

# LLMLight: Large Language Models as Traffic Signal Control Agents

Siqi Lai

The Hong Kong University of Science  
and Technology (Guangzhou)  
slai125@connect.hkust-gz.edu.cn

Zhao Xu

The Hong Kong University of Science  
and Technology (Guangzhou)  
zxu674@connect.hkust-gz.edu.cn

Weijia Zhang

The Hong Kong University of Science  
and Technology (Guangzhou)  
wzhang411@connect.hkust-  
gz.edu.cn

Hao Liu\*

The Hong Kong University of Science  
and Technology (Guangzhou)  
liuh@ust.hk

Hui Xiong

The Hong Kong University of Science  
and Technology (Guangzhou)  
xionghui@ust.hk

## ABSTRACT

Traffic Signal Control (TSC) is a crucial component in urban traffic management, aiming to optimize road network efficiency and reduce congestion. Traditional TSC methods, primarily based on transportation engineering and reinforcement learning (RL), often struggle with generalization abilities across varied traffic scenarios and lack interpretability. This paper presents LLMLight, a novel framework employing Large Language Models (LLMs) as decision-making agents for TSC. Specifically, the framework begins by instructing the LLM with a knowledgeable prompt detailing real-time traffic conditions. Leveraging the advanced generalization capabilities of LLMs, LLMLight engages a reasoning and decision-making process akin to human intuition for effective traffic control. Moreover, we build LightGPT<sup>1</sup>, a specialized backbone LLM tailored for TSC tasks. By learning nuanced traffic patterns and control strategies, LightGPT enhances the LLMLight framework cost-effectively. Extensive experiments conducted on ten real-world and synthetic datasets, along with evaluations by fifteen human experts, demonstrate the exceptional effectiveness, generalization ability, and interpretability of LLMLight with LightGPT, outperforming nine baseline methods and ten advanced LLMs. Our project is available at <https://github.com/usail-hkust/LLMTSCS>.

## CCS CONCEPTS

- Computing methodologies → Planning for deterministic actions.

## KEYWORDS

traffic signal control, large language model, traffic control agent, intelligent transportation

<sup>1</sup>The model weights are available at <https://huggingface.co/lightgpt>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/18/06  
<https://doi.org/XXXXXXX.XXXXXXX>

## ACM Reference Format:

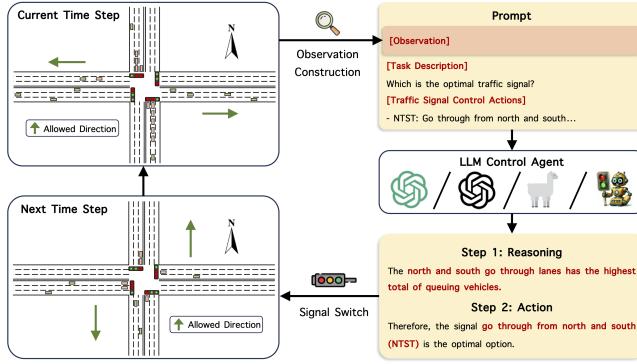
Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu\*, and Hui Xiong. 2018. LLMLight: Large Language Models as Traffic Signal Control Agents. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 17 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 INTRODUCTION

Traffic congestion has emerged as a critical issue impacting both human society and the environment. As urban migration continues accelerating, city populations are expanding rapidly, further intensifying this challenge. This trend has necessitated a focused effort on optimizing Traffic Signal Control (TSC), a pivotal research area in intelligent transportation [51, 62]. Efficient TSC not only promises substantial economic and environmental benefits but also enhances societal well-being. However, effectively managing traffic signals in complex and dynamic urban road networks remains daunting.

Existing work on TSC primarily falls into two categories: transportation [11, 22, 25, 29, 43] and Reinforcement Learning (RL)-based approaches [6, 8, 35, 47, 48, 51, 52, 59]. Transportation methods primarily focus on crafting efficient heuristic algorithms, dynamically adapting traffic signal configurations based on lane-level traffic conditions. However, these methods heavily depend on manual design and often exhibit limited effectiveness in complex scenarios. The emergence of deep neural networks (DNNs) [31] led to the introduction of deep RL-based techniques to address this challenge [1, 15]. These approaches have exhibited remarkable performance across various traffic scenarios. Nevertheless, RL-based methods also present several drawbacks. Primarily, they may struggle with limited generalization ability, particularly when transferring to larger road networks or under highly uncommon scenarios (e.g., extreme high-traffic situations), as their training data only covers limited traffic situations. Additionally, RL-based methods lack interpretability due to the black-box nature of DNNs, which makes it hard to explain the rationale behind control actions under specific traffic conditions. This lack of transparency significantly hampers their feasibility for real-world deployment.

The recent emergence of Large Language Models (LLMs) has revolutionized several domains with their exceptional zero-shot learning and generalization capabilities, emulating human-like reasoning to solve complex tasks. For instance, AutoGPT [16] proposes



**Figure 1: The workflow of LLMLight.**

to break tasks into multiple sub-goals and iterate until the main task is completed. In the transportation domain, GPT-Driver [30] proposes to instruct GPT-3.5 to tackle motion planning tasks in autonomous driving. For traffic control, PromptGAT [13] uses LLMs to generate human knowledge to help the DNN model understand long-tail cases (e.g., extreme weather) in TSC tasks, aiming to bridge the gap between real-world and simulations. Despite these advancements, existing research heavily relies on generalist LLMs [45] (e.g., ChatGPT) and primarily uses them as auxiliary tools [13, 37]. However, generalist LLMs often lack specialized domain knowledge in TSC, potentially limiting their effectiveness in professional applications. The direct application of LLMs as TSC agents for human-like decision-making remains largely unexplored. Addressing these issues involves overcoming two major challenges.

First, LLMs are typically pre-trained on large-scale natural language corpora and rarely incorporate non-textual traffic data, such as sensor readings and GPS trajectories [10, 42]. Despite their generalization capability across various tasks and domains, an inherent gap exists between real-time traffic data and linguistic understanding. The first challenge lies in enabling LLMs to comprehend real-time traffic dynamics and effectively interact with the traffic environment, which is critical for effective LLM-based TSC.

Second, selecting and developing an effective LLM for TSC poses another significant challenge. Generalist LLMs often lack specific domain knowledge and are prone to hallucination problems in professional fields [21, 23]. Although state-of-the-art LLMs such as GPT-4 demonstrate promising generalization abilities, their closed-source nature and substantial usage costs pose barriers to their optimization for real-time TSC tasks. Consequently, building a specialized LLM tailored for TSC tasks is crucial to deliver more effective and human-aligned control policies.

To this end, we introduce LLMLight, a traffic signal control agent framework based on LLMs, as depicted in Figure 1. Specifically, we consider TSC as a partially observable Markov Game, where each agent, armed with an LLM, manages the traffic light at an intersection. At each signal-switching time step, the agent collects traffic conditions of the target intersection and verbalizes them into human-readable text as real-time observation. Additionally, we incorporate task descriptions enriched with commonsense knowledge about a control strategy to aid the LLM's understanding of traffic management tasks. The combined real-time observation, task description, and control action space form a knowledgeable prompt

that guides the LLM's decision-making. Then, the agent leverages Chain-of-Thought (CoT) reasoning to determine the optimal traffic signal configuration for the subsequent time step.

Furthermore, we construct a specialized LLM, LightGPT, to enhance the LLMLight framework. On the one hand, we propose imitation fine-tuning, enabling the specialized LLM to exploit high-quality control actions and underlying rationales derived from GPT-4. On the other hand, we introduce a policy refinement process that utilizes a well-trained critic model to evaluate and improve the agent's actions. The optimized LightGPT produces more effective control policies and showcases remarkable generalization ability across diverse traffic scenarios in a more cost-effective manner than the leading LLMs, including GPT-4.

We summarize the major contributions of this paper as follows: (1) We pioneer the application of advanced LLMs as control agents, enabling human-like decision-making for interpretable traffic signal control. To our knowledge, this is the first exploration of directly using LLMs as decision-making agents in TSC. (2) We introduce LLMLight, a tailored framework that engages various LLMs in effectively regulating traffic signals via knowledgeable instructions. (3) We construct LightGPT, a specialized LLM backbone optimized for TSC tasks. This significantly enhances the framework's capability to derive effective control policies cost-effectively. (4) Extensive experiments on ten traffic datasets, along with evaluations by fifteen human experts, demonstrate superior effectiveness, generalization capabilities, and interpretability of LLMLight and LightGPT compared to nine baseline methods and ten advanced generalist LLMs. Additionally, these experiments reveal key insights into the potential of integrating LLMs into intelligent transportation systems.

## 2 PRELIMINARY

In this section, we first introduce the key concepts in traffic signal control tasks.

**Definition 2.1. Road network:** The road network is a directed graph consisting of intersection joints  $I$  and lanes  $L$ . Lanes are categorized into go-through ( $L_{go}$ ), left-turn ( $L_{left}$ ), and right-turn ( $L_{right}$ ) lanes, determined by the movement of vehicles on the lane. Each lane is divided into multiple segments  $S = \{s_1, \dots, s_n\}$ .

**Definition 2.2. Traffic signal control action:** The traffic signal control action corresponds to a specific signal phase, denoted as  $a = \text{set}(L_{allow})$ , where  $L_{allow}$  is a group of allowed lanes without conflicting movements.  $\text{set}(L_{allow})$  indicates the green light for activating the signal phase. At each signal-switching time step, the agent controlling the intersection chooses an action from  $A = \{a_1, \dots, a_m\}$ . The active signal phase allows vehicles in a specific lane group to pass, while other phases display the red light.

Please refer to Figure 8 (See Appendix A.1) for more details of the most used setting of the intersection, lanes, and signal phases.

## 3 LLM FOR TRAFFIC SIGNAL CONTROL

This section first introduces the task formulation of traffic signal control in the LLM-empowered context. Then, we delve into the workflow of our proposed LLMLight, elucidating the methodology for prompting LLMs to perceive, analyze, and regulate traffic at the target intersection directly. Finally, we elaborate on the training

procedures implemented for developing LightGPT, aiming to augment the backbone LLM’s domain expertise in traffic management, thereby generating proficient control policies adaptable to diverse traffic scenarios.

### 3.1 Task Formulation

We define traffic signal control as a partially observable Markov Game [47, 48]. The LLM-based agent manages the traffic light of an intersection with policy  $\pi_\theta$ . Based on the observation  $o$  of the real-time traffic condition at the intersection, traffic scenario description  $d_{\text{scene}}$ , task descriptions  $d_{\text{task}}$ , commonsense knowledge  $d_{\text{know}}$ , and control action space  $A$ , the LLM outputs the reasoning trajectory  $Y$  with the control action  $a$  that regulates the traffic signal of the intersection, with the long-term goals of maximizing the transportation efficiency within the road network:

$$(Y, a) = \pi_\theta(\text{Prompt}(o, d_{\text{scene}}, d_{\text{task}}, d_{\text{know}}, A)). \quad (1)$$

### 3.2 Workflow of LLMLight

This subsection details the workflow of our proposed LLMLight, including (1) *observation collection*: it collects the traffic condition of the intersection; (2) *commonsense augmented prompt construction*: it composes a commonsense knowledge enhanced prompt, instructing the LLM to devise the optimal traffic signal configuration; (3) *reasoning and action execution*: the LLM engages a decision-making process by employing the constructed prompt, and subsequently executes its selected control action.

