

A Dual Large Language Models Architecture with Herald Guided Prompts for Parallel Fine Grained Traffic Signal Control

Qing Guo*, Xinhang Li*, Junyu Chen*, Zheng Guo*,
Xiaocong Li[†], Lin Zhang^{*,†}, Lei Li*

* School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China

[†] Beijing Big Data Center, Beijing, China

[‡] Eastern Institute of Technology, Ningbo, China

Abstract—Leveraging large language models (LLMs) in traffic signal control (TSC) improves optimization efficiency and interpretability compared to traditional reinforcement learning (RL) methods. However, existing LLM-based approaches are limited by fixed time signal durations and are prone to hallucination errors, while RL methods lack robustness in signal timing decisions and suffer from poor generalization. To address these challenges, this paper proposes HeraldLight, a dual LLMs architecture enhanced by Herald guided prompts. The Herald Module extracts contextual information and forecasts queue lengths for each traffic phase based on real-time conditions. The first LLM, LLM-Agent, uses these forecasts to make fine grained traffic signal control, while the second LLM, LLM-Critic, refines LLM-Agent’s outputs, correcting errors and hallucinations. These refined outputs are used for score-based fine-tuning to improve accuracy and robustness. Simulation experiments using CityFlow on real world datasets covering 224 intersections in Jinan (12), Hangzhou (16), and New York (196) demonstrate that HeraldLight outperforms state of the art baselines, achieving a 20.03% reduction in average travel time across all scenarios and a 10.74% reduction in average queue length on the Jinan and Hangzhou scenarios. The source code is available on GitHub: <https://github.com/BUPT-ANTlab/HeraldLight>.

Index Terms—Traffic Signal Control, Large Language Models, Parallel Decision Making

I. INTRODUCTION

Advancements in artificial intelligence (AI) and sensor technologies are rapidly propelling the evolution of Intelligent Transportation Systems (ITS), offering significant potential to improve traffic efficiency and reduce congestion. Among these developments, Traffic Signal Control (TSC) has become a crucial solution for optimizing traffic efficiency [1]. Most current approaches rely on fixed timing strategies, which lack adaptability to varying traffic conditions, limiting system efficiency. In contrast, dynamic timing requires managing additional decision variables and precise control to respond to real-time traffic fluctuations. Addressing such challenges is essential for improving urban traffic management and optimizing the performance of transportation systems.

Reinforcement learning (RL) has been extensively utilized in traffic signal control (TSC) as an effective approach for optimizing traffic flow and alleviating congestion. Xu et al. proposed HiLight, a hierarchical RL framework designed for short-term traffic optimization [2]. Gu et al. introduced π -Light, an interpretable RL model tailored for resource-limited settings [3]. Liang et al. proposed OAM for multi-intersection control [4]. RL methods has demonstrated effectiveness in decentralized multi-agent systems. However, most RL methods adopt a fixed timing strategy that limits flexibility, while RL methods employing dynamic timing often provide limited action-level justification and exhibit weak generalization.

Recent advancements have shown that large language models can address interpretability issues in RL methods [5]. By leveraging the reasoning capabilities of LLMs, traffic signal control systems can benefit from transparent and consistent decision-making [6], [7]. Approaches like PromptGAT have been used to adapt RL-trained policies to real-world traffic conditions [8], while Open-TI facilitated coordinated traffic analysis and policy training [9]. Integration frameworks such as iLLM-TSC combined RL with LLMs to manage incomplete state information for real-time adjustments [10]. However, challenges persist in adapting to dynamic traffic signal timing, particularly in complex urban environments. Additionally, the occurrence of hallucinations of LLMs may lead to inaccurate decisions [11], thereby undermining reliability. Addressing these limitations is necessary to make LLMs based TSC systems more adaptive and efficient for urban traffic management.

Existing studies have attempted to tackle the high dimensionality of the action space in dynamic signal timing. Wang et al. proposed UniTSA for single-intersection control and reported fixed time variants (Fix-30/Fix-40) [12]. Zhang et al. introduced DynamicLight, a two-stage dynamic timing method that first selects the active phase and then sets its duration [13]. Kim et al. developed a deep RL strategy for prioritized phase split optimization, dynamically adjusting signal durations to improve network efficiency [14]. Above methods improve the accuracy of action selection for TSC. However, the limited interpretability and weak generalization of existing methods

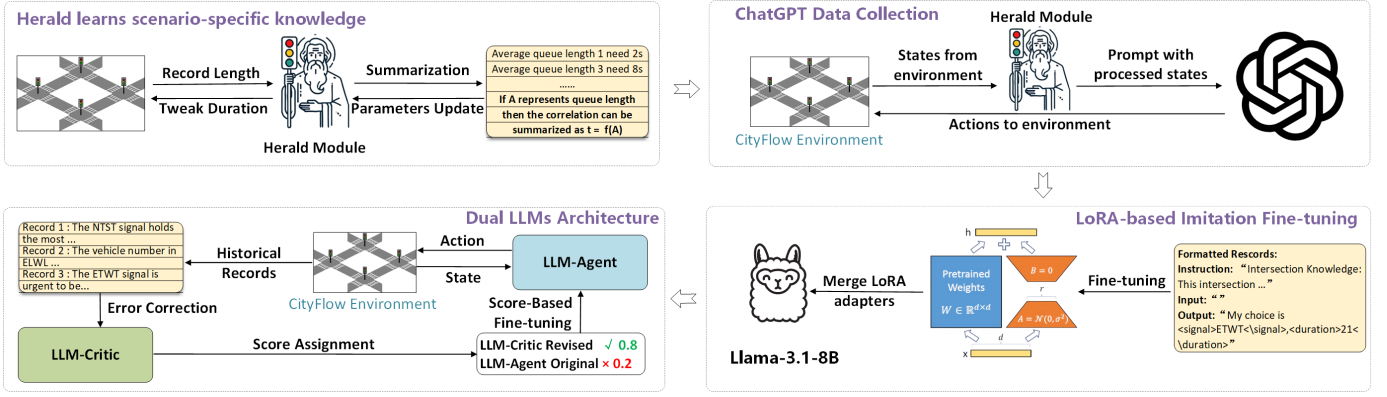


Fig. 1. Overview of HeraldLight. Herald Module learns and updates; ChatGPT assisted with Herald Module curates imitation data; Llama-3.1-8B is LoRA-finetuned to instantiate the LLM-Agent; an LLM-Critic (ChatGPT) evaluates and revises decisions; scored corrections drive iterative, score-based fine-tuning.

constrain the broader adoption of dynamic signal timing.

