4	**Grid5 × 5**	**Arterial4 × 4**	**Ingolstadt21**	**Cologne8**
FTC	206.68 ± 0.54	550.38 ± 8.31	828.38 ± 8.17	319.41 ± 24.48	124.40 ± 1.99
MaxPressure	175.97 ± 0.70	274.15 ± 15.23	686.12 ± 9.57	375.25 ± 2.40	95.96 ± 1.11
CoLight	163.52 ± 0.00	242.37 ± 0.00	409.93 ± 0.00	337.46 ± 0.00	89.72 ± 0.00
MPLight	179.51 ± 0.95	261.76 ± 6.60	541.29 ± 45.24	319.28 ± 10.48	98.44 ± 0.62
IPPO	167.62 ± 2.42	259.28 ± 9.55	431.31 ± 28.55	379.22 ± 34.03	90.87 ± 0.40
rMAPPO	164.96 ± 1.87	300.90 ± 8.31	565.67 ± 44.80	453.61 ± 29.66	97.68 ± 2.03
MetaLight	169.21 ± 1.26	247.83 ± 5.99	381.77 ± 12.85	292.26 ± 4.40	91.57 ± 0.75
MetaGAT	165.23 ± 0.00	266.60 ± 0.00	374.80 ± 0.87	290.73 ± 0.45	96.74 ± 0.00
GESA	161.33 ± 1.34	252.11 ± 9.94	393.57 ± 13.72	320.02 ± 5.57	90.59 ± 0.74
CoSLight	$\underline{159.11 \pm 3.12}$	$\mathbf{220.32 \pm 1.71}$	364.21 ± 4.78	284.62 ± 2.57	90.46 ± 0.55
DuaLight	161.04 ± 0.00	221.83 ± 0.00	396.65 ± 0.00	317.97 ± 0.00	89.74 ± 0.00
X-Light	162.47 ± 0.00	$\underline{220.63 \pm 0.00}$	$\underline{349.60 \pm 0.00}$	$\underline{278.05 \pm 0.00}$	$\underline{88.55 \pm 0.00}$
HydraLight	$\mathbf{159.11 \pm 0.85}$ (0%)	221.77 ± 5.38 (−0.65%)	$\mathbf{303.90 \pm 8.11}$ (13.07%)	$\mathbf{271.04 \pm 13.92}$ (2.58%)	$\mathbf{87.32 \pm 0.45}$ (1.38%)

Performance comparison on scenarios encountered during training. Single-scenario baselines (FTC to rMAPPO) are trained and tested independently on the specific scenario of each column. Conversely, multi-scenario methods (MetaLight to HydraLight) are trained concurrently across a unified set of five scenarios, with evaluation results reported for the five scenarios presented here. The best results are bolded, and the second-best are underlined.

6.2. Performance Analysis by Network Topology

HydraLight achieves its most substantial improvements in scenarios characterized by structural irregularity and asymmetric demand, on the Arterial 4 × 4 network, which simulates a realistic hierarchy of major and minor roads, our method reduces average travel time by 45.7 s compared to X-Light (303.90*s* vs. 349.60*s*), a margin of 13.07%. Similarly, on the real-world *Ingolstadt21* and *Cologne8* maps, HydraLight consistently achieves the lowest trip times. These gains suggest that our hierarchical architecture excels in *structured heterogeneity*. In arterial systems, congestion propagates directionally; a blockage on a main artery is far more critical than one on a side street. HydraLight's Global Pooling Context effectively captures this network-level imbalance, allowing agents to gate inflow from minor roads during systemic overload, thereby preserving flow on critical corridors.

On the synthetic Grid 5 × 5 network, HydraLight (221.77 s) performs comparably to state-of-the-art baselines like CoSLight (220.32 s) and X-Light (220.63 s), while significantly outperforming purely local methods like CoLight (242.37 s). This result highlights the scalability of the proposed architecture. While simple local heuristics struggle as the grid size increases ($N = 25$), HydraLight's Global Pooling Context effectively bridges the network diameter. The parity with X-Light suggests that while Global Pooling is necessary to overcome local myopia, the uniform grid structure creates an optimization ceiling, where distinct global strategies converge to similar travel times.