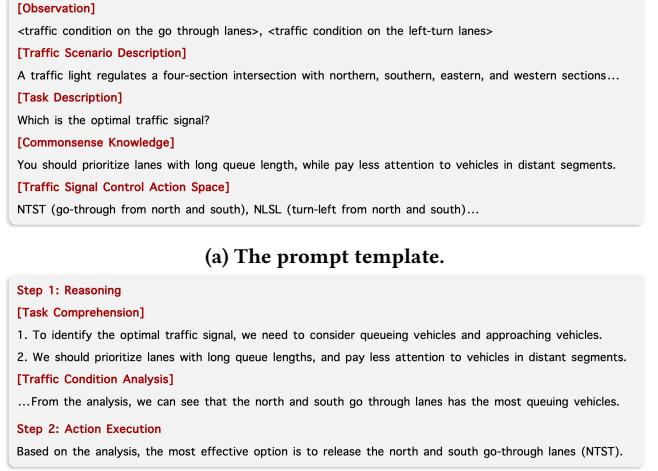
**3.2.1 Observation Feature Construction.** We collect traffic condition features that can be easily obtained from the real-world traffic control environment:

- **Queuing vehicle count:** Vehicles with speeds slower than the threshold  $v_{\text{stop}}$  are considered as queuing vehicles. We count their number in lane  $l$  as  $n_q^l$ .
- **Approaching vehicle count:** Vehicles faster than  $v_{\text{stop}}$  are considered as approaching vehicles. We count the number of approaching vehicles in the segment  $s$  of lane  $l$  as  $n_s^l$ .

Subsequently, these features on every incoming lane (where the vehicle enters the intersection) are verbalized into human-readable text as the observation  $o$  of the LLM.

**3.2.2 Commonsense Augmented Prompt Construction.** In addition to observation, we provide the LLM with comprehensive information essential in addressing traffic signal control, including traffic scenario description  $d_{\text{scene}}$ , task descriptions  $d_{\text{task}}$ , and control action space  $A$ . This enables the LLM to thoroughly understand the task, facilitating reasonable control actions. Furthermore, we integrate commonsense knowledge  $d_{\text{know}}$  to mitigate generalist LLMs’ limitations in traffic management, a domain they are not explicitly trained for. Specifically, this knowledge dictates that the agent needs to prioritize lanes with extended queue lengths while paying less attention to vehicles distant from the intersection. Formally, we denote the composed prompt as  $X = \text{Prompt}(o, d_{\text{scene}}, d_{\text{task}}, d_{\text{know}}, A)$ . A glimpse of the prompt template is shown in Figure 2a, and a detailed example is provided in Table 11 (see Appendix A.12).

**3.2.3 Reasoning and Action Execution.** We instruct the LLM to perform zero-shot reasoning using the constructed prompt. Its



**(b) The reasoning and action execution process.**

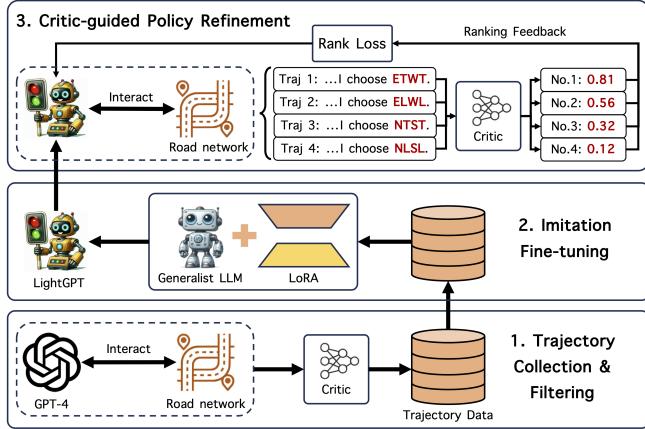
**Figure 2: The prompt template of LLMLight and decision-making process of the backbone LLM. In this scenario, the intersection connects four lane sections, including north, south, east, and west.**

decision-making unfolds in two key steps: reasoning and action execution. Initially, it embarks on comprehensive analysis, assimilating the given task, leveraging commonsense knowledge, and assessing the current traffic conditions across various lanes. Subsequently, it selects the signal for the lanes experiencing the highest congestion, thereby optimizing traffic flow and ensuring the smooth passage of vehicles. In this manner, LLMLight can not only make effective control actions but also distinguishes itself by providing detailed rationales for each decision, contributing to a more explainable and transparent traffic control system. Formally, we denote the reasoning and action execution as  $(Y, a) = \pi_\theta(X)$ . An example of the backbone LLM’s decision-making process is shown in Figure 2b. We conduct an interpretability analysis in Subsection 4.8.

### 3.3 Backbone LLM Training

Fine-tuning has become a common routine to align LLM with diverse downstream applications [5, 57] by augmenting the domain expertise in the specific realm. We propose a training procedure to optimize an LLM specifically tailored for traffic signal control, leading to the development of LightGPT. Figure 3 presents the training workflow. It primarily consists of three stages: (1) *reasoning trajectory collection and filtering*: it collects the CoT reasoning trajectory of GPT-4 for imitation fine-tuning, then we select the trajectories that most align with the long-term goal to ensure the data quality; (2) *imitation fine-tuning*: it utilizes the filtered reasoning trajectories and the corresponding actions taken by GPT-4 to train the backbone LLM; (3) *critic-guided policy refinement*: it further improves the decision-making process of the LLM by fine-tuning with the feedback of a well-trained critic model.

**3.3.1 Trajectory Collection and Filtering.** Following the prompt construction process detailed in Section 3.2.2, we first instruct GPT-4 to interact with simulated traffic environments for gathering



**Figure 3: The training procedure of LightGPT.**

reasoning trajectories for subsequent imitation fine-tuning. To ensure the quality of the collected data, we employ a filtering strategy that identifies reasoning trajectories aligning most closely with the long-term objective of traffic signal control, such as minimizing future queue length. This filter operation is achieved by aligning with a pre-trained action-value network (the architecture is derived from [59]), which is trained by optimizing Bellman equation within the simulation environment:

$$\mathcal{L}_{value} = \mathbb{E}[(R(o_t, a_t) + \gamma \max_{a_{t+1}} Q(o_{t+1}, a_{t+1}) - Q(o_t, a_t))^2], \quad (2)$$

where  $o_t$  and  $a_t$  are the observation and action chosen at the signal-switching time step  $t$ , and  $\gamma \in [0, 1]$  is the discount factor.  $R(o, a)$  is the reward function that provides feedback for executing action  $a$  under observation  $o$  (e.g., the negative of queue length).  $Q(o, a)$  is the action-value function estimating the cumulative future reward earned after executing  $a$ . Actions estimated to yield higher future rewards suggest greater alignment with the long-term goal.

Subsequently, the trained action-value function is utilized as a critic to evaluate the decision-making of GPT-4. We selectively retain only reasoning trajectories where the selected actions are associated with the highest estimated future rewards, as:

$$\{(X_t, Y_t, a_t) | a_t = \operatorname{argmax}_{a \in A} Q(o_t, a)\}_{t=0}^T, \quad (3)$$

where  $T$  is the simulation duration,  $X_t$  is prompt, and  $Y_t$  is the reasoning trajectory of GPT-4.

**3.3.2 Imitation Fine-tuning.** In the initial stage of backbone LLM training, we employ an imitation learning process, allowing the LLM to train on the control actions taken by GPT-4 and their underlying rationales. We treat prompt  $X$  as fine-tuning instructions and GPT-4's reasoning trajectory  $Y = [Y; a]$  containing its selected action  $a$  as the anticipated response. Negative log-likelihood (NLL) is adopted as the loss function:

$$\mathcal{L}_{IFT} = - \sum_w^{|Y|} \log P_{\pi_\theta}(y_w | X, Y_{<w}), \quad (4)$$

where  $P_{\pi_\theta}(y_w | X, Y_{<w})$  is the probability of being prompted with  $X$  to generate the token  $y_w$ . Due to the enormous GPU memory usage of full model training, we adopt parameter-efficient fine-tuning

(PEFT) techniques, specifically low-rank adaptation (LoRA) [20], in our training. Instead of adjusting all model weights, LoRA introduces low-rank matrices to model the weight updates, significantly reducing the number of parameters to be trained. This approach not only decreases computational and storage costs but also preserves the original model's performance.

**3.3.3 Critic-guided Policy Refinement.** To further enhance the effectiveness of the LLM's control policy, we propose a policy refinement procedure, which adjusts the LLM's reasoning trajectory for directing better actions. Similarly, we use the pre-trained action-value function elaborated in Section 3.3.1 as the critic model to evaluate the actions derived by the backbone LLM. Higher evaluation scores correspond to greater estimated future rewards. Subsequently, we implement an alignment fine-tuning algorithm to adjust the reasoning trajectory, ultimately guiding the LLM towards actions that yield higher scores. Although previous studies [4, 36] prove RL (e.g., PPO [39]) is a potent tool for alignment, this method proved challenging to converge and yielded unsatisfactory results, as validated by [46] and the experiment in Appendix A.9. Alternatively, we leverage a ranking loss [46, 56] function to achieve this goal.

Specifically, we have  $k$  different reasoning trajectories  $Y = \{Y_i\}_{i=1}^k$  sampled by policy  $\pi_\theta$  under instruction  $X$ . The critic gives scores for actions derived by each trajectory  $Y_i$  as  $q_i = Q(o, a_i)$ . Afterwards, the token-averaged log-likelihood value of each  $Y_i$  is derived to denote the corresponding probability for  $\pi_\theta$  to generate  $Y_i$ :

$$p_i = \frac{1}{|Y_i|} \sum_w^{|Y_i|} \log P_{\pi_\theta}(y_i, w | X, Y_i, <w). \quad (5)$$

Following [46], we adopt ranking-feedback loss with a boundary-constrained term (RBC) for generating reasoning trajectories that result in actions with higher scores:

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

where  $p_{j*} = \min_{q_k > q_j} p_k$  is the probability of the least favorable reasoning trajectory rated higher than  $Y_j$ ,  $\beta$  is a hyper-parameter,  $e^{(p_j - p_i)}$  is the alignment term to elevate the probability of generating trajectories with actions rated higher scores, and  $e^{(2p_{j*} - 2\beta - p_i - p_j)}$  is the constraint term for prohibiting performance degradation.