This paper presents HeraldLight, a dual LLMs architecture driven by Herald guided prompts. The LLM-Agent, enhanced via LoRA-based imitation fine-tuning, infers the active signal phase and duration from real-time traffic conditions. The LLM-Critic, informed by Herald, evaluates and revises LLM-Agent’s outputs to mitigate hallucinations and improve consistency and reliability. An overview is provided in Fig. 1, and the contributions are summarized as follows:

- The Herald Module evaluates intersection traffic flow by extracting environmental features and projecting queue states up to 40 seconds ahead. By precisely predicting queue lengths and accurately modeling the relationship among environmental factors, signal phase durations, and queue lengths, it enables dynamic traffic signal control with second-level fine grained adjustments.
- This paper proposes a collaborative agent-critic architecture that integrates two LLMs to advance environmental analysis and reasoning in TSC. Through iterative interactions between the LLM-Agent and the LLM-Critic, the architecture progressively mitigates hallucination and systematically enhances the reasoning capabilities required for dynamic TSC strategies.
- All experiments are conducted in the CityFlow simulator using three open-source real-world datasets. Jinan (1–3), Hangzhou (1–2), and New York (1–2). Compared with state-of-the-art method, the proposed approach achieves a 20.03% reduction in Average Travel Time (ATT) across all scenarios and a mean 10.74% reduction in Average Queue Length (AQL) on the Jinan and Hangzhou networks.

The rest of this paper is organized as follows: Section 2 describes the intersection modeling. Section 3 presents the composition of the Herald Module. Section 4 outlines HeraldLight framework including prompt design, imitation fine-tuning and Dual LLMs architecture. Section 5 presents the experimental setup and results. Section 6 concludes with a summary and future directions.

II. INTERSECTION MODELING

A. Lanes, Phases, and Duration

Lanes. For each direction $d \in \{N, S, E, W\}$, incoming lanes are grouped by movement $L_d^{\text{in}} = \{L_d^l, L_d^s, L_d^r\}$ (left/straight/right), with corresponding outgoing lanes L_d^{out} . Right-turn lanes are treated as permissive (unsignalized).

Phases. Let $P = \{P_1, P_2, P_3, P_4\}$. Each phase controls two lanes (two opposing approaches): P_1, P_2 serve the N–S approaches (straight/left), and P_3, P_4 serve the E–W approaches (straight/left).

Duration / Action. At each decision step, the agent selects a phase and its duration (green time):

$$a = (P_i, t_i), \quad P_i \in P, \quad t_i \in (0, 40] \text{ s},$$

while yellow and all-red intervals are fixed constants and not part of the action space. A detailed illustration is shown in Appendix A.1

III. HERALD MODULE

A. Herald Learns Scenario-Specific Knowledge

Herald augments LLM-based traffic signal control by learning scenario-specific dynamics from simulation. Herald adjusts phase durations as a function of queue length and logs key metrics to support short-horizon traffic prediction.

For each approach, Herald records maximum speed, mean egress time, and the release time associated with a given queue length (the interval from the departure of the first vehicle to the moment the last vehicle is entirely outside the incoming lane). Measuring release time over a range of queue lengths yields a practical mapping from queue length to release time while abstracting vehicle-level details (e.g., length, acceleration).

Based on the collected data, a monotonic mapping from queue length to release time is estimated and represented with a piecewise-linear model to balance fidelity and parsimony. The mapping supports second-level, fine grained signal control and guides the LLMs in selecting phase durations under varying traffic conditions.

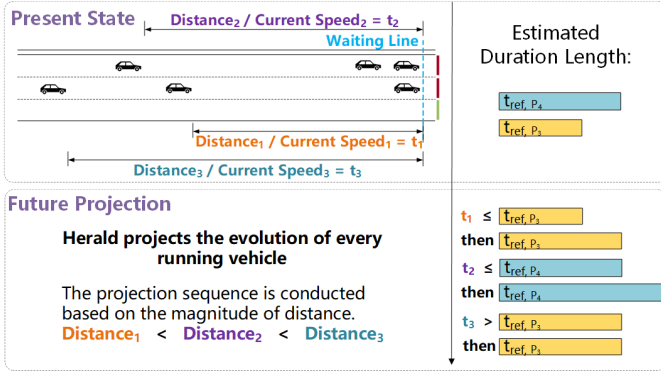


Fig. 2. Herald Module Assist for Duration Calculation

B. Herald Module Assist Process for Duration Selection

Herald Module determines signal phase durations by combining a queue-based release time model with predicted arrivals of running vehicles.

At each timestep, for every phase P_i ($i = 1, \dots, 4$), a reference release time t_{ref, P_i} is obtained from the learned mapping from queue length to release time under the current queue state, providing the baseline duration for phase P_i .

Running vehicles approaching the intersection are then evaluated. For each vehicle v associated with phase P_i , the time-to-waiting-line τ_v is estimated from current distance and speed. A vehicle v is admitted to phase P_i if $\tau_v \leq t_{\text{ref}, P_i}$. Equivalently,

$$\mathcal{A}_i = \{v \in \mathcal{V}_i : \tau_v \leq t_{\text{ref}, P_i}\}.$$

Phase P_i is extended to serve vehicles in $\mathcal{A}_i$ (admitted set); otherwise, the baseline duration is retained. As illustrated in Fig. 2

By accounting for both queued and approaching vehicles, Herald supplies assistive, online phase duration estimates to distributed LLM-based agents as conditions evolve.

IV. HERALDLIGHT FRAMEWORK

The phase selection prompt is a structured template composed of six blocks: $d_{\text{intersection}}$ (static geometry and lane-phase mapping), d_{task} (objectives: phase choice and duration setting), d_{details} (computational rules), d_{req} (I/O constraints and formatting), d_{rules} (fallbacks for edge cases), and $d_{\text{important}}$ (priority constraints to enforce consistency).