6.3. Ablation Study: Component Contribution Analysis

To isolate the specific contributions of HydraLight's architectural innovations, we conducted an ablation study by selectively deactivating the Global Pooling Context and the Unified PER mechanism. The results, visualized in Figure 8, address two critical design questions. These questions stem directly from the two central architectural hypotheses introduced in Section 4: whether global context improves network-level coordination and whether normalized prioritization prevents multi-scenario task dominance. To align contribution claims with empirical validation, we present the ablation evidence in the same naming scheme as the core contributions, using Figure 8 and Table 7 as the primary proof sources.

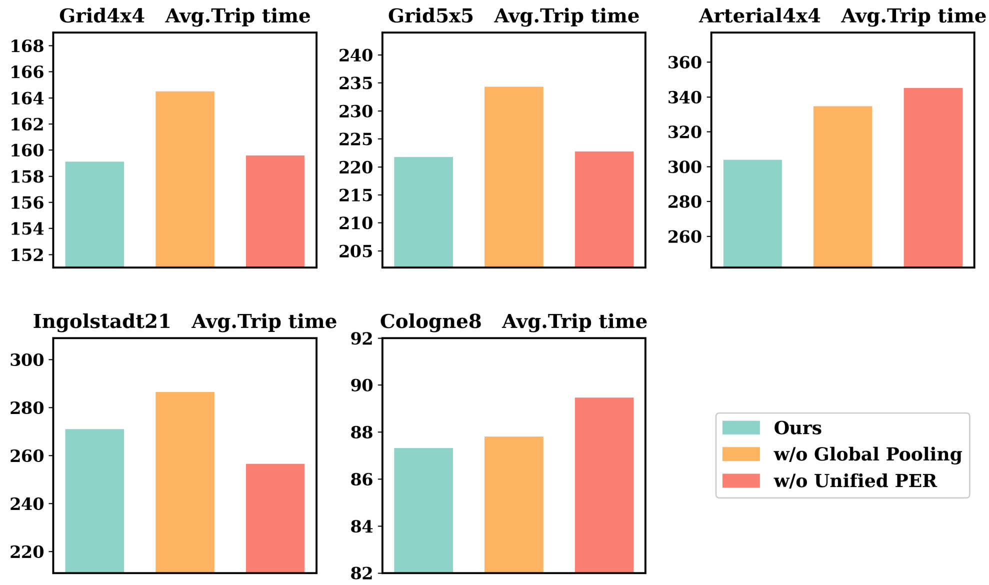


Figure 8. Impact of key modules on model performance. The ablation study compares the full HydraLight framework against variants excluding the Global Pooling Context and Unified PER. The performance drop on *Arterial4x4* highlights the necessity of global context for arterial flows, while the degradation on *Ingolstadt21* confirms the role of Unified PER in handling complex real-world topologies.

6.3.1. All-in-One Space and Time Modeling

The full HydraLight model (Table 7) achieves the largest gains on structurally demanding maps such as *Arterial 4x4* and *Ingolstadt21*, indicating that joint spatial-temporal reasoning is essential when congestion propagates both across intersections and over time. Figure 8 further supports this interpretation: when key coordination modules are removed, performance degrades markedly on these same maps, showing that the integrated representation is not cosmetic but functionally necessary.

6.3.2. Data-Driven Heterogeneous Alignment

Table 7 shows that one shared policy remains competitive across synthetic grids and irregular real-world topologies, including *Ingolstadt21* and *Cologne8*. This cross-topology consistency indicates that a simple shared linear projection layer is sufficient to align heterogeneous intersection observations into a transferable latent space, without requiring a deeper per-intersection MLP for feature extraction. Instead, feature extraction is effectively offloaded to the downstream GAT layers, which learn topology-aware representations after alignment.