It is crucial to be aware that while this refinement process is guided by a pre-trained critic model developed from RL, it does not imply that the backbone LLM learns exclusively from it. The alignment process is trained on reasoning trajectories sampled by the LLM's policy, aiming to help the model identify trajectories with reasonable decisions that align closely with long-term goals. However, it is important to note that we do not provide any ground-truth labels in the training, thus the LLM is not bound to follow the critic's predicted "best" action directly. Experiments in Subsection 4.7 demonstrate that RL models show performance degradation in varied traffic scenarios. In contrast, LLMLight equipped with LightGPT remains robust based on our proposed training procedure.

## 4 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate our proposed LLMLight by answering the following research questions:

**Table 1: Statistics of datasets.**

Flow dataset	# of intersections	# of vehicles	Arrival rate (vehicles/5min)			
			Mean	Std	Max	Min
Jinan 1	12	6295	523.67	98.52	671	255
Jinan 2		4365	362.83	74.81	493	236
Jinan 3		5494	456.92	46.38	544	362
Jinan Extreme		24000	1999.08	44.14	2097	1932
Jinan 24-hour		58200	201.09	160.88	569	2
Hangzhou 1	16	2983	247.67	40.44	332	211
Hangzhou 2		6984	581.08	318.43	1145	202
Hangzhou Extreme		24000	1999.08	40.66	2073	1934
New York 1	196	11058	849.69	174.06	964	382
New York 2		16337	1255.77	264.85	1440	475

- **RQ1:** How is the **effectiveness** of LLMLight compared with traditional transportation and RL methods?
- **RQ2:** How is the **generalization** ability of LLMLight across different traffic scenarios and vehicle volumes?
- **RQ3:** How is the **interpretability** of LLMLight in providing rationales behind the selected control actions?

## 4.1 Experimental Settings

**4.1.1 Datasets.** Our experiments use seven real-world [49] and three synthetic traffic flow datasets. Statistics for these datasets are detailed in Table 1. Visualizations of road networks are presented in Figure 18 (see Appendix A.11).

- **Jinan:** This dataset is from Dongfeng sub-district, Jinan, China, featuring 12 intersections. It includes three traffic flow datasets recorded over three different periods. The area covered is 400 meters east-west and 800 meters north-south.
- **Hangzhou:** Collected from Gudang sub-district, Hangzhou, China, this dataset features 16 intersections and includes two traffic flow datasets from different periods. The area spans 800 meters east-west and 600 meters north-south.
- **New York:** Collected in the Upper East Side of Manhattan, New York, USA, using taxi trip data, this dataset covers 196 intersections with lanes spanning 300 meters. It includes two large-scale traffic flow datasets from different periods to evaluate the scalability of different methods.
- **Extreme High-traffic:** Synthesized based on real-world traffic in Jinan and Hangzhou, these datasets feature approximately four times more vehicles per five minutes than the original datasets to simulate high-traffic scenarios.
- **24-hour Cycle:** Modeled on Jinan's actual traffic flows, this dataset replicates realistic daily traffic fluctuations over a 24-hour period, including rush hours [2] from 6:00 am to 10:00 am and 3:00 pm to 8:00 pm.

**4.1.2 Data Anonymization.** The datasets we used do not include any identifiable driver information, such as names or contact details. These datasets are open-sourced and publicly available at [58].

**4.1.3 Environment Settings.** We conduct experiments on CityFlow, an open-source simulator widely used for benchmarking traffic signal control policies. CityFlow supports large-scale, city-wide traffic simulations and features an interactive render for monitoring. It

provides extensive APIs to access traffic state features and execute control actions. In our experiments, each intersection connects four lane sections: East, West, North, and South. We utilize four signal phases as corresponding control actions: ETWT (go-through from east and west), ELWL (left-turn from east and west), NTST (go-through from north and south), and NLSL (left-turn from north and south). Right-turn movements are always allowed in the simulation. The green signal phase duration is set to thirty seconds. Following existing studies [48, 59], a three-second yellow signal and a two-second all-red time follow each green signal to prepare the transition. The simulation spans twenty-four hours for the 24-hour cycle dataset, while others span one hour.

## 4.2 Metrics

Following previous studies [59], we leverage *average travel time* (ATT), *average queue length* (AQL), and the *average waiting time* (AWT) of vehicles to evaluate the performance of different policies made by traffic signal control agents.

- **Average traveling time (ATT)** quantifies the average duration of all the vehicles traveling from their origins to their respective destinations.
- **Average queue length (AQL)** is defined as the average number of queuing vehicles waiting in the road network.
- **Average waiting time (AWT)** quantifies the average queuing time of vehicles at every intersection in the road network.

## 4.3 Compared Models

For transportation methods, we adopt Random, FixedTime, and Maxpressure as baselines. For RL methods, we compare MPLight, AttendLight, PressLight, CoLight, Efficient-CoLight, and Advanced-CoLight with our proposed method. For LLMs, we utilize GPT-4, ChatGPT-3.5, Qwen, Llama2, and Llama3 as backbone models for LLMLight. For open-source LLMs, such as Qwen and Llama, we evaluate their performance both with and without fine-tuning. Due to the high training and deployment costs of models larger than 70B parameters (e.g., Llama2-70B or Qwen2-72B) in real-world traffic scenarios, we only test and report their performances without fine-tuning. Please refer to Appendix A.2 for more details about the abovementioned models.

## 4.4 Model Settings

All RL methods are trained with uniform hyperparameters, encompassing a learning rate of  $1 \times 10^{-3}$ , a replay buffer size of 12000, a sample size of 3000, and a hidden size of 20. The top\_p value is standardized at 1.0 for all LLMs. To stabilize the LLMs' output, a temperature setting of zero is implemented for ChatGPT-3.5 and GPT-4, while open-sourced LLMs are set to 0.1. The settings of LoRA, including a rank of 8 and a scaling factor  $\alpha$  of 16, were set across all training stages. In the imitation fine-tuning, the batch and mini-batch sizes are set to 128 and 16, respectively. We sample four reasoning trajectories under each instruction for critic-guided policy refinement. The fine-tuning learning rate is set to  $3 \times 10^{-4}$ .

**Table 2: Overall performances of LLMLight and previous traditional methods on Jinan and Hangzhou datasets. The best, second-best, and third-best results are highlighted through boldface, double underline, and underline, respectively.**

Models	Jinan												Hangzhou											
	1			2			3			1			2			1			2			1		
	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT	ATT	AQL	AWT
<b>Transportation Methods</b>																								
Random	597.62	687.35	99.46	555.23	428.38	100.40	552.74	529.63	99.33	621.14	295.81	96.06	504.28	432.92	92.61									
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	486.72	425.15	72.80									
Maxpressure	281.58	170.71	<u>44.53</u>	273.20	106.58	<b>38.25</b>	265.75	133.90	<u>40.20</u>	325.33	68.99	49.60	347.74	215.53	70.58									
<b>RL Methods</b>																								
MPLight	307.82	215.93	97.88	304.51	142.25	90.91	291.79	171.70	89.93	345.60	84.70	81.97	358.56	237.17	100.16									
AttendLight	291.29	186.25	61.73	280.94	115.52	52.46	273.02	144.05	55.93	322.94	66.96	55.19	358.81	239.05	72.88									
PressLight	291.57	185.46	50.53	281.46	115.99	47.27	275.85	148.18	54.81	364.13	98.67	90.33	417.01	349.25	150.46									
CoLight	279.60	168.53	58.87	274.77	108.28	54.14	266.39	135.08	53.33	322.85	66.94	61.82	342.90	212.09	99.74									
Efficient-CoLight	277.11	163.60	<u>43.41</u>	<u>269.24</u>	<u>102.98</u>	<u>39.74</u>	262.25	129.72	<b>39.99</b>	311.96	<u>58.20</u>	<b>36.83</b>	<u>333.27</u>	<u>189.65</u>	<b>61.70</b>									
Advanced-CoLight	<u>274.67</u>	<u>160.85</u>	<u>49.30</u>	<u>268.25</u>	<u>102.12</u>	41.11	260.66	<u>127.83</u>	43.54	<b>304.47</b>	<b>52.94</b>	<u>41.75</u>	<b>329.16</b>	<b>186.34</b>	76.59									
<b>LLMLight (with Generalist LLMs)</b>																								
Qwen2-72B	291.95	185.20	58.65	277.78	112.13	53.12	274.33	144.95	54.94	321.91	65.75	66.52	342.37	206.07	100.55									
Llama2-70B	353.03	286.52	85.96	324.52	162.34	99.87	320.41	210.13	100.90	357.95	93.21	106.63	361.53	250.69	121.58									
Llama3-70B	290.19	186.43	61.09	277.49	112.86	51.76	271.60	142.95	54.55	325.85	69.42	71.51	339.23	198.82	78.81									
ChatGPT-3.5	536.79	952.93	155.62	524.81	393.21	179.34	501.36	479.50	154.19	463.04	181.95	191.87	418.75	336.47	130.56									
GPT-4	275.26	160.93	46.61	271.34	105.22	47.55	264.70	132.53	46.16	318.71	62.84	58.09	335.81	193.32	66.02									
<b>LLMLight (with LightGPTs)</b>																								
LightGPT (Qwen2-0.5B)	296.71	193.93	46.80	292.99	129.93	80.63	290.08	169.56	91.20	328.11	70.90	84.49	351.21	233.16	102.39									
LightGPT (Qwen2-7B)	275.92	163.34	48.56	270.41	104.40	44.93	263.10	130.94	45.75	313.37	59.03	50.65	335.30	198.47	72.93									
LightGPT (Llama2-7B)	<u>275.11</u>	<u>161.35</u>	46.38	269.01	102.92	43.06	<u>260.53</u>	<u>127.75</u>	41.84	314.24	59.59	39.66	333.94	191.63	<u>65.49</u>									
LightGPT (Llama3-8B)	<u>275.10</u>	161.92	48.25	268.81	102.54	42.74	262.29	<u>130.32</u>	44.96	<u>311.72</u>	58.40	47.60	333.85	192.06	69.75									
LightGPT (Llama2-13B)	<b>274.03</b>	<b>159.39</b>	<b>43.24</b>	<b>266.94</b>	<b>100.46</b>	<u>40.34</u>	<b>260.17</b>	<b>127.08</b>	41.00	<u>310.78</u>	<u>56.93</u>	<u>38.64</u>	<u>330.71</u>	<u>189.09</u>	<u>64.16</u>									

## 4.5 Comparison Between the Traditional Methods and LLMLight (RQ1)

We first conduct experiments on Jinan and Hangzhou with five real-world traffic flow datasets. The comparison between traditional approaches and LLMLight is shown in Table 2. LightGPT refers to the backbone LLMs trained through our proposed fine-tuning method. The experimental results reveal that LLMLight, equipped with LightGPT, consistently achieves state-of-the-art (SOTA) or comparable performance against all baselines.