A. Task Description (d_{task})

Two objectives are addressed: (i) select the next signal phase and (ii) determine its duration. Herald provides the quantities needed by the LLMs to reason over current and predicted demand.

1) *Signal Phase Selection*: Let P_i ($i = 1, \dots, 4$) denote the phases and $Q_{P_i,1}, Q_{P_i,2}$ the lane queues served by P_i . The total demand for phase i is

$$Q_i = Q_{P_i,1} + Q_{P_i,2}, \quad (1)$$

with $Q_i = \max(Q_{P_i,1}, Q_{P_i,2})$ when one lane is empty. Lane imbalance is quantified by

$$I_i = |Q_{P_i,1} - Q_{P_i,2}|. \quad (2)$$

Phase priority is computed from (Q_i, I_i) . Two queue sources are available: a Herald prediction (forward looking) and an Original measurement (current state). The predictive source is used by default; the original measurement is invoked by d_{rules} when prediction implies implausible durations or excessive imbalance.

2) *Duration Determination*: Herald supplies a mapping from queue length to release time, yielding a reference duration for each phase. For phase P_i , the reference duration is

$$t_{\text{ref}} = A\tau + \delta, \quad A = \max(Q_{P_i,1}, Q_{P_i,2}), \quad (3)$$

where τ is the mean per-vehicle egress time and δ corrects for tail speed-up effects. Then piecewise adjustments are conducted to refine the reference value:

$$t_{\text{ref}} = \begin{cases} t_{\text{ref}} + \kappa_{\text{over}}, & \text{if } t_{\text{ref}} \geq T_{\text{over}}, \\ t_{\text{ref}} + \kappa_{\text{ineff}}, & \text{if } t_{\text{ref}} = T_{\text{ineff}}, \\ \vdots & \\ t_{\text{ref}}, & \text{otherwise.} \end{cases} \quad (4)$$

Thresholds $T_{\text{over}}, T_{\text{ineff}}$ and gains $\kappa_{\text{over}}, \kappa_{\text{ineff}}$ are provided by Herald Module. Durations are computed under both predictive (*Herald*) and measured (*Original*) queues, and the final actions are selected by LLM-based distributed agents consistently across phases to balance throughput and fairness.

B. Distributed Dual LLMs Agents

In multi-intersection settings, a distributed, parallel inference architecture deploys a population of agents. Task-specific agents are constructed via LoRA-based fine-tuning combined with dual LLMs error-corrective learning, and execute parallel inference at the action decision time of the dynamic signal-timing loop for each intersection, enabling low-latency, consistent control across all scenarios [15].

1) *Imitation Fine-tuning with LoRA*: An online LLM (ChatGPT) interacts with CityFlow through Herald-assisted prompts. Herald Module encodes the evolving simulator state into compact, model-aligned text. The LLM outputs actions in a schema (e.g., `<signal>ETWT</signal>` `<duration>15</duration>`); a lightweight parser extracts the phase and duration and executes them in CityFlow, enabling closed-loop control.

Each step logs the structured prompt, model response, and executed action to build an imitation corpus. Low-Rank Adaptation (LoRA) [16] inserts low-rank adapters into selected layers and updates parameters, reducing computation and storage while preserving the base model. Applied to Llama-3.1-8B, LoRA yields a task-specialized LLM-Agent for traffic-signal control with modest resource demands and improved inference efficiency.

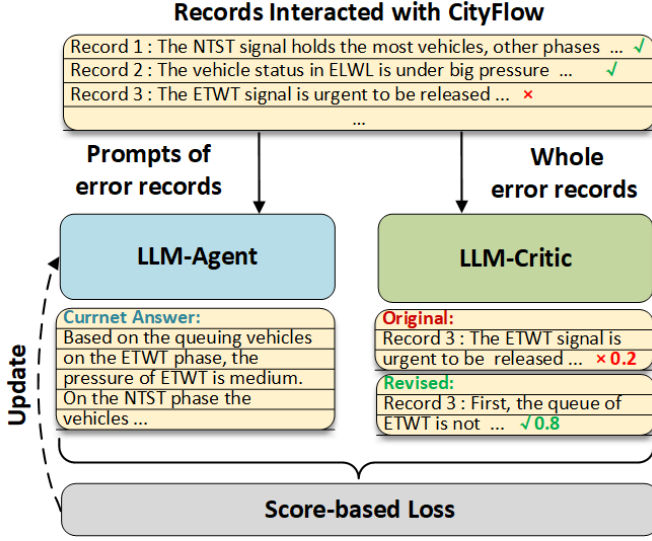


Fig. 3. Score-Based Dual LLMs Correction

2) *Dual LLMs architecture*: To mitigate hallucination in large models [11], a dual LLMs architecture is adopted: LLM-Agent generates actions; LLM-Critic (a stronger reasoner, instantiated with the ChatGPT) evaluates outputs, proposes corrections, and assigns quality scores (shown in Fig. 3). Interaction logs from CityFlow provide prompts, agent responses, and outcomes. The critic identifies erroneous cases, produces corrected responses, and pairs {original, corrected} with scores to create preference-style training data.

Let Y_i denote a revised (critic-corrected) reasoning trajectory for input X , and π_θ the agent policy. The average token log-likelihood is

$$p_i = \frac{1}{|Y_i|} \sum_{w \in Y_i} \log \pi_\theta(y_{i,w} | X, Y_i < w). \quad (5)$$

A score-based ranking loss [17] prioritizes higher-scored trajectories over lower-scored ones:

$$L_{\text{score}} = \log \left(1 + \sum_{q_i > q_j} \left[e^{(p_j - p_i)} + e^{(2p_{j^*} - 2\beta - p_i - p_j)} \right] \right), \quad (6)$$

where q_i is the critic-assigned quality ($Q_{\text{error/corrected}}$) for Y_i , $p_{j^*} = \min_{q_k > q_j} p_k$ is the least favorable higher-rated path, and β is a margin hyperparameter. The first term increases preference for higher-rated trajectories; the second term prevents performance degradation.