6.3.3. Big-Picture Information Sharing

Figure 8 shows that removing the Global Pooling Context degrades performance across all tested topologies. The impact is most severe on *Arterial 4x4* and *Ingolstadt21*, where travel times increase by over 13%, and even on the uniform *Grid 5x5* we observe a distinct 6% performance drop without global context. Combined with Table 7, this confirms that city-level context is essential not only for irregular networks but also for larger networks where local message passing alone is insufficient.

6.3.4. Robust Multi-Scenario Training

The Unified PER ablation in Figure 8 shows a catastrophic failure on *Ingolstadt21*: deactivating Unified PER (reverting to standard experience replay) increases delay by nearly 20% relative to the full model. This evidence, together with the full-model superiority in Table 7, confirms that difficulty-aware replay is crucial for preventing task dominance and maintaining stable learning across mixed-complexity scenarios.

6.4. Impact of Unified PER vs. Dual-Buffer Architectures

A key distinction in our framework is the implementation of a Unified PER, which contrasts with existing state-of-the-art methods such as DuaLight [42]. While DuaLight utilizes a dual-buffer system to separate scenario-shared and scenario-specific knowledge, our single-buffer approach demonstrates superior generalization for two primary reasons:

The first is in regard to Dynamic Difficulty Allocation in which dual-buffer architectures often rely on a fixed sampling ratio between shared and specific knowledge; our Unified PER, however, treats the multi-map experience as a single distribution. If a complex real-world intersection (e.g., *Ingolstadt21*) presents higher optimization difficulty than a simple grid, HydraLight naturally shifts its training capacity toward those bottlenecks by merit of their higher TD-errors. Secondly, Information Density and TD-Normalization is accomplished. By normalizing TD-errors across all scenarios within a single global buffer, HydraLight identifies "surprising" transitions that might be sparse in one map but critical for global policy stability. This prevents the "gradient dilution" that occurs when sampling from independent buffers with varying reward scales.

This theoretical advantage is empirically supported by the ablation results in Figure 8. On the complex *Ingolstadt21* network, removing the Unified PER mechanism (red bar) results in a notable performance degradation compared to the full model. This confirms that without difficulty-aware prioritization, the agent struggles to master the irregular topology of real-world intersections. Furthermore, as shown in Table 7, HydraLight achieves significantly lower average trip times than DuaLight, most notably in the *Arterial4x4* scenario (a $\sim$23% improvement), proving that *sample difficulty* is a more robust metric for generalization than *scenario-type separation*.

6.5. Generalization to Real-World Topologies

On the *Ingolstadt21* and *Cologne8* benchmarks, HydraLight achieves the lowest travel times. By padding observations to a unified dimension of 64 and utilizing a shared projection layer, our model effectively transfers knowledge from synthetic grids to irregular real-world structures, proving that the learned policy is not overfit to specific geometries. The ability of the model to generalize can be reinforced by the fact that the model can accommodate heterogeneous intersection layouts and non-homogenous traffic densities without necessitating retuning according to individual scenarios. The use of GAT facilitates the capacity of the model to adjust to the changing graph structures through learning inter-node interactions, whereas the Temporal Transformer extracts invariant temporal behaviors regardless of the surrounding environments. In addition, the use of the Global Context Pooling technique facilitates the awareness of the model about global congestion conditions, which is highly relevant in practice due to the presence of uneven traffic flows and choke points within the networks.

6.6. Effect of Unified PER

To evaluate the effectiveness of the proposed Unified PER mechanism, an ablation study was conducted by comparing HydraLight equipped with the standard PER against HydraLight using the proposed Unified PER framework under the same hyperparameter configurations and traffic scenarios. The experimental findings show that conventional PER tends to disproportionately prioritize samples from relatively simpler yet high-density traffic scenarios, which can negatively affect training stability and limit generalization performance on more structurally complex road networks. In contrast, the proposed Unified PER strategy achieves a more balanced replay distribution by normalizing TD-error values according to scenario difficulty, enabling the model to learn more consistently across diverse traffic environments and improving overall robustness.