In detail, most transportation methods lag behind RL and LLMLight. Among RL models, the CoLight series stands out by integrating neighboring information beyond the target intersection. Although Advanced-CoLight, the SOTA RL method, slightly outperforms LLMLight on Hangzhou datasets, it requires communication among adjacent intersections. In contrast, LLMLight achieves competitive results using observation features solely from the target intersection, showcasing its notable effectiveness. To provide a fair comparison, we also compare Advanced-CoLight without the agent cooperation mechanism and LLMLight, detailed in Appendix A.10. The results show that LLMLight excels in all metrics under this fair setting. We believe LLMLight can be further improved in the future by incorporating multi-agent cooperation.

In our evaluation of LLMLight powered by generalist LLMs, GPT-4 excels and competes with top RL methods, highlighting

**Table 3: Abalation tests of four LightGPT variants.**

Model	Average Travel Time (ATT, s)					
	GPT-4	Qwen2-0.5B	Qwen2-7B	Llama2-7B	Llama3-8B	Llama2-13B
Origin	318.71	1124.84	621.80	973.50	1136.17	722.73
IFT (W/O filtering)	-	<u>788.93</u>	<u>314.33</u>	<u>316.43</u>	<u>316.70</u>	313.32
IFT	-	<u>331.32</u>	<u>314.32</u>	<u>315.94</u>	<u>314.56</u>	312.43
IFT + CGPR	-	<u>328.11</u>	<u>313.37</u>	<u>314.24</u>	<u>311.72</u>	<u>310.78</u>

the effectiveness of our proposed control framework. Llama3-70B and Qwen2-7B rank second and third. Surprisingly, ChatGPT-3.5 shows the least favorable performance, indicating that generalist LLMs may lack the specialized expertise for optimal traffic management. This suggests a potential need for domain-specific LLMs for intelligent traffic control in the future. Appendix A.8 provides more performance comparisons and analysis of their limitations.

Examining the models trained by our proposed method (*i.e.*, LightGPTs), even LLMLight powered by the smallest model, LightGPT (Qwen2-0.5B), outperforms the much larger Llama2-7B. This highlights the effectiveness of our backbone model training approach. While lightweight models with 7B or 8B parameters exhibit a slight performance drop compared to LightGPT (Llama2-13B), they offer more economical alternatives for deployment. Please refer to Appendix A.4 for a detailed cost-effectiveness analysis.

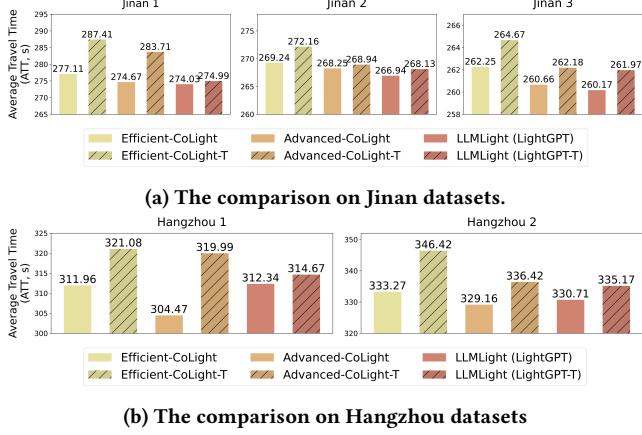


Figure 4: The transferability of different methods.

## 4.6 Abalation Study

To verify the effectiveness of each stage in our proposed backbone LLM training method, we conduct ablation studies on four variants of our proposed LightGPT: (1) Origin: The original LLM without any fine-tuning. (2) IFT (W/O filtering): The LLM trained only with imitation fine-tuning, excluding the high-quality reasoning trajectory filtering strategy. (3) IFT: The LLM trained only with imitation fine-tuning (IFT). (4) IFT + CGPR: The LLM trained through all stages, including imitation fine-tuning (IFT) and critic-guided policy refinement (CGPR). The experiment results on five backbone LLMs are reported in Table 3. The results demonstrate that the effectiveness of each backbone LLM significantly improves after applying our proposed fine-tuning strategy. Furthermore, each training stage consistently enhances the control policy of the LLMs, as evidenced by the progressively improved performance with the incorporation of each proposed stage.

By fine-tuning to imitate the reasoning trajectories of GPT-4, LightGPT demonstrates comparable or even better performance than GPT-4. This highlights the innate capability of the backbone model to excel in traffic signal control tasks if not constrained by their absence of traffic management expertise. Furthermore, The integration of policy refinement corrects the reasoning trajectories and decisions of the backbone LLM, resulting in a discernible enhancement. This implementation attains SOTA or competitive performance across all the datasets and significantly surpasses GPT-4, accentuating the promising prospects for advancing intelligent transportation-oriented LLMs.

## 4.7 Generalization Comparision (RQ2)

In this section, we evaluate the generalization ability of traditional methods and our proposed LLMLight in unseen environments. LLMLight is powered by the LightGPT pre-trained on Llama2-13B. All models undergo uniform pre-training using either the Jinan 1 or Hangzhou 1 dataset. An additional 24-hour cycle experiment is detailed in Appendix A.5.

**4.7.1 Transferability.** We first study the transferability ability of different methods by implementing the pre-trained model in another distinct road network. The experiment results are shown in

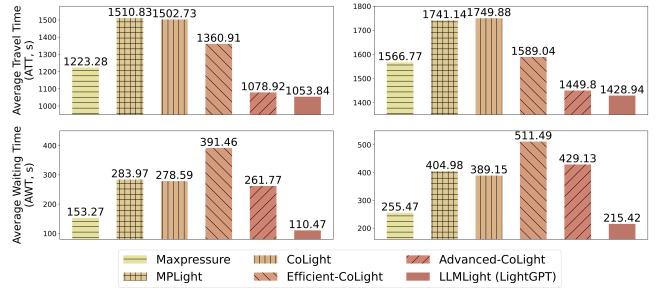


Figure 5: The performance in the large-scale road network.

Figure 4. Models labeled “-T” are pre-trained on a distinct road network (e.g., we evaluate the transferability in Hangzhou using the models pre-trained in Jinan). Otherwise, they are trained and tested in the same dataset. We observe a notable decline in the performance of RL-based methods after the transfer, particularly evident in the Jinan 1 and Hangzhou 1 datasets. In contrast, LLMLight consistently achieves superior performance and demonstrates remarkable transferability across all datasets. These findings highlight LightGPT’s exceptional generalization capability across diverse traffic contexts and robustness for practical implementations.

**4.7.2 Scalability.** Second, we delve into the scalability analysis of both traditional methods and LLMLight, scrutinizing their performance when applied to a considerably larger road network, New York, compared to the original road networks they are pre-trained on (*i.e.*, the Jinan 1 dataset). The results of our experiments are detailed in Figure 5. We can observe significant performance degradations happening on most RL methods, as they demonstrate much worse performance than the heuristic method, Maxpressure. While the best RL methods exhibit suboptimal average travel time (ATT) in contrast to LLMLight, it is noteworthy that their policies result in significantly prolonged waiting times. Specifically, the waiting times are 57.80% and 49.80% longer than those achieved by ours. This observation suggests that the mechanism of the RL methods focuses on minimizing the overall number of queuing vehicles, possibly at the expense of causing extended waiting times for certain queues with fewer vehicles. The significance of minimizing waiting times cannot be overlooked in practical scenarios, as prolonged waits can induce anxiety and constitute a serious concern about the fairness of different drivers. In contrast, our proposed method ensures both the shortest travel and waiting times, highlighting its remarkable scalability and applicability when transitioning to much larger road networks.

**4.7.3 Extreme High-traffic Scenarios.** Next, we delve into an uncommon scenario when considerable traffic flow continuously appears at the intersection, a situation that rarely appears during training. To assess the efficacy of different methods in extreme high-traffic scenarios, we generate two synthetic traffic flow datasets on the Jinan and Hangzhou road networks, featuring an approximately fourfold increase in vehicles arriving within each 5-minute interval compared to the original datasets. Figure 6 shows the performance of both traditional methods and LLMLight armed with LightGPT. All models are pre-trained on regular traffic flow datasets from the same road network (*i.e.*, the Jinan 1 or Hangzhou 1 dataset). Similar

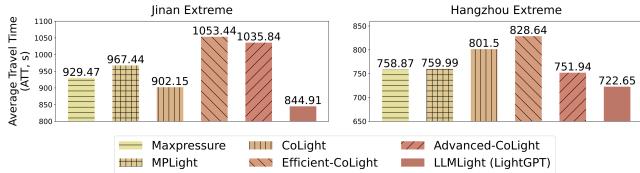


Figure 6: The performance in extreme high-traffic scenarios.

#### Step 1: Analysis

To identify the optimal traffic signal, we need to consider the number of early queued vehicles and approaching vehicles in different segments of the allowed lanes of each signal. However, the early queued vehicles have the most significant impact on traffic congestion, so we should pay the most attention to lanes with long queue lengths.

#### Looking at the data:

- Signal ETWT has a total of 5 early queued vehicles and 5 approaching vehicles in segments 2 and 3.
- Signal NTST has a total of 2 early queued vehicles and 7 approaching vehicles in segments 1, 2, and 3.
- Signal ELWL has no early queued vehicles and only 1 approaching vehicle in segment 3.
- Signal NLSL has a total of 7 early queued vehicles and 3 approaching vehicles in segment 3.

From the above data, it is clear that the signal **NLSL** has the highest number of early queued vehicles (7), which is the most significant factor in traffic congestion. Therefore, relieving the flow of vehicles in the northern and southern left-turn lanes would most significantly improve the traffic condition.

#### Step 2: Answer

Based on the analysis, the most effective traffic signal that will most significantly improve the traffic condition during the next phase is the one that allows vehicles to flow in the northern and southern left-turn lanes (NLSL).

Figure 7: The decision-making process of LightGPT.

to experiments on scalability, RL methods also showcase significant performance degradations, which exhibit poorer results than Maxpressure. This suggests that these pre-trained RL methods struggle with out-of-distribution (OOD) traffic conditions when confronted with a significantly increased volume of vehicles compared to their training stage. In contrast, LLMLight consistently shows superior performance, underscoring its robustness and practicality in much heavier traffic conditions.

## 4.8 Interpretability of LLMLight (RQ3)

To assess LLMLight's interpretability, we analyze a scenario in the Hangzhou dataset. Figure 7 illustrates the decision-making process of LightGPT (Llama2-13B). Detailed traffic conditions are reported in Appendix A.8 (Table 7). In this scenario, The north section encounters severe congestion with a significant queue. LightGPT first understands the task and analyzes traffic at the intersection. Subsequently, it adeptly identifies that the allowed lanes of signal NLSL demonstrate the heaviest congestion situation, thereby selecting it as the optimal signal phase to set. Unlike RL methods, LLMLight not only excels in making effective control policies but also in providing detailed explanations for each decision. This unique feature enhances the transparency and interpretability of LLMLight, contributing to a more informed understanding of its actions.