Training alternates among interaction data collection, critic-guided correction, and updates of the LLM-Agent using L_{score} . Leveraging erroneous and corrected actions aligns the agent with critic-calibrated behavior, reducing hallucinations and improving decision reliability. The overall procedure is shown in Algorithm 1. This regimen improves robustness when the LLM-Agent performs parallel inference across multiple intersections with asynchronous decision steps.

Algorithm 1: Score-Based Dual LLMs Decision and Fine-tuning Process

Input: State $s(t)$, Pre-trained $\text{LLM}_{\text{Agent}}$, $\text{LLM}_{\text{Critic}}$
Output: Updated $\text{LLM}_{\text{Agent}}$

Initialize: Simulation environment and $\text{LLM}_{\text{Agent}}$;

- 1 **for** each episode **do**
- 2 $s(t) \leftarrow$ Environment Initialization;
- 3 Prompt $\leftarrow$ Conversion($s(t)$);
- 4 **for** $t = 1$ to T **do**
- 5 $a_1(t) \leftarrow \text{LLM}_{\text{Agent}}$ (Prompt);
- 6 Execute($a_1(t)$) $\leftarrow s(t+1)$;
- 7 Record(Prompt, $a_1(t)$); **if** $a_1(t)$ fails **then**
- 8 Mark($a_1(t)$, Hallucinated);
- 9 **end**
- 10 **for** each hallucinated $a_1(t)$ **do**
- 11 $a_2(t) \leftarrow \text{LLM}_{\text{Critic}}$ (Prompt, $a_1(t)$);
- 12 Add to Fine-tuning Dataset ((Prompt, $a_1(t)$), $a_2(t)$);
- 13 **end**
- 14 **for** each ($a_1(t)$, $a_2(t)$) **do**
- 15 $q_1 \leftarrow Q_{\text{error}}$; $q_2 \leftarrow Q_{\text{corrected}}$;
- 16 $L_{\text{score}} \leftarrow \log \left(1 + \sum_{q_i > q_j} \left[e^{(p_j - p_i)} + e^{(2p_{j^*} - 2\beta - p_i - p_j)} \right] \right)$;
- 17 Finetune($\text{LLM}_{\text{Agent}}$);
- 18 **end**
- 19 **end**

V. EXPERIMENTS AND RESULTS

The evaluation of HeraldLight is organized around three research questions:

RQ1: Effectiveness relative to a state-of-the-art method.

RQ2: Generalization across cities and traffic densities (scalability and transferability).

RQ3: Robustness to hallucinations and interpretability.

A. Datasets and Scenario Configuration

Benchmarks use CityFlow datasets [18] from three regions (datasets: <https://traffic-signal-control.github.io/#open-datasets>).

Jinan (Dongfeng; Fig. 4(a)): 12 intersections; vehicles: 6295/4365/5494.

Hangzhou (Gudang; Fig. 4(b)): 16 intersections; vehicles: 2983/6984.

New York (Upper East Side; taxi-trip derived; Fig. 4(c)): 196 intersections; vehicles: 11058/16337.



Fig. 4. Visualizations of road networks in three study areas.

TABLE I
RESULTS ON **JINAN** (JN-1/2/3) AND **HANGZHOU** (HZ-1/2) ACROSS THREE METRICS. ALL VALUES ARE ROUNDED TO TWO DECIMAL PLACES. THE BEST RESULTS ARE IN **BOLD**, THE SECOND-BEST ARE UNDERLINED, AND THE THIRD-BEST ARE DOUBLE UNDERLINED.

Method	JN-1			JN-2			JN-3			HZ-1			HZ-2		
	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT
Transportation Methods															
Random	584.08	657.86	97.16	541.95	405.17	90.44	548.53	516.38	101.37	605.63	287.78	93.52	524.12	457.20	109.40
FixedTime	481.79	491.03	70.99	441.19	294.14	66.72	450.11	394.34	69.19	616.02	301.33	73.99	486.69	425.12	72.80
MaxPressure	282.58	170.71	44.53	273.20	106.58	<u>38.25</u>	265.75	133.90	40.20	325.33	68.99	49.60	347.74	215.53	70.58
RL Methods															
MPLight	299.53	202.29	92.34	301.40	138.34	94.73	284.60	161.79	84.02	343.81	83.34	93.88	368.36	255.73	112.41
AttendLight	296.97	195.48	67.72	281.29	115.36	56.63	272.56	143.46	56.11	324.43	67.41	58.78	351.12	223.73	67.89
PressLight	288.91	182.28	51.47	276.70	111.61	47.06	274.58	145.45	41.71	347.39	85.11	84.45	395.84	297.85	129.80
CoLight	284.26	175.84	61.80	274.68	108.41	51.68	265.67	134.11	50.41	319.92	64.67	55.84	339.79	206.65	81.94
Efficient-CoLight	276.89	163.20	42.60	268.85	102.35	39.35	261.85	129.06	40.44	312.14	58.26	<u>36.33</u>	328.65	181.81	56.27
Advanced-CoLight	<u>273.20</u>	<u>158.65</u>	47.26	267.56	<u>101.26</u>	40.81	<u>260.16</u>	127.04	42.84	<u>304.54</u>	<u>52.97</u>	<u>40.67</u>	<u>322.86</u>	<u>173.38</u>	69.93
DynamicLight	<u>238.61</u>	<u>83.48</u>	27.35	<u>221.91</u>	<u>32.46</u>	16.55	<u>220.07</u>	<u>50.23</u>	20.95	<u>264.73</u>	<u>13.02</u>	12.33	<u>314.73</u>	146.06	<u>53.99</u>
Large-Scale AI Models															
Llama-2-13B	401.92	368.41	123.61	406.44	254.05	150.78	396.29	324.13	148.24	468.79	180.01	183.17	425.96	342.27	143.18
ChatGPT-3.5	315.95	222.40	53.72	279.83	112.77	43.62	288.98	163.13	49.49	324.55	68.25	46.16	340.29	196.38	<u>55.28</u>
ChatGPT-4	276.96	164.33	48.81	<u>263.82</u>	133.04	47.58	271.78	<u>105.60</u>	45.12	314.91	59.93	53.27	333.57	191.38	<u>67.02</u>
ChatGPT-4o-mini	275.00	161.06	44.83	<u>267.01</u>	101.03	41.33	261.14	<u>128.73</u>	42.89	313.22	58.78	42.10	332.47	186.75	61.01
DeepSeek-R1-671B	276.78	163.44	46.05	271.18	105.56	40.35	261.82	129.38	41.52	312.09	57.68	45.72	331.99	186.83	72.93
ERNIE-4.0-8K	275.93	161.90	44.07	270.20	103.94	38.49	261.71	129.05	41.83	311.54	57.82	43.67	333.22	187.99	57.40
ERNIE-Lite	301.51	202.62	63.10	292.92	126.50	61.81	285.13	159.83	59.98	328.38	70.02	67.58	346.60	215.39	84.63
ERNIE-Speed-8K	313.75	222.46	87.56	301.87	137.58	83.14	287.35	163.82	76.14	338.21	78.55	99.49	352.92	230.70	99.38
Hunyuan-Lite	1135.71	1431.58	950.23	1299.84	1173.71	1004.29	1174.65	1261.17	980.14	1266.47	712.10	1041.25	810.36	795.93	872.42
Qwen-Long	288.67	179.99	49.41	276.50	110.38	45.80	270.24	140.28	45.11	319.71	64.22	48.27	334.90	189.07	50.90
Spark-Lite	569.47	670.15	320.70	535.76	416.11	272.29	514.62	505.74	249.83	554.95	255.74	244.67	451.81	383.45	186.11
LLM-Based Methods															
LLMLight	277.01	162.56	<u>41.77</u>	269.07	103.23	43.33	261.42	128.95	42.12	313.72	59.58	51.36	336.99	194.86	70.87
Traffic-R1	277.69	164.74	<u>45.70</u>	270.11	104.32	41.78	260.12	125.91	<u>40.18</u>	311.01	57.57	40.99	331.68	185.12	58.95
HeraldLight	234.64	73.69	<u>40.55</u>	220.78	28.58	<u>27.63</u>	217.69	44.15	<u>29.96</u>	262.74	10.03	<u>32.24</u>	297.85	<u>153.51</u>	120.03