Table 8 and Figure 8 present the quantitative comparison between standard PER and Unified PER. Unified PER reduces the average trip time from 214.70 s to 208.63 s compared with standard PER, yielding an average improvement of 2.83%. The benefit is most pronounced on *Arterial4x4*, where standard PER increases average trip time by 13.57% relative to the full model. The only exception is *Ingolstadt21*, where standard PER obtains 256.55 s compared with 271.04 s for Unified PER. This behavior is consistent with the motivation for Unified PER: raw PER may allocate disproportionate replay probability to a difficult map and improve that map in isolation, but it sacrifices multi-scenario stability, as shown by worse results on four of the five scenarios and a higher overall average. Thus, the unified, difficulty-normalized replay strategy is necessary for robust multi-scenario training, especially when the training set mixes regular grid networks with structurally complex arterial and real-world maps.

Table 8. Ablation comparison of HydraLight with Unified PER, without Global Pooling, and with standard PER. Values report average trip time in seconds; lower is better. Bold values are the best ones.

Variant	Grid4 × 4	Grid5 × 5	Arterial4 × 4	Ingolstadt21	Cologne8	Average
HydraLight (Unified PER)	**159.11**	**221.77**	**303.90**	271.04	**87.32**	**208.63**
w/o Global Pooling	164.48	234.33	334.64	286.41	87.81	221.53
HydraLight + Standard PER	159.59	222.76	345.15	**256.55**	89.47	214.70

7. Discussion

In order to go beyond aggregate performance scores, this section focuses on the operational boundaries and deployment limitations of HydraLight. Although the proposed framework is effective across heterogeneous scenarios, real-world constraints such as sensing noise, communication reliability, and computational latency define practical limits that should be explicitly acknowledged.

While the proposed architecture demonstrates strong performance and robustness, it is important to consider its *error characteristics and operational boundaries* to better understand its practical limitations. In particular, performance may deteriorate under highly unusual or extreme network states that were not represented during training, such as sudden traffic bursts or anomalous congestion patterns. Additionally, the effectiveness of the model is closely tied to the accuracy of input observations, making it susceptible to noise or incomplete data from real-world sensors. From a computational perspective, the integration of GAT and Transformer components introduces additional latency compared to simpler heuristic-based approaches, which may pose challenges in time-critical applications. Furthermore, although the model is trained across multiple scenarios to enhance generalization, it may still struggle to adapt to rare or unseen topological edge cases without targeted fine-tuning. These limitations manifest in several observable behaviors.

One notable issue is attention collapse under saturation. In conditions of extreme, full-network saturation (i.e., gridlock), the attention weights tend to become uniform across nodes. As a result, the model loses its ability to effectively prioritize critical intersections, reducing its capacity to distinguish between severe bottlenecks and less significant congestion points. Another important observation relates to the impact scaling with network complexity. The effectiveness of the Global Pooling Context increases non-linearly with the size and irregularity of the traffic network. In smaller and more regular topologies, such as *Grid 4x4*, the performance improvement remains relatively limited (approximately 2.1%), as local information propagation is already sufficient. However, as the network becomes larger (e.g., *Grid 5x5*) or structurally more complex (e.g., *Arterial 4x4*), the performance gains increase significantly (up to over 13%), indicating that the Global Pooling mechanism becomes increasingly beneficial in large-scale and heterogeneous environments.

In order to maintain the reliability of interpretation of our findings, it is essential to clarify the *scope and deployment limitations* of the proposed framework. First, all experiments are conducted in the SUMO simulator, which, despite its realism, cannot fully capture the stochasticity, sensor noise, and unpredictable disturbances present in real-life traffic environments. Second, the framework assumes reliable data collection from all intersections; however, in practice, communication delays and data loss may degrade system performance. Furthermore, the increased computational complexity of the proposed architecture, particularly due to the use of GAT and Transformer layers, may introduce latency constraints in edge deployments with limited hardware capabilities. These considerations give rise to several practical challenges, which are discussed below.

A primary concern is the sim-to-real gap. The experimental setup assumes ideal sensing conditions, such as accurate lane-level queue detection provided by the simulator. In real-world deployments, however, sensors such as inductive loops and cameras are prone to noise, occlusion, and measurement errors. This discrepancy may affect the reliability of the Global Pooling Context, whose sensitivity to imperfect input data remains an open research question that requires validation through field experiments.