## 4.9 Deployment Analysis

We are partnering with the Institute of Deep Perception Technology<sup>2</sup> in China to deploy LLMLight in real-world transportation systems<sup>3</sup>. We conduct a human expert evaluation of LLMLight across five randomly selected traffic scenarios to assess its reasoning and

<sup>2</sup><http://www.idpt-jitri.com/>

<sup>3</sup>A prototype system for the public is available at [https://lightgpt2024.github.io/LLMLight\\_Demo/](https://lightgpt2024.github.io/LLMLight_Demo/).

decision-making capabilities in various traffic scenarios. Feedback was gathered from a diverse group: four traffic police officers from the Kunming Southeast Ring Expressway Jinsong Traffic Police Brigade in Yunnan and the Zhongshan Traffic Police Detachment in Guangdong, China, nine experienced drivers, and two AI specialists. The consensus from these evaluations was overwhelmingly positive, with all participants acknowledging LLMLight's reasoning ability and decisions made in all examined scenarios. They highly praised its applicability and notable interpretability in real-world contexts. The detailed evaluation results, studied traffic scenarios, and LightGPT's reasoning processes are shown in Appendix A.6.

Additionally: (1) We conduct a cost-effectiveness analysis comparing LightGPT with GPT-4, detailed in Appendix A.4, which includes training and deployment budgets and efficiency evaluations. (2) A 24-hour experiment in Appendix A.5 validated LightGPT's robustness under daily long-term traffic fluctuations. (3) In Appendix A.7.1, we discuss potential challenges and data privacy preservation methods in real-world deployment. In conclusion, LightGPT consistently demonstrates superior cost-effectiveness and applicability for real-world deployment scenarios.

## 5 RELATED WORK

### 5.1 Traffic Signal Control

Traffic signal control (TSC) methods fall into two main categories: transportation methods and reinforcement learning (RL)-based approaches. Transportation methods include FixedTime [25], which uses a set cycle length and phase split, and Maxpressure [43], which optimizes signal phases to maximize network throughput. PressLight [47] extends Maxpressure by integrating RL to enhance intersection throughput. Recent studies also focus on improving state feature representation for better traffic condition modeling. For instance, Advanced-MP [59] develops advanced state features based on vehicle movement and traffic demand.

### 5.2 Intelligent Transportation

Intelligent transportation applications primarily emphasize forecasting [27, 28, 32] and control [41, 54]. For instance, [26] presents a meta-learning-based framework designed to deliver personalized vehicle energy consumption predictions tailored to individual drivers. Additionally, [61] introduces a dynamic charging pricing model for electric vehicles (EVs) using a multi-agent graph convolutional reinforcement learning approach.

### 5.3 Large Language Models for Decision Making

LLMs recently emerged and exhibited impressive zero-shot reasoning and generalization abilities. Exploiting this proficiency, numerous studies [7, 12, 14, 17, 19, 24, 38, 40, 44, 50, 53] have increasingly employed LLMs as central planners in various control tasks. For example, [18] introduces CityBench, an interactive simulation-based benchmark for evaluating LLMs' capabilities in urban domain perception and decision-making. DriveGPT4 [55] showcases an end-to-end autonomous driving system that interprets vehicle actions and responds to user queries. Additional studies on LLMs in intelligent transportation are detailed in [60, 63].

## 6 CONCLUSION AND LIMITATIONS

This study introduces LLMLight, a framework employing LLMs as traffic signal control agents. By performing a human-like, step-by-step analysis of traffic conditions, LLMLight adjusts traffic lights to improve traffic efficiency. We further propose a backbone LLM training procedure for developing the TSC-oriented LLM, LightGPT. Extensive experiments on ten traffic datasets show that LLMLight with LightGPT significantly outperforms nine baseline methods and ten leading LLMs, demonstrating its superior effectiveness, generalization, and interoperability across diverse traffic scenarios.

Our findings include: (1) Traditional traffic signal control methods have limited generalization capabilities, leading to suboptimal performance in unseen traffic scenarios. (2) Generalist LLMs lack in-depth traffic management understanding, but tailored prompting and domain-specific fine-tuning can significantly improve their control policies. (3) LLM-empowered traffic signal control demonstrates superior performance across diverse scenarios and can provide detailed explanations for each decision, contributing to a more comprehensible and accountable traffic control system.

Our limitations include: (1) Our approach focuses solely on traffic features at the target intersection, neglecting multi-agent cooperation mechanisms for global traffic improvement. (2) We do not incorporate additional modality features, such as camera images, to aid the LLM agent in decision-making. (3) The framework primarily regulates vehicle traffic flow and does not consider the impact of other dynamic factors at the intersection, such as pedestrians and bicycles. We plan to address these limitations to enhance the effectiveness of the LLMLight framework in the future.

## REFERENCES

- [1] Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. 2003. Reinforcement Learning for True Adaptive Traffic Signal Control. *Journal of Transportation Engineering* 129, 3 (2003), 278–285.
- [2] Oluwaseun John Adeyemi, Ahmed A. Arif, and Rajib Paul. 2021. Exploring the Relationship of Rush Hour Period and Fatal and Non-Fatal Crash Injuries In The U.S.: A Systematic Review and Meta-Analysis. *Accident; analysis and prevention* 163 (2021), 106462.
- [3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Bin Yuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. *CoRR* abs/2309.16609 (2023). arXiv:2309.16609
- [4] Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2023. Grounding Large Language Models in Interactive Environments with Online Reinforcement Learning. In *International Conference on Machine Learning, ICML 2023, 23–29 July 2023, Honolulu, Hawaii, USA (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 3676–3713.
- [5] Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. FireAct: Toward Language Agent Fine-tuning. *CoRR* abs/2310.05915 (2023). arXiv:2310.05915
- [6] Chaicha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*. AAAI Press, 3414–3421.
- [7] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. 2023. Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving. *CoRR* abs/2310.01957 (2023). arXiv:2310.01957
- [8] Zhao Chenguang, Hu Xiaorong, and Wang Gang. 2021. PRG-Light: A Novel Traffic Light Control Framework with Pressure-based Reinforcement Learning and Graph Neural Network. In *IJCAI 2021 Reinforcement Learning for Intelligent Transportation Systems (RL4ITS) Workshop*, August, 2021, virtual.
- [9] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 4299–4307. <https://proceedings.neurips.cc/paper/2017/hash/d5e2c0dad503c91f1df240d0cd4e49-Abstract.html>
- [10] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. *CoRR* abs/2210.11416 (2022). arXiv:2210.11416
- [11] Seung-Bae Cools, Carlos Gershenson, and Bart D’Hooghe. 2013. Self-organizing Traffic Lights: A Realistic Simulation. *Advances in applied self-organizing systems* (2013), 45–55.
- [12] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. 2023. Drive as You Speak: Enabling Human-Like Interaction with Large Language Models in Autonomous Vehicles. *CoRR* abs/2309.10228 (2023). arXiv:2309.10228
- [13] Longchao Da, Minchuan Gao, Hao Mei, and Hua Wei. 2023. LLM Powered Sim-to-real Transfer for Traffic Signal Control. *CoRR* abs/2308.14284 (2023). arXiv:2308.14284
- [14] Longchao Da, Kuanru Liou, Tiejin Chen, Xuesong Zhou, Xiangyong Luo, Yezhou Yang, and Hua Wei. 2024. Open-TI: Open Traffic Intelligence with Augmented Language Model. *CoRR* abs/2401.00211 (2024). arXiv:2401.00211
- [15] Samah El-Tantawy and Baher Abdulhai. 2010. An Agent-based Learning Towards Decentralized and Coordinated Traffic Signal Control. In *13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Madeira, Portugal, 19–22 September 2010*. IEEE, 665–670.
- [16] Torantulino et al. 2023. AutoGPT. <https://github.com/Significant-Gravitas/AutoGPT>. Accessed on 1 December 2023.
- [17] Jie Feng, Yuwei Du, Tianhui Liu, Siqi Guo, Yuming Lin, and Yong Li. 2024. CityGPT: Empowering Urban Spatial Cognition of Large Language Models. *CoRR* abs/2406.13948 (2024). arXiv:2406.13948
- [18] Jie Feng, Jun Zhang, Junbo Yan, Xin Zhang, Tianjian Ouyang, Tianhui Liu, Yuwei Du, Sqi Guo, and Yong Li. 2024. CityBench: Evaluating the Capabilities of Large Language Model as World Model. *CoRR* abs/2406.13945 (2024). arXiv:2406.13945
- [19] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. 2023. Drive Like a Human: Rethinking Autonomous Driving with Large Language Models. *CoRR* abs/2307.07162 (2023). arXiv:2307.07162
- [20] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022*. OpenReview.net.
- [21] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *CoRR* abs/2311.05232 (2023). arXiv:2311.05232
- [22] PB Hunt, DI Robertson, RD Bretherton, and M Cr Royle. 1982. The SCOOT On-line Traffic Signal Optimisation Technique. *Traffic Engineering & Control* 23, 4 (1982).
- [23] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* 55, 12 (2023), 248:1–248:38.
- [24] Ye Jin, Xiaoxi Shen, Huijing Peng, Xiaona Liu, Jingli Qin, Jiayang Li, Jintao Xie, Peizhong Gao, Guyue Zhou, and Jiangtao Gong. 2023. SurrealDriver: Designing Generative Driver Agent Simulation Framework in Urban Contexts based on Large Language Model. *CoRR* abs/2309.13193 (2023). arXiv:2309.13193
- [25] Peter Koonce and Lee Rodegerds. 2008. *Traffic Signal Timing Manual*. Technical Report. United States. Federal Highway Administration.
- [26] Siqi Lai, Weijia Zhang, and Hao Liu. 2023. A Preference-aware Meta-optimization Framework for Personalized Vehicle Energy Consumption Estimation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6–10, 2023*. ACM, 4346–4356.
- [27] Fan Liu, Hao Liu, and Wenzhai Jiang. 2022. Practical Adversarial Attacks on Spatiotemporal Traffic Forecasting Models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 – December 9, 2022*.
- [28] Fan Liu, Weijia Zhang, and Hao Liu. 2023. Robust Spatiotemporal Traffic Forecasting with Reinforced Dynamic Adversarial Training. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6–10, 2023*. ACM, 1417–1428.