B. Simulation Platform and Configuration

Simulations are conducted in CityFlow [19]. Each intersection runs four phases (NTST, NLSL, ETWT, ELWL) with right turns permitted; safety timing is 3 s yellow + 2 s all-red. Two strategies are evaluated: *Fixed* (30 s per phase) and *Dynamic* (adaptive up to 40 s). Each scenario runs for 1 hour.

1) *Performance Metrics*: **ATT** (s): average travel time; **AQL** (m): average queue length (network-wide); **AWT** (s): average waiting time at intersections.

2) *Compared Models*: Baselines include transportation methods (Random, FixedTime [20], MaxPressure [21]), deep RL families (PressLight [22], MPLight [23], CoLight [24], Efficient-CoLight [25], Advanced-CoLight [26], AttendLight [27]), Large-scale AI Models (Llama-2-13B, ChatGPT-3.5/4/4o-mini, ERNIE-4.0-8K/ERNIE-Lite/ERNIE-Speed-8K, Hunyuan-Lite, Qwen-Long, Spark-Lite, LLM-Light [5], Traffic-R1 [28]), the dynamic timing method DynamicLight [13], and the proposed HeraldLight.

3) *Resource Usage*: Dual-socket Intel Xeon Platinum 8457C with a single NVIDIA L20 (48 GiB, CUDA 12.4). Under synchronous batching (avg. batch size ≈ 2.85 ; avg. input/output 960/356 tokens), we observe an average batch-level inference time of 9.09 s, a token throughput of 413.35 tokens/s,

and GPU utilization of 95.96% with peak memory usage of 44,280.6 MiB.

C. Comparison Between HeraldLight and Other TSC Methods (RQ1)

As shown Table I, across five CityFlow scenarios Jinan (1-3), Hangzhou (1-2), HeraldLight, relative to a state-of-the-art baseline (DynamicLight), ranks first in ATT and remains top-2 in AQL across five scenarios. In Jinan 1, AQL is lower than DynamicLight by **11.72%**, whereas DynamicLight attains lower AWT in multiple scenarios. Dynamic timing methods collectively outperform fixed timing baselines; fixed timing RL variants (e.g., Advanced-CoLight) are competitive in isolated cases but underperform on aggregate.

Variation in performance stems from distinct mechanisms: RL improves as optimization reduces policy error and sharpens value estimates, aligning states and actions. Large-scale AI models (e.g., ChatGPT, DeepSeek) match Advanced-Colight without task-specific training due to strong reasoning priors and instruction following, while LLMLight remains competitive under fixed timing but is constrained by capacity and schedule rigidity. Dynamic timing resolutions DynamicLight uses 5-s multiples, whereas HeraldLight provides 1-s control,

TABLE II
ABLATION ON HANGZHOU 2 WITH LLAMA-3.1-8B (LOWER IS BETTER).
— INDICATES THE TASK WAS NOT COMPLETED.

No.	Configuration	ATT	AQL	AWT
1	Llama-3.1-8B (base)	—	—	—
2	No. 1 + Herald Module	—	—	—
3	No. 2 + Imitation Fine-tuning	303.581	154.084	125.120
4	No. 3 + Dual LLMs Architecture	297.852	153.507	120.029

enabling finer timing that reduces delay and increases throughput.

D. Ablation Study

An ablation on Hangzhou 2 (high demand) with Llama-3.1-8B quantifies module contributions (Table II). The base model and the base augmented with Herald failed to complete the task, indicating that traffic-aware forecasting alone is insufficient under heavy load. Imitation fine-tuning stabilized control (ATT 303.581). Incorporating the Dual LLMs architecture further reduced ATT to 297.852, supporting complementary gains from forecasting and critic-guided arbitration.