Another limitation arises from the dependence on communication reliability. The architecture assumes that the global context information is transmitted instantaneously and without loss to all agents at regular intervals. While this assumption may hold under advanced communication infrastructures such as 5G or vehicle-to-everything (V2X) systems, real-world networks may experience latency, jitter, or packet loss. Such issues can lead to inconsistencies in the shared global state, potentially causing unstable or oscillatory control behaviors.

Although HydraLight is capable of dynamically learning traffic-aware attention coefficients, the current version of the study does not include a detailed visualization of these attention distributions across intersections, mainly due to space constraints. Nevertheless, understanding how the model allocates its attention under varying traffic conditions remains an important aspect of interpretability. Therefore, future work will focus on incorporating attention-map visualizations to provide deeper insight into how HydraLight identifies and prioritizes congested neighboring intersections, particularly in complex and heterogeneous urban traffic scenarios.

Finally, the scalability of the global context mechanism introduces additional constraints. The use of a global pooling layer requires aggregating information from the entire network and distributing it back to all agents, which can become a communication bottleneck as the network size increases. Moreover, centralized training with a shared experience buffer leads to a memory footprint that grows linearly with the number of intersections, posing challenges for deployment in large-scale metropolitan environments.

8. Conclusions

In this paper, we introduced HydraLight, a novel global-context spatio-temporal graph transformer framework for scalable multi-agent traffic signal control. The proposed architecture addresses the fundamental limitations of existing Traffic Signal Control systems by jointly modeling spatial dependencies, long-term temporal dynamics, and network-wide coordination within a unified learning framework. Specifically, HydraLight integrates Graph Attention Networks for localized spatial reasoning, a Temporal Transformer for sequence modeling under partial observability, and a Global Context Pooling mechanism to overcome the local myopia problem by broadcasting system-level congestion signals.

To further enhance robustness across heterogeneous environments, we proposed the Unified Prioritized Experience Replay mechanism, which normalizes Temporal-Difference errors at the scenario level. This ensures balanced learning across diverse traffic topologies and prevents the dominance of simpler scenarios during training. Extensive experiments conducted on multiple synthetic and real-world traffic networks demonstrate that HydraLight consistently outperforms state-of-the-art baselines, particularly in complex and irregular urban settings, achieving significant reductions in average travel time and improved network-wide efficiency. Experimental evaluation on the RESCO benchmark confirmed that HydraLight yields statistically significant improvements in complex environments, achieving a 13.07% reduction in average travel time on arterial networks and consistent gains on real-world maps like Ingolstadt21. Beyond improving operational traffic efficiency, the proposed framework also contributes to sustainable urban mobility by reducing unnecessary vehicle idling, mitigating congestion-related fuel consumption, and supporting lower transportation-related carbon emissions in densely populated smart-city environments.

Despite these promising results, several limitations remain. First, the experimental evaluation is conducted within a simulated environment (SUMO), which cannot fully capture real-world uncertainties such as sensor noise, communication delays, and unexpected disruptions. Second, the computational complexity introduced by the combination of GAT and Transformer components may pose challenges for real-time deployment in resource-constrained edge environments. Nevertheless, HydraLight represents the proposed system is a significant step toward intelligent, scalable, and cooperative traffic signal control systems, contributing to the development of more efficient and sustainable urban transportation infrastructures. In this context, HydraLight demonstrates the potential of Artificial Intelligence-driven traffic management systems to support environmentally sustainable smart-city ecosystems by improving transportation efficiency while simultaneously addressing urban energy consumption and environmental impact challenges.

Future work will focus on addressing these limitations by exploring real-world deployment scenarios, including integration with live traffic data and edge-computing optimization. Additionally, we plan to investigate lightweight model compression techniques to reduce inference latency, as well as adaptive communication strategies to further enhance scalability in large-scale urban networks. Another promising direction involves extending the framework toward multimodal traffic intelligence, incorporating data from connected vehicles, cameras, and IoT sensors to improve situational awareness. Finally, incorporating meta-learning and transfer learning strategies could enable rapid adaptation of HydraLight to unseen cities with minimal retraining. Future sustainability-oriented extensions may also include integrating energy-aware traffic optimization objectives, carbon-emission estimation models, and eco-driving coordination mechanisms to further support greener and more resilient urban transportation systems.