- [29] P Lowrie. 1990. Scats-a Traffic Responsive Method of Controlling Urban Traffic. *Sales information brochure published by Roads & Traffic Authority, Sydney, Australia* (1990).
- [30] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. 2023. GPT-Driver: Learning to Drive with GPT. *CoRR* abs/2310.01415 (2023). arXiv:2310.01415
- [31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level Control Through Deep Reinforcement Learning. *Nature* 518, 7540 (2015), 529–533.
- [32] Yansong Ning, Hao Liu, Hao Wang, Zhenyu Zeng, and Hui Xiong. 2023. UUKG: Unified Urban Knowledge Graph Dataset for Urban Spatiotemporal Prediction. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [33] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023). arXiv:2303.08774
- [34] OpenAI. 2023. Introducing ChatGPT. <https://openai.com/blog/chatgpt>. Accessed on 1 December 2023.
- [35] Afshin Oroojlooy, MohammadReza Nazari, Davood Hajinezhad, and Jorge Silva. 2020. AttendLight: Universal Attention-Based Reinforcement Learning Model for Traffic Signal Control. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- [36] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [37] Aoyu Pang, Maonan Wang, Man-On Pun, Chung Shue Chen, and Xi Xiong. 2024. iLLM-TSC: Integration reinforcement learning and large language model for traffic signal control policy improvement. *arXiv preprint arXiv:2407.06025* (2024).
- [38] Timo Schick, Jane Dwivedi-Yu, Roberto Densi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *CoRR* abs/2302.04761 (2023). arXiv:2302.04761
- [39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347 (2017). arXiv:1707.06347
- [40] Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. 2023. LanguageMPC: Large Language Models as Decision Makers for Autonomous Driving. *CoRR* abs/2310.03026 (2023). arXiv:2310.03026
- [41] Qian Sun, Le Zhang, Huan Yu, Weijia Zhang, Yu Mei, and Hui Xiong. 2023. Hierarchical Reinforcement Learning for Dynamic Autonomous Vehicle Navigation at Intelligent Intersections. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, ACM, 4852–4861.
- [42] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenjin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardasz, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jinya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Miaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poultoun, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR* abs/2307.09288 (2023). arXiv:2307.09288
- [43] Pravin Varaiya. 2013. Max Pressure Control of a Network of Signalized Intersections. *Transportation Research Part C: Emerging Technologies* 36 (2013), 177–195.
- [44] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *CoRR* abs/2305.16291 (2023). arXiv:2305.16291
- [45] Maonan Wang, Aoyu Pang, Yuheng Kan, Man-On Pun, Chung Shue Chen, and Bo Huang. 2024. LLM-Assisted Light: Leveraging Large Language Model Capabilities for Human-Mimetic Traffic Signal Control in Complex Urban Environments. *arXiv preprint arXiv:2403.08337* (2024).
- [46] Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhipang Sui. 2023. Making Large Language Models Better Reasoners with Alignment. *CoRR* abs/2309.02144 (2023). arXiv:2309.02144
- [47] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash V. Gayah, Kai Xu, and Zhenhui Li. 2019. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, ACM, 1290–1298.
- [48] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. CoLight: Learning Network-level Cooperation for Traffic Signal Control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, ACM, 1913–1922.
- [49] Hua Wei, Guanjie Zheng, Vikash V. Gayah, and Zhenhui Li. 2019. A Survey on Traffic Signal Control Methods. *CoRR* abs/1904.08117 (2019). arXiv:1904.08117
- [50] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. 2023. DiLu: A Knowledge-Driven Approach to Autonomous Driving with Large Language Models. *CoRR* abs/2309.16292 (2023). arXiv:2309.16292
- [51] Qiang Wu, Mingyuan Li, Jun Shen, Linyuan Lü, Bo Du, and Ke Zhang. 2023. TransformerLight: A Novel Sequence Modeling Based Traffic Signaling Mechanism via Gated Transformer. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, ACM, 2639–2647.
- [52] Qiang Wu, Liang Zhang, Jun Shen, Linyuan Lü, Bo Du, and Jianqing Wu. 2021. Efficient Pressure: Improving Efficiency for Signalized Intersections. *CoRR* abs/2112.02336 (2021). arXiv:2112.02336
- [53] Huatao Xu, Liying Han, Qirui Yang, Mo Li, and Mani Srivastava. 2024. Penetrative AI: Making LLMs Comprehend the Physical World. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 7324–7341.
- [54] Qian Xu, Zhiguang Liu, and Zhigang Xu. 2022. A Novel Ramp Metering Algorithm based on Deep Reinforcement Learning. In *2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI)*, 2022, IEEE, 128–133.
- [55] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee. K. Wong, Zhenguo Li, and Hengshuang Zhao. 2023. DriveGPT4: Interpretable End-to-end Autonomous Driving via Large Language Model. *CoRR* abs/2310.01412 (2023). arXiv:2310.01412
- [56] Zheng Yuan, Hongyi Yuan, Chuanchi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: Rank Responses to Align Language Models with Human Feedback without tears. *CoRR* abs/2304.05302 (2023). arXiv:2304.05302
- [57] Aohan Zeng, Mingdiao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. AgentTuning: Enabling Generalized Agent Abilities for LLMs. *CoRR* abs/2310.12823 (2023). arXiv:2310.12823
- [58] Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2023. Reinforcement Learning for Traffic Signal Control. <http://traffic-signal-control.github.io>. Accessed on 1 December 2023.
- [59] Liang Zhang, Qiang Wu, Jun Shen, Linyuan Lü, Bo Du, and Jianqing Wu. 2022. Expression Might be Enough: Representing Pressure and Demand for Reinforcement Learning Based Traffic Signal Control. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*, PMLR, 26645–26654.
- [60] Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Hao Liu, and Hui Xiong. 2024. Towards Urban General Intelligence: A Review and Outlook of Urban Foundation Models. *CoRR* abs/2402.01749 (2024). arXiv:2402.01749
- [61] Weijia Zhang, Hao Liu, Jindong Han, Yong Ge, and Hui Xiong. 2022. Multi-Agent Graph Convolutional Reinforcement Learning for Dynamic Electric Vehicle Charging Pricing. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, ACM, 2471–2481.
- [62] Weijia Zhang, Le Zhang, Jindong Han, Hao Liu, Jingbo Zhou, Yu Mei, and Hui Xiong. 2023. Irregular Traffic Time Series Forecasting Based on Asynchronous Spatio-Temporal Graph Convolutional Network. *CoRR* abs/2308.16818 (2023). arXiv:2308.16818
- [63] Xingcheng Zhou, Mingyu Liu, Bare Luka Zagar, Ekim Yurtsever, and Alois C. Knoll. 2023. Vision Language Models in Autonomous Driving and Intelligent Transportation Systems. *CoRR* abs/2310.14414 (2023). arXiv:2310.14414

## A APPENDIX

### A.1 Settings of Intersection, Lanes, and Signal Phases

We present the most used setting of intersection, lanes, and signal phases in Figure 8.

### A.2 Compared Models

#### A.2.1 Transportation Methods.

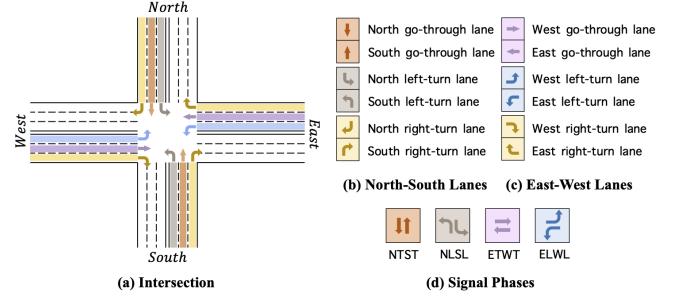
- **Random**: A baseline policy randomly switches signal phases with a fixed duration.
- **FixedTime** [25]: A policy gives a pre-defined fixed cycle length with phase time, widely implemented in most traffic scenarios.
- **Maxpressure** [43]: The state-of-the-art transportation-based method, which greedily relieves vehicles on the lanes with the highest pressure (a variable derived from the difference between upstream and downstream queue lengths).

#### A.2.2 RL-based Methods.

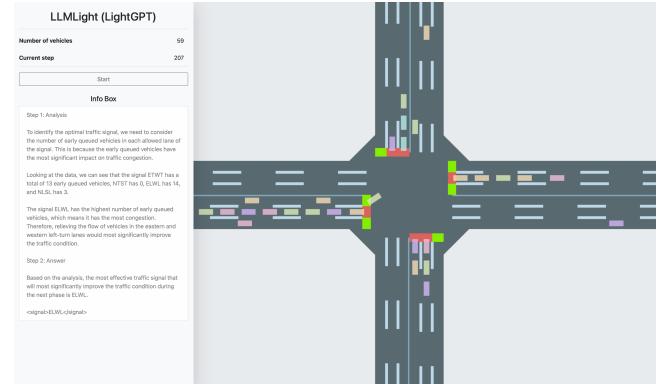
- **MPLight** [6]: A reinforcement learning (RL)-based method that utilizes pressure as both observation and reward. It employs FRAP, a network structure specifically designed to manage unbalanced traffic flow, as the foundational model.
- **AttendLight** [35]: Leveraging an attention mechanism, this model constructs observation features and predicts phase transition probability.
- **PressLight** [47]: Leveraging the concept of Maxpressure with DRL, optimizing the pressure of the intersection.
- **CoLight** [48]: This model employs a graph attention network (GAT) to enable communication among neighboring intersections, enhancing coordination and decision-making.
- **Efficient-CoLight** [52]: Building upon the CoLight model, it integrates efficient pressure as an observation, optimizing its decision-making capabilities.
- **Advanced-CoLight** [59]: Building upon the CoLight model, it incorporates advanced traffic state features, achieving the SOTA performance in traffic signal control tasks.

#### A.2.3 Large Language Models.

- **Llama** [42]: This is a series of pre-trained and finely-tuned large language models (LLMs) developed by Meta, ranging in scale from 7B to 70B parameters. They are adaptable for various tasks related to natural language generation. We use the second- and third-generation, Llama-2 and Llama-3, in our experiments.
- **Qwen2** [3]: Alibaba's Qwen2 series are LLMs with parameters ranging from 0.5B to 72B, supporting up to 128K tokens context, enhancing capabilities in natural language understanding, code writing, and multi-language processing.
- **ChatGPT-3.5** [34]: Developed by OpenAI, ChatGPT-3.5 is a chatbot explicitly designed for engaging in conversations, answering queries, and assisting with diverse tasks. We adopt gpt-3.5-turbo-0613 API from OpenAI.
- **GPT-4** [33]: The newest advancement in LLM by OpenAI. It is more reliable and creative compared to GPT-3.5. It excels in handling significantly more nuanced instructions and tasks. We utilize gpt-4-0613 API in our experiments.



**Figure 8: An illustration of intersection, lanes, and signal phases.** Lanes sharing a common color signify non-conflicting movements, and the passage is allowed in the corresponding signal phase. Signal phases include ETWT (go-through from east and west), ELWL (left-turn from east and west), NTST (go-through from north and south), and NLSL (left-turn from north and south).



**Figure 9: Prototype system.**

### A.3 Prototype System

Figure 9 shows a snapshot of our demo system. We provide users with real-time visualizations of the current traffic conditions on the right, along with statistical data on vehicles in the upper left corner. Additionally, the system further presents the reasoning behind LightGPT’s decisions through a message box in the interface’s lower left section.

### A.4 Cost-effectiveness of LightGPT

To assess the cost-effectiveness of our proposed LightGPT with the most widely used and advanced LLM, GPT-4. We report the annual deployment cost in different city scales, budget in data collection and model training, and efficiency evaluation of backbone LLMs in Tables 4, 5, and 6, respectively. These calculations were performed on a cloud computing machine with  $4 \times$  A800-80GB GPUs, 56 cores, 400GB of memory, and a 1TB SSD. Please note that our model weights have been open-sourced. With its exceptional generalization ability, it can be directly deployed in most unseen traffic environments without further fine-tuning.