E. Generalization Comparison (RQ2)

1) *Scalability*: In the multi-intersection New York network, HeraldLight scales better than competing methods shown in Fig. 5. Across the Jinan, Hangzhou, and New York scenarios, HeraldLight achieves a 20.03% reduction in average travel time relative to SOTA method. Herald-guided queue forecasting and LLM-based reasoning enable distributed agents to issue context-aware phase releases under dynamic timing, accelerating discharge on high-pressure approaches and reducing delay and secondary queues. The advantage is amplified on the 196 intersections New York network approximately twelvefold larger than Jinan and Hangzhou.

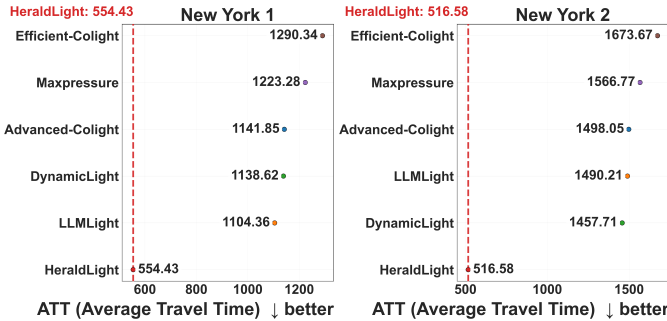


Fig. 5. The performance in the large-scale road network

2) *Transferability*: As shown in Fig. 6, The original model is trained and evaluated in the target scenario, whereas the transferred model uses the same method trained on other scenarios and is evaluated in the target scenario.

Across Jinan (1-3) and Hangzhou (1-2), in-scenario training generally yields better performance than cross-scenario transfer. HeraldLight exhibits the strongest transferability: ATT differences between in-scenario and transferred models are

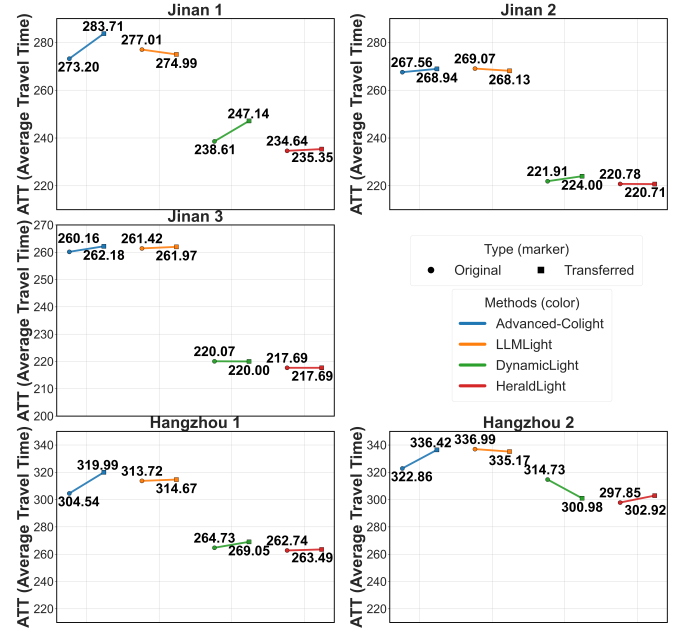


Fig. 6. Transferability compared in Jinan and Hangzhou

typically below one second; in Jinan 3 the difference is 0.006 s (217.68729 and 217.69368). LLMLight ranks second, with most gaps in the range of 2–3 s and stable variability. DynamicLight and Advanced-CoLight transfer less effectively than the LLM-based agents; DynamicLight ranks third and approaches HeraldLight in Jinan 3. The marked separation between LLMLight and DynamicLight is consistent with their fixed timing and dynamic-timing designs.

F. Reducing Hallucination (RQ3)

Hallucination is defined as context-incoherent, repetitive, or constraint-violating outputs at the decision step. The metric is computed as the fraction of interactions that exhibit any such symptoms. Hangzhou 2 is the most challenging setting in training due to higher demand and more intersections.

TABLE III
HALLUCINATION COMPARISON ON HANGZHOU 2.

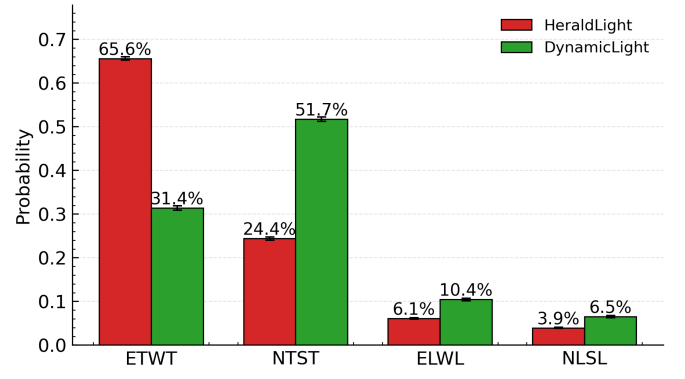
Method	Interactions	Hallucinations	Rate (%)
LLM-Agent	6048	558	9.23
HeraldLight	5525	9	0.163

Results indicate a substantial reduction with HeraldLight: hallucination cases drop from 558 to 9 (98.4% relative decrease) and the rate falls from 9.23% to 0.163%. The interaction count also declines (6048 → 5525, ~8.6%), suggesting more stable decision making. These outcomes attribute the improvement to the Dual LLMs architecture and traffic-aware guidance, which increase interpretability and enhance reliability under heavy traffic.

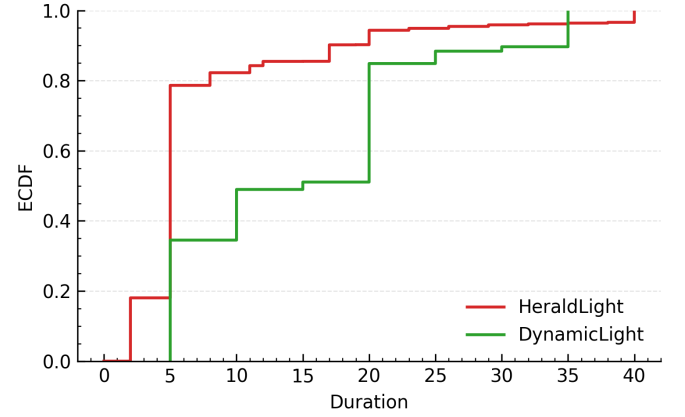
TABLE IV

EW (EXTREME WEATHER) VS. BASE PER SCENE/METRIC. EACH CELL SHOWS BASE $\rightarrow$ EW AND THE PERCENTAGE CHANGE VS. ITS OWN BASE. BOLD INDICATES THE BEST RESULTS (LOWER IS BETTER) WITHIN EACH ROW.