Author Contributions: Conceptualization, A.D. and G.Y.; methodology, A.D. and G.Y.; software, A.D.; validation, E.C.B. and O.K.S.; formal analysis, O.K.S.; investigation, E.C.B. and G.Y.; resources, E.C.B. and O.K.S.; data curation, E.C.B. and O.K.S.; writing—original draft preparation, O.K.S., A.D. and G.Y.; writing—review and editing, A.D., G.Y., E.C.B. and O.K.S.; visualization, E.C.B. and A.D.; supervision, O.K.S. and G.Y.; project administration, O.K.S. and G.Y.; funding acquisition, G.Y. and E.C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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. Zhu, Y.; Wen, H.; Min, G.; Luo, M. HALO: Hierarchical Reinforcement Learning for Large-Scale Adaptive Traffic Signal Control. In Proceedings of the ACM Web Conference 2026 (WWW '26), Dubai, United Arab Emirates, 13–17 April 2026. [CrossRef]
2. Xiao, H.; Li, H.; Qi, S.; Zhang, J.; Cai, D. FGLight: Learning Neighbor-level Information for Traffic Signal Control. In Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, MI, USA, 19–23 May 2025; pp. 2181–2189. [CrossRef]
3. Jilani, U.; Asif, M.; Zia, M.Y.I.; Rashid, M.; Shams, S.; Otero, P. A Systematic Review on Urban Road Traffic Congestion. *Wirel. Pers. Commun.* **2025**, *140*, 81–109. [CrossRef]
4. Macherla, H.; Muvva, V.R.; Lella, K.K.; Palisetti, J.R.; Pulugu, D.; Vatambeti, R. A Deep Reinforcement Learning-Based RNN Model in a Traffic Control System for 5G-Envisioned Internet of Vehicles. *Math. Model. Eng. Probl.* **2024**, *11*, 75–83. [CrossRef]
5. Webster, F.V. *Traffic Signal Settings*; Road Research Technical Paper 39; Road Research Laboratory: London, UK; Her Majesty's Stationery Office: London, UK, 1958.
6. Radin, S.; Chajka-Cadin, L.; Futcher, E.; Badgley, J.; Mittleman, J. *Adaptive Signal Control*; Technical Report FHWA-HRT-17-007; Federal Highway Administration: Washington, DC, USA, 2018.
7. Figliozzi, M.; Monsere, C. *Evaluation of the Performance of the Sydney Coordinated Adaptive Traffic System (SCATS)*; Technical Report OTREC-RR-13-07; OTREC: Liberia, Costa Rica, 2013.
8. Xiao, F.; Lu, J.; Li, L.; Tu, W.; Li, C. Advances in reinforcement learning for traffic signal control: A review of recent progress. *Intell. Transp. Infrastruct.* **2025**, *4*, liaf009. [CrossRef]
9. Michailidis, P.; Michailidis, I.; Lazaridis, C.R.; Kosmatopoulos, E.B. Traffic Signal Control via Reinforcement Learning: A Review on Applications and Innovations. *Infrastructures* **2025**, *10*, 114. [CrossRef]