**Table 4: The annual deployment cost (in USD) of backbone LLMs. LightGPT-8B and LightGPT-13B states LightGPT (Llama3-8B) and LightGPT (Llama2-13B), respectively.**

City scale	Small			Medium			Large			Metropolitan		
# of traffic lights	100			500			1000			5000		
Model	GPT-4	LightGPT-13B	LightGPT-8B	GPT-4	LightGPT-13B	LightGPT-8B	GPT-4	LightGPT-13B	LightGPT-8B	GPT-4	LightGPT-13B	LightGPT-8B
# of machines	-	2	1	-	10	5	-	20	10	-	100	50
Annual cost	\$1,680K	\$44.91K	\$22.45K	\$8,400K	\$224.54K	\$112.27K	\$16,800K	\$449.08K	\$224.54K	\$84,100K	\$2,250K	\$1,120K

**Table 5: The training budget (in USD) of LightGPT.**

	Data Collection	IFT (1 GPU)	CGPR (4 GPUs)	Sum
Cost	\$135.53	\$11.60	\$5.52	\$152.65

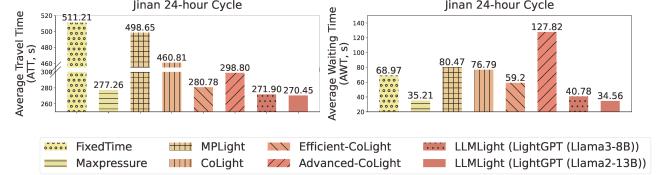
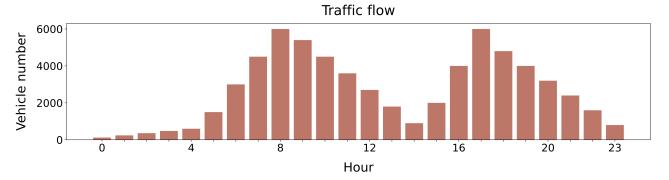
**Table 6: The efficiency (seconds per batch) of backbone LLMs.**

# of traffic intersections (batch size)	1	5	10	15	20
GPT-4	8.60s	9.23s	11.69s	11.35s	12.95s
LightGPT (Llama2-13B)	3.53s	4.61s	5.31s	5.74s	6.62s
LightGPT (Llama3-8B)	2.52s	3.30s	3.85s	4.15s	4.69s

For real-time efficiency in deployment, we set the number of parallelly controlled intersections to 20 for LightGPT (Llama3-8B) and 10 for LightGPT (Llama2-13B). When controlling asynchronously, one machine with LightGPT (Llama3-8B) can serve up to 100 intersections, while LightGPT (Llama2-13B) can serve up to 50. We can observe that LightGPT is the most cost-effective solution, as it achieves the best performance in both effectiveness (see Section 4) and real-time traffic control, while also having the lowest deployment cost. Further efficiency and cost-effectiveness improvements are planned for future work.

## A.5 24-hour Cycle Experiment

We extend our experiment to a full 24-hour period to examine model performance under daily long-term traffic fluctuations, including distinct morning and evening rush hours. Detailed traffic flow distribution of this dataset can be found in Figure 11. All models are pre-trained using the Jinan 1 dataset. The results are presented in Figure 10. Consistent with previous experiments, we observe that the performance of RL methods drops significantly when traffic flow shifts across different times of the day, as evidenced by their poorer results compared to Maxpressure. As concluded in Subsection 4.7.2, RL methods result in significantly longer waiting times than Maxpressure and LLMLight, due to their focus on minimizing overall queuing vehicles, which may possibly extend the waiting times of queues with fewer vehicles. In contrast, our proposed method consistently ensures both the shortest travel and waiting times, demonstrating its robustness and real-world applicability in diverse and fluctuating traffic conditions. Moreover, LLMLight significantly outperforms the widely used FixedTime method in real-world scenarios, further underscoring the practical benefits of implementing our approach in real-world deployments.

**Figure 10: The performance in a 24-hour period.****Figure 11: The traffic flow distribution in the 24-hour.**

## A.6 Human Expert Evaluation

We conduct a human evaluation across five randomly selected traffic scenarios to evaluate LightGPT’s reasoning and decision-making capabilities under diverse traffic environments. Feedback was solicited from a diverse group: four traffic police officers from the Kunming Southeast Ring Expressway Jinsong Traffic Police Brigade in Yunnan and the Zhongshan Traffic Police Detachment in Guangdong, China, nine experienced drivers, and two AI specialists. The evaluation results, the studied traffic scenarios, and the reasoning process of LightGPT are presented in Figure 12-16. The consensus from these evaluations was overwhelmingly positive, with participants praising its applicability and notable interpretability in real-world contexts.

## A.7 Other Deployment Considerations

**A.7.1 Challenges in Real-world Deployment.** We outline several challenges that may arise during real-world deployment: (1) *Integration with existing systems*: LLMLight relies on traffic cameras or ground sensing coils for data collection. While these devices are common in most traffic scenarios, modifications might be needed to integrate our framework seamlessly with current systems. (2) *Regulatory compliance*: Our experiments have primarily been conducted in China and the U.S., which have different regulatory frameworks compared to other countries. Variations in regulations, such as driver seat positioning and road rules, may require additional adjustments to tailor our model for deployment in various regions. (3) *Impact on driver behavior*: Unlike traditional traffic signal control (TSC) systems, which follow a predetermined sequence, intelligent TSC systems like LLMLight introduce unpredictability in signal

**Table 7: The statistics of the simulation case.**

Signal	Allowed Lanes	Queue Length	Approaching
ETWT	East and west go-through	5	5
ELWL	East and west left-turn	0	1
NTST	North and south go-through	2	7
NSL	North and south left-turn	7	3

**Table 8: The performances of LLMLight with the original LLMs (without fine-tuning) and LightGPTs on the Jinan 1 and Hangzhou 1 datasets.**

Methods	Jinan 1			Hangzhou 1		
	ATT	AQL	AWT	ATT	AQL	AWT
<b>LLMLight (with Generalist LLMs)</b>						
Qwen2-0.5B	1306.46	1771.49	953.10	1124.84	789.44	880.79
Qwen2-7B	618.86	790.53	469.08	788.98	497.46	616.32
Llama2-7B	1115.97	1580.44	896.20	973.50	664.58	819.00
Llama3-8B	983.48	1236.04	830.66	1136.17	643.44	806.26
Llama2-13B	508.72	567.61	169.67	722.72	418.64	311.55
<b>LLMLight (with LightGPTs)</b>						
LightGPT (Qwen2-0.5B)	296.71	193.93	46.80	328.11	70.90	84.49
LightGPT (Qwen2-7B)	275.92	163.34	48.56	313.37	59.03	50.65
LightGPT (Llama2-7B)	275.11	161.35	46.38	314.24	59.59	39.66
LightGPT (Llama3-8B)	275.10	161.92	48.25	311.72	58.40	47.60
LightGPT (Llama2-13B)	<b>274.03</b>	<b>159.39</b>	<b>43.24</b>	<b>310.78</b>	<b>56.93</b>	<b>38.64</b>

switching. This unpredictability could potentially influence driving behaviors. Further adjustments to address these challenges are planned for future work.

**A.7.2 Data Privacy.** For real-world deployment, we plan to utilize existing infrastructure like traffic cameras and ground sensing coils for observation feature collection. Our system solely extract non-personal data (*i.e.*, vehicle counts) to obtain traffic conditions. Any images captured will be securely stored on the traffic department's server and will not be made public, ensuring the maintenance of privacy integrity.

## A.8 Failure of Other LLMs

To explore the shortcomings of other LLMs, we analyze the reasoning processes of Llama-2 and ChatGPT-3.5, as illustrated in Figure 17. Table 7 details the traffic conditions, highlighting severe congestion in the north and south sections, with notable queues in the left-turn lane and a heavy influx in the go-through lane. Llama-2 models commonly show limitations in traffic management expertise and instruction-following, often disregarding provided commonsense knowledge. Although they recognize the need to address the most congested lanes, Llama-2 prioritizes approaching vehicles, overlooking the critical queuing vehicles. Conversely, ChatGPT-3.5 exhibits a major logical flaw, claiming that releasing lanes with the lowest vehicle count is most effective for improving traffic conditions. This observation suggests a significant hallucination issue [21, 23] in ChatGPT-3.5's traffic signal control reasoning. The performances of more generalist LLMs are detailed in Table 8.

**Table 9: The comparison between PPO and CGPR.**

Methods	Jinan 1			Hangzhou 1		
	ATT	AQL	AWT	ATT	AQL	AWT
LLMLight (LightGPT (PPO))	275.35	164.64	57.07	315.32	60.63	50.63
LLMLight (LightGPT (CGPR))	<b>274.03</b>	<b>159.39</b>	<b>43.24</b>	<b>310.78</b>	<b>56.93</b>	<b>38.64</b>

**Table 10: The comparison between the SOTA RL and LLMLight without cooperation.**

Methods	Hangzhou 1			Hangzhou 2		
	ATT	AQL	AWT	ATT	AQL	AWT
Advanced-CoLight (W/O COOP)	<b>310.48</b>	58.02	54.54	332.08	192.97	88.51
LLMLight (LightGPT (Llama2-13B))	310.78	<b>56.93</b>	<b>38.64</b>	<b>330.71</b>	<b>189.09</b>	<b>64.16</b>

## A.9 Experiment on Policy Network Learning by Using the Actor-Critic Paradigm

We explore using the actor-critic paradigm, specifically Proximal Policy Optimization (PPO) [39], to learn the traffic control policy. Similar to reinforcement learning with human feedback (RLHF) [9], we integrate a multi-layer perceptron (MLP) into the final decoder layer of the LLM as the value function head. The reward function is based on the negative queue length at the target intersection. PPO is employed to facilitate automatic exploration and learning of the optimal control policy and value function. However, this approach struggled to converge and yielded unsatisfactory results, as reported by [46]. Consequently, we adopted our proposed critic-guided policy refinement (CGPR), which uses the critic model for supervised alignment training, ensuring robust performance. Table 9 compares the PPO algorithm and our proposed CGPR on the Jinan 1 and Hangzhou 1 datasets.

## A.10 Comparison with the SOTA RL-based Method Under a Fair Setting

Despite Advanced-CoLight's marginal advantage in the Hangzhou datasets, note it requires additional communication among neighboring intersections. However, LLMLight achieves competitive results by leveraging observation features solely from the target intersection. We conducted an ablation study, removing the multi-agent cooperation component from Advanced-CoLight, denoted as Advanced-CoLight (W/O COOP). The experiment results are reported in Table 10. In this comparison, LLMLight showcased superior performance in ATT, AQL, and AWT on the Hangzhou dataset. In summary, LLMLight achieves better performance with the same input, and we believe LLMLight can be further improved in the future by incorporating multi-agent cooperation mechanisms.