Scene	Metric	HeraldLight	Advanced-CoLight	DynamicLight
JN-1	ATT	234.64 $\rightarrow$ 361.18 (Δ +53.93%)	273.20 $\rightarrow$ 390.18 (Δ +42.82%)	238.61 $\rightarrow$ 372.36 (Δ +56.05%)
JN-1	AQL	73.69 $\rightarrow$ 161.51 (Δ +119.17%)	158.65 $\rightarrow$ 218.21 (Δ +37.54%)	83.48 $\rightarrow$ 188.37 (Δ +125.65%)
JN-1	AWT	40.55 $\rightarrow$ 84.05 (Δ +107.27%)	47.26 $\rightarrow$ 66.64 (Δ +41.01%)	27.35 $\rightarrow$ 76.68 (Δ +180.37%)
JN-2	ATT	220.78 $\rightarrow$ 310.49 (Δ +40.63%)	267.56 $\rightarrow$ 353.57 (Δ +32.15%)	221.91 $\rightarrow$ 317.60 (Δ +43.12%)
JN-2	AQL	28.58 $\rightarrow$ 45.74 (Δ +60.04%)	101.26 $\rightarrow$ 106.97 (Δ +5.64%)	32.46 $\rightarrow$ 57.28 (Δ +76.46%)
JN-2	AWT	27.63 $\rightarrow$ 33.32 (Δ +20.59%)	40.81 $\rightarrow$ 51.54 (Δ +26.29%)	16.55 $\rightarrow$ 27.82 (Δ +68.10%)
JN-3	ATT	217.69 $\rightarrow$ 317.98 (Δ +46.07%)	260.16 $\rightarrow$ 351.53 (Δ +35.12%)	220.07 $\rightarrow$ 325.67 (Δ +47.98%)
JN-3	AQL	44.15 $\rightarrow$ 84.39 (Δ +91.14%)	127.04 $\rightarrow$ 146.94 (Δ +15.66%)	50.23 $\rightarrow$ 101.35 (Δ +101.77%)
JN-3	AWT	29.96 $\rightarrow$ 48.99 (Δ +63.52%)	42.84 $\rightarrow$ 54.83 (Δ +27.99%)	20.95 $\rightarrow$ 40.60 (Δ +93.79%)
HZ-1	ATT	262.74 $\rightarrow$ 362.75 (Δ +38.06%)	304.54 $\rightarrow$ 399.88 (Δ +31.31%)	264.73 $\rightarrow$ 372.42 (Δ +40.68%)
HZ-1	AQL	10.03 $\rightarrow$ 15.37 (Δ +53.24%)	52.97 $\rightarrow$ 56.92 (Δ +7.46%)	13.02 $\rightarrow$ 26.72 (Δ +105.22%)
HZ-1	AWT	32.24 $\rightarrow$ 33.65 (Δ +4.37%)	40.67 $\rightarrow$ 57.61 (Δ +41.65%)	12.33 $\rightarrow$ 26.98 (Δ +118.82%)
HZ-2	ATT	297.85 $\rightarrow$ 385.30 (Δ +29.36%)	322.86 $\rightarrow$ 419.83 (Δ +30.03%)	314.73 $\rightarrow$ 397.51 (Δ +26.30%)
HZ-2	AQL	153.51 $\rightarrow$ 143.17 (Δ -6.74%)	173.38 $\rightarrow$ 194.29 (Δ +12.06%)	146.06 $\rightarrow$ 145.53 (Δ -0.36%)
HZ-2	AWT	120.03 $\rightarrow$ 136.81 (Δ +13.98%)	69.93 $\rightarrow$ 80.72 (Δ +15.43%)	53.99 $\rightarrow$ 59.37 (Δ +9.96%)



(a) Phase distribution



(b) Duration ECDF

Fig. 7. HeraldLight vs DynamicLight action comparison (a) Phase distribution. (b) Duration ECDF (Empirical Cumulative Distribution Function).

G. Extreme Weather Condition

Based on the performance presented in Table I, the top three methods, HeraldLight, DynamicLight, and Advanced-Colight, are selected according to performance under normal weather conditions. Stability under extreme weather conditions was then assessed by adjusting vehicle parameters to simulate the effects of adverse weather. Key modifications included a 50% reduction in maximum positive acceleration, a 30% reduction in negative acceleration, and a 30% reduction in maximum speed.

Under extreme conditions, HeraldLight shows average increases of 41.61 s in ATT, 63.39% in AQL, and 41.946% in AWT. DynamicLight experienced greater degradation (42.83% ATT, 81.75% AQL, 94.23% AWT), while Advanced-Colight showed smaller increases (34.28% ATT, 15.67% AQL, 30.47% AWT). HeraldLight's transparent decision-making process allows it to quickly adjust parameters like vehicle speed and queue maps, making it more adaptable to extreme weather.

H. Actions Analysis of HeraldLight and DynamicLight

Action selection in New York 1: In the New York grid (7×28 , 196 intersections), we compare HeraldLight and DynamicLight. From Fig. 7(a), both methods emphasize ETWT and NTST phases. HeraldLight favors ETWT phase releases,

aligning with heavier inflows along the 28 intersections east-west spine; high-volume arterials benefit from faster phase switching to sustain progression. Its finer control granularity enables more precise actions during peaks. DynamicLight biases NTST phase releases, which can aid longer cross-network trips, but during peaks this intensifies queues on east-west approaches, increasing average travel time; HeraldLight is therefore more efficient under congestion.

Duration distributions. From the ECDF in Fig. 7(b), HeraldLight's curve is smoother, indicating finer timing resolution, whereas DynamicLight exhibits 5 s step increments (coarser granularity). HeraldLight concentrates green durations below 20 s, improving responsiveness to rapid peak fluctuations. DynamicLight allocates longer (mostly ≤ 35 s), which elevates the ATT of the other three phases during peaks, consistent with its weaker performance in New York 1.