10. 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 (CIKM), Beijing, China, 3–7 November 2019; pp. 1913–1922. [CrossRef]
11. Guo, Q.; Li, X.; Chen, J.; Guo, Z.; Li, X.; Li, Z.; Li, L. A Dual Large Language Models Architecture with Herald Guided Prompts for Parallel Fine Grained Traffic Signal Control. *arXiv* **2025**. [CrossRef]
12. Wu, J.; Ran, Y.; Lou, Y.; Shu, L. CenLight: Centralized traffic grid signal optimization via action and state decomposition. *IET Intell. Transp. Syst.* **2023**, *17*, 1247–1261. [CrossRef]
13. Jiang, H.; Li, Z.; Wei, H.; Xiong, X.; Ruan, J.; Lu, J.; Mao, H.; Zhao, R. X-Light: Cross-City Traffic Signal Control Using Transformer-on-Transformer as Meta Reinforcement Learning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Jeju, Republic of Korea, 3–9 August 2024; pp. 3121–3129. [CrossRef]
14. Jiang, H.; Li, Z.; Li, Z.; Bai, L.; Mao, H.; Ketter, W.; Zhao, R. A General Scenario-Agnostic Reinforcement Learning for Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 11330–11344. [CrossRef]
15. Carcedo, J.M. Predicting traffic intensity in the urban area of Madrid: Integrating route network topology into a machine-learning model. *Eng. Appl. Artif. Intell.* **2024**, *137*, 109154. [CrossRef]
16. Xiao, H.; Li, H.; Wang, Z.; Yang, Z.; Qi, S.; Zhang, J.; Cai, D.; Yin, J. Intelligent traffic signal control based on reinforcement learning: A survey. *Artif. Intell. Rev.* **2026**, *59*, 128. [CrossRef]
17. Varaiya, P. The Max-Pressure Controller for Arbitrary Networks of Signalized Intersections. Available online: https://link.springer.com/chapter/10.1007/978-1-4614-6243-9_2 (accessed on 2 April 2026).
18. Li, L.; Lv, Y.; Wang, F.Y. Traffic Signal Timing via Deep Reinforcement Learning. *IEEE/CAA J. Autom. Sin.* **2016**, *3*, 247–254. [CrossRef]
19. Zheng, G.; Xiong, Y.; Zang, X.; Feng, J.; Wei, H.; Zhang, H.; Li, Y.; Xu, K.; Li, Z. Learning Phase Competition for Traffic Signal Control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), Beijing, China, 3–7 November 2019; pp. 1937–1946. [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 (KDD), Anchorage, AK, USA, 4–8 August 2019; pp. 1290–1298. [CrossRef]
21. Wu, L.; Wang, M.; Wu, D.; Wu, J. Dynamic Spatial-Temporal Graph Attention Network for Traffic Signal Control. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event, QLD, Australia, 1–5 November 2021; pp. 4682–4691.