VI. CONCLUSION

This study presents HeraldLight, a traffic signal control framework that combines a scene-aware Herald Module with large language model agents to realize dynamic, second-level timing. Findings indicate that: (i) fixed time strategies lack the precision required for fine grained control in complex traffic; (ii) coupling LLM reasoning with the Herald Module enables

precise timing and improves control effectiveness; and (iii) LLMs are susceptible to hallucination, which is effectively mitigated by the proposed Dual LLMs architecture, enhancing stability and reliability. Future work will emphasize online self-evolution within the LLMs for action reasoning and output generation, and target deployment in real-world systems.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 62176024).

APPENDIX A INTERSECTION MODELING

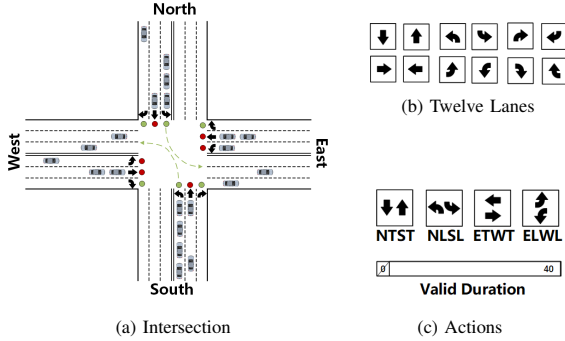


Fig. A.1. Intersection Modeling

REFERENCES

- [1] J. Guo, S. Ghanadhashi, S. Wang, and F. Golpayegani, "Urban traffic signal control at the edge: An ontology-enhanced deep reinforcement learning approach," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 6027–6033.
- [2] B. Xu, Y. Wang, Z. Wang, H. Jia, and Z. Lu, "Hierarchically and cooperatively learning traffic signal control," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 1, 2021, pp. 669–677.
- [3] Y. Gu, K. Zhang, Q. Liu, W. Gao, L. Li, and J. Zhou, " π -light: Programmatic interpretable reinforcement learning for resource-limited traffic signal control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 107–21 115.
- [4] E. Liang, Z. Su, C. Fang, and R. Zhong, "OAM: An option-action reinforcement learning framework for universal multi-intersection control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4550–4558.
- [5] S. Lai, Z. Xu, W. Zhang, H. Liu, and H. Xiong, "Large language models as traffic signal control agents: Capacity and opportunity," *arXiv preprint arXiv:2312.16044*, 2023.
- [6] J. Fan, H. Chu, L. Liu, and H. Ma, "Llmair: Adaptive reprogramming large language model for air quality prediction," in *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, 2024, pp. 423–430.
- [7] X. Chen, J. Cumin, F. Ramparany, and D. Vaufraydaz, "Towards llm-powered ambient sensor based multi-person human activity recognition," in *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, 2024, pp. 609–616.
- [8] L. Da, M. Gao, H. Mei, and H. Wei, "Prompt to transfer: Sim-to-real transfer for traffic signal control with prompt learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 82–90.
- [9] L. Da, K. Liou, T. Chen, X. Zhou, X. Luo, Y. Yang, and H. Wei, "Open-ti: Open traffic intelligence with augmented language model," *International Journal of Machine Learning and Cybernetics*, pp. 1–26, 2024.
- [10] A. Pang, M. Wang, M.-O. Pun, C. S. Chen, and X. Xiong, "illm-tsc: Integration reinforcement learning and large language model for traffic signal control policy improvement," *arXiv preprint arXiv:2407.06025*, 2024.
- [11] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, Jan. 2025. [Online]. Available: <https://doi.org/10.1145/3703155>
- [12] M. Wang, X. Xiong, Y. Kan, C. Xu, and M.-O. Pun, "Unitsa: A universal reinforcement learning framework for v2x traffic signal control," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 10, pp. 14 354–14 369, 2024.
- [13] L. Zhang, Y. Zhang, S. Xie, J. Deng, and C. Li, "DynamicLight: Two-stage dynamic traffic signal timing," 2024. [Online]. Available: <https://arxiv.org/abs/2211.01025>
- [14] H. Kim, Z. Jin, H. Tak, H. Yu, and H. Yeo, "Prioritized phase split optimization for coordinated traffic signal control in urban network using deep reinforcement learning," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 833–838.
- [15] W. Dong, W. Liu, R. Xi, M. Hou, and S. Fan, "Mletune: Streamlining database knob tuning via multi-llms experts guided deep reinforcement learning," in *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, 2024, pp. 226–235.
- [16] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.
- [17] P. Wang, L. Li, L. Chen, F. Song, B. Lin, Y. Cao, T. Liu, and Z. Sui, "Making large language models better reasoners with alignment," 2024.
- [18] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," *arXiv preprint arXiv:1904.08117*, 2019.
- [19] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. J. Li, "CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario," *The World Wide Web Conference*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:153312553>
- [20] P. Koonce, "Traffic signal timing manual," Tech Report FHWA-HOP-08-024, 2008.
- [21] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
- [22] H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "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*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1290–1298.
- [23] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3414–3421, Apr. 2020.
- [24] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "Colight: Learning network-level cooperation for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1913–1922.
- [25] Q. Wu, L. Zhang, J. Shen, L. Lu, B. Du, and J. Wu, "Efficient pressure: Improving efficiency for signalized intersections," *ArXiv*, vol. abs/2112.02336, 2021.
- [26] L. Zhang, Q. Wu, J. Shen, L. Lü, B. Du, and J. Wu, "Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 645–26 654.
- [27] A. Oroojlooy, M. Nazari, D. Hajinezhad, and J. Silva, "Attendlight: Universal attention-based reinforcement learning model for traffic signal control," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 4079–4090.
- [28] X. Zou, Y. Yang, Z. Chen, X. Hao, Y. Chen, C. Huang, and Y. Liang, "Traffic-r1: Reinforced llms bring human-like reasoning to traffic signal control systems," 2025. [Online]. Available: <https://arxiv.org/abs/2508.02344>