22. Oroojlooy, A.; Nazari, M.; Hajinezhad, D.; Silva, J. AttendLight: Universal Attention-Based Reinforcement Learning Model for Traffic Signal Control. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020.
23. Liu, C.; Zhang, H.; Zhang, W.; Zheng, G.; Yu, Y. GeneraLight: Improving Environment Generalization of Traffic Signal Control via Meta Reinforcement Learning. In Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM), Virtual Event, Ireland, 19–23 October 2020; pp. 1713–1722. [CrossRef]
24. Zang, X.; Yao, H.; Zheng, G.; Xu, N.; Xu, K.; Li, Z. MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 1153–1160. [CrossRef]
25. Ruan, J.; Li, Z.; Wei, H.; Jiang, H.; Lu, J.; Xiong, X.; Mao, H.; Zhao, R. CoSLight: Co-Optimizing Collaborator Selection and Decision-Making to Enhance Traffic Signal Control. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Barcelona, Spain, 25–29 August 2024; pp. 2781–2790. [CrossRef]
26. Jiang, J.; Han, C.; Zhao, W.X.; Wang, J. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 4365–4373. [CrossRef]
27. Luo, Q.; Lu, X.; Zang, Z.; Gong, H.; Guo, X.; Chen, X. A Real-Time Early Warning Framework for Multi-Dimensional Driving Risk of Heavy-Duty Trucks Using Trajectory Data. *Systems* **2026**, *14*, 204. [CrossRef]
28. Zhai, C.; Wu, W.; Xiao, Y.; Zhang, J.; Zhai, M.; Wu, Y. A novel throttle-based self-stabilizing control scheme integrated into an anisotropic continuum model to mitigate cyber-attacks in connected vehicle scenarios. *Chaos Solitons Fractals* **2025**, *201*, 117319. [CrossRef]
29. Zhao, H.; Dong, C.; Cao, J.; Chen, Q. A survey on deep reinforcement learning approaches for traffic signal control. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108100. [CrossRef]
30. Veličković, P.; Casanova, A.; Liò, P.; Cucurull, G.; Romero, A.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
31. Van Hasselt, H.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. *Proc. AAAI Conf. Artif. Intell.* **2016**, *30*. [CrossRef]
32. Wang, Z.; Schaul, T.; Hessel, M.; Hasselt, H.V.; Lanctot, M.; de Freitas, N. Dueling Network Architectures for Deep Reinforcement Learning. *arXiv* **2016**. [CrossRef]
33. Hausknecht, M.; Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. *arXiv* **2015**. [CrossRef]
34. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**. [CrossRef]
35. Hessel, M.; Soyer, H.; Espeholt, L.; Czarnecki, W.; Schmitt, S.; Van Hasselt, H. Multi-Task Deep Reinforcement Learning with PopArt. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 3796–3803. [CrossRef]
36. Franceschi, L.; Donini, M.; Perrone, V.; Klein, A.; Archambeau, C.; Seeger, M.; Pontil, M.; Frasconi, P. Hyperparameter Optimization in Machine Learning. *Found. Trends Mach. Learn.* **2025**, *18*, 975–1109. [CrossRef]
37. Ilemobayo, J.A.; Durodola, O.I.; Alade, O.; Awotunde, O.J.; Olanrewaju, A.T.; Falana, O.; Ogungbire, A.; Osinuga, A.; Ogunbiyi, D.; Ifeanyi, A.; et al. Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *J. Eng. Res. Rep.* **2024**, *26*, 388–395. [CrossRef]
38. Zhu, X.; Zhang, Y.; Ying, H.; Chi, H.; Sun, G.; Zeng, L. Modeling epidemic dynamics using Graph Attention based Spatial Temporal networks. *PLoS ONE* **2024**, *19*, e0307159. [CrossRef]
39. Qadri, S.S.S.M.; Gökçe, M.A.; Öner, E. State-of-art review of traffic signal control methods: Challenges and opportunities. *Eur. Transp. Res. Rev.* **2020**, *12*, 55. [CrossRef]
40. Zhu, F.; Li, G.; Li, Z.; Chen, C.; Ding, W. A Case Study of Evaluating Traffic Signal Control Systems Using Computational Experiments. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1220–1226. [CrossRef]
41. Krajzewicz, D. Traffic Simulation with SUMO–Simulation of Urban Mobility. In *Fundamentals of Traffic Simulation*; International Series in Operations Research & Management Science; Barceló, J., Ed.; Springer: New York, NY, USA, 2010; Volume 145. [CrossRef]
42. Lu, J.; Ruan, J.; Jiang, H.; Li, Z.; Mao, H.; Zhao, R. DuaLight: Enhancing Traffic Signal Control by Leveraging Scenario-Specific and Scenario-Shared Knowledge. In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, 6–10 May 2024. [CrossRef]
43. Eom, M.; Kim, B. The traffic signal control problem for intersections: A review. *Eur. Transp. Res. Rev.* **2020**, *12*, 50. [CrossRef]
44. Roess, R.P.; Prassas, E.S.; McShane, W.R. *Traffic Engineering*, 3rd ed.; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2004.
45. Ault, J.; Hanna, J.; Sharon, G. Learning an Interpretable Traffic Signal Control Policy. In Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, 9–13 May 2020; pp. 88–96. [CrossRef]
46. Ault, J.; Sharon, G. Reinforcement Learning Benchmarks for Traffic Signal Control. In Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS 2021) Datasets and Benchmarks Track, Online, 6–14 December 2021.

47. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; WU, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Proceedings of the Advances in Neural Information Processing Systems*; Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 24611–24624.
48. 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. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 3414–3421. [CrossRef]
49. Lou, Y.; Wu, J.; Ran, Y. Meta-Reinforcement Learning for Multiple Traffic Signals Control. In Proceedings of the CIKM '22 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 4264–4268. [CrossRef]