## A.11 Visualizations of Road Networks

The visualizations of the road networks are shown in Figure 18.

## A.12 Prompt Template

We present the prompt template of LLMLight in Table 11.

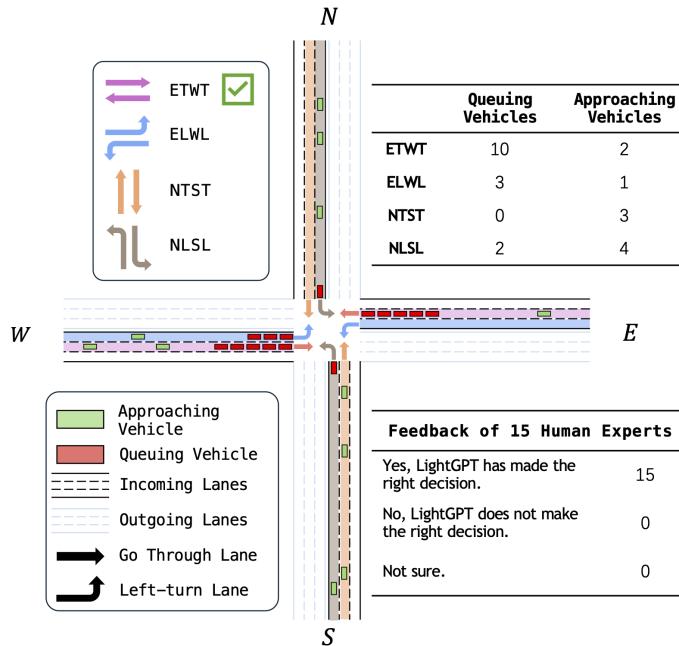


Figure 12: Human expert evaluation: Case 1.

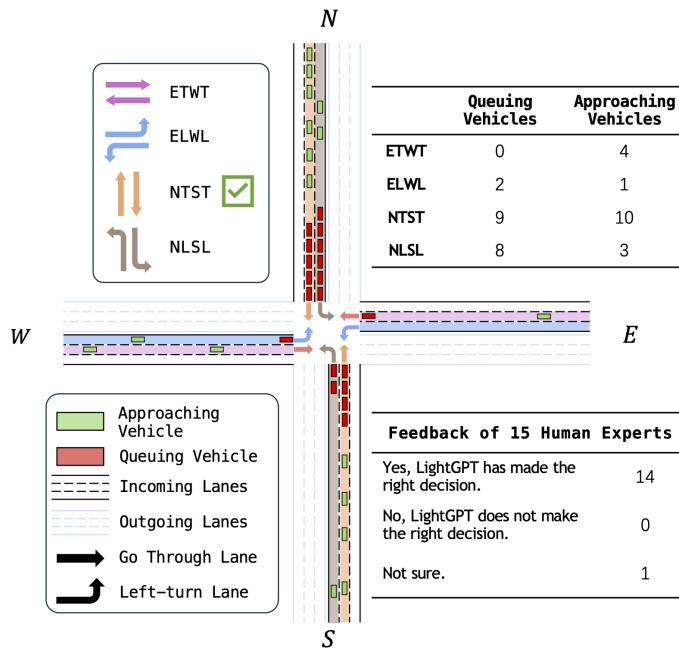


Figure 13: Human expert evaluation: Case 2.

**Step 1: Analysis**

To identify the optimal traffic signal, we need to consider the number of early queued vehicles and approaching vehicles in each lane. The early queued vehicles have the most significant impact on traffic congestion, so we should prioritize the lanes with the most early queued vehicles.

Looking at the data, we can see that the signal ETWT has a total of 10 early queued vehicles, NTST has 0, ELWL has 7, and NLSL has 2. Therefore, the signal ETWT has the most early queued vehicles, which means it has the highest potential to relieve traffic congestion. However, we also need to consider the approaching vehicles in different segments. Although they have less impact on the current traffic situation, they will become early queued vehicles in the future. The signal ETWT has a total of 2 approaching vehicles, NTST has 3, ELWL has 2, and NLSL has 4.

Considering both early queued vehicles and approaching vehicles, the signal ETWT still has the highest total number of vehicles that can be relieved.

**Step 2: Answer**

Based on the analysis, the most effective traffic signal that will most significantly improve the traffic condition during the next phase is **ETWT**.

**Step 1: Analysis**

To identify the optimal traffic signal, we need to consider the number of early queued vehicles and approaching vehicles in different segments of the allowed lanes of each signal. As the problem states, the traffic congestion is primarily dictated by the early queued vehicles, so we should pay the most attention to lanes with long queue lengths.

Let's analyze each signal:

- Signal ETWT: There are no early queued vehicles, and only a few approaching vehicles in segments 2 and 3. This signal does not seem to be the most effective at this time.
- Signal NTST: There are 9 early queued vehicles, which is the highest among all signals. There are also several approaching vehicles in segments 2 and 3. This signal seems to be a good candidate.
- Signal ELWL: There are only 2 early queued vehicles, and only one approaching vehicle in segment 2. This signal does not seem to be the most effective at this time.
- Signal NLSL: There are 8 early queued vehicles, which is the second highest among all signals. There are also a few approaching vehicles in segments 2 and 3. This signal seems to be a good candidate, but not as effective as NTST.

**Step 2: Answer**

Based on the analysis, the most effective traffic signal that will most significantly improve the traffic condition during the next phase is the one that relieves the most vehicles' flow of the allowed lanes of the signal. In this case, the signal with the highest number of early queued vehicles and approaching vehicles in segments 2 and 3 is **NTST**.

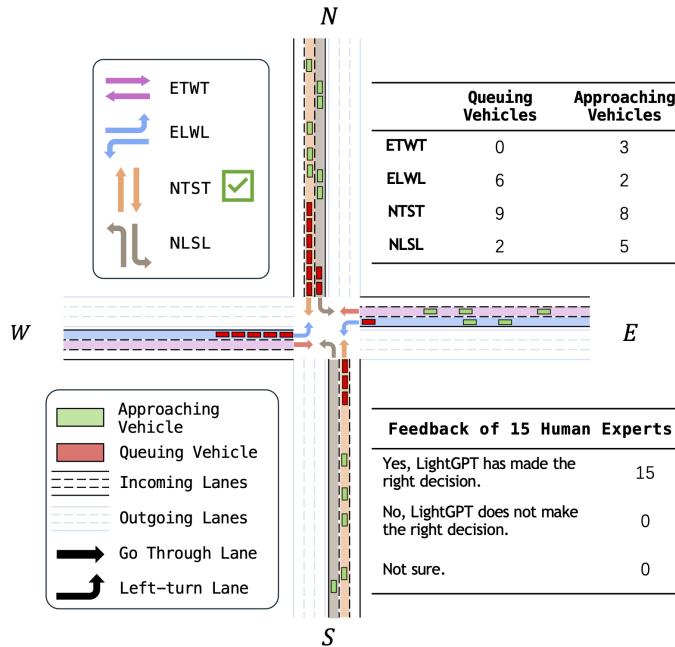


Figure 14: Human expert evaluation: Case 3.

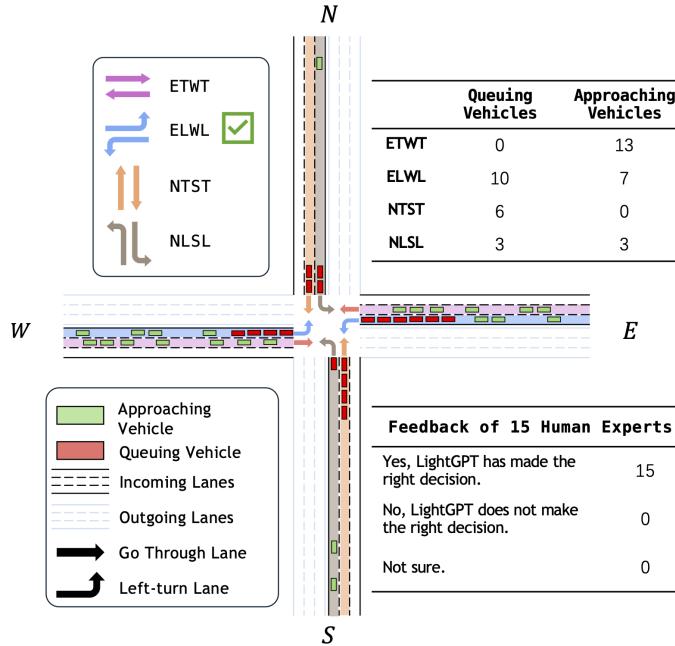


Figure 15: Human expert evaluation: Case 4.

**Step 1: Analysis**

To identify the optimal traffic signal, we need to consider the number of early queued vehicles and approaching vehicles in different segments of the allowed lanes of each signal. As the problem states, the traffic congestion is primarily dictated by the early queued vehicles, so we should pay the most attention to lanes with long queue lengths.

Let's analyze each signal:

- Signal ETWT: There are no early queued vehicles and only 3 approaching vehicles in total. This signal does not seem to be the most effective at this time.
- Signal NTST: There are 9 early queued vehicles and 8 approaching vehicles in total. This signal seems to be a good candidate as it has the highest number of early queued vehicles.
- Signal ELWL: There are 6 early queued vehicles and 2 approaching vehicles in total. This signal is a possible candidate, but it has fewer early queued vehicles than NTST.
- Signal NLSL: There are 2 early queued vehicles and 5 approaching vehicles in total. This signal has fewer early queued vehicles and approaching vehicles than the other signals.

**Step 2: Answer**

Based on the analysis, the most effective traffic signal that will most significantly improve the traffic condition during the next phase is the one that relieves the most early queued vehicles. Therefore, the optimal traffic signal is **NTST**, which has the highest number of early queued vehicles.

**Step 1: Analysis**

To identify the optimal traffic signal, we need to consider the number of early queued vehicles and approaching vehicles in different segments of the allowed lanes of each signal. As the problem states, the traffic congestion is primarily dictated by the early queued vehicles, so we should pay the most attention to lanes with long queue lengths.

Let's analyze each signal:

- Signal ETWT: There are no early queued vehicles, but there are 13 approaching vehicles in total. However, most of these vehicles are in segment 3, which is the farthest from the intersection, so they are not urgent to consider.
  - Signal NTST: There are 6 early queued vehicles, but no approaching vehicles. This signal could relieve the immediate congestion, but it wouldn't affect the future traffic flow.
  - Signal ELWL: There are 10 early queued vehicles, which is the highest among all signals. There are also 7 approaching vehicles in total, but most of them are in segment 3.
  - Signal NLSL: There are 3 early queued vehicles and 3 approaching vehicles. This signal has the least impact on relieving traffic congestion.
- Step 2: Answer**
- Based on the analysis, the signal that will most significantly improve the traffic condition during the next phase is the one that relieves the most early queued vehicles. Therefore, the optimal traffic signal is **ELWL**, which has the highest number of early queued vehicles (10 in total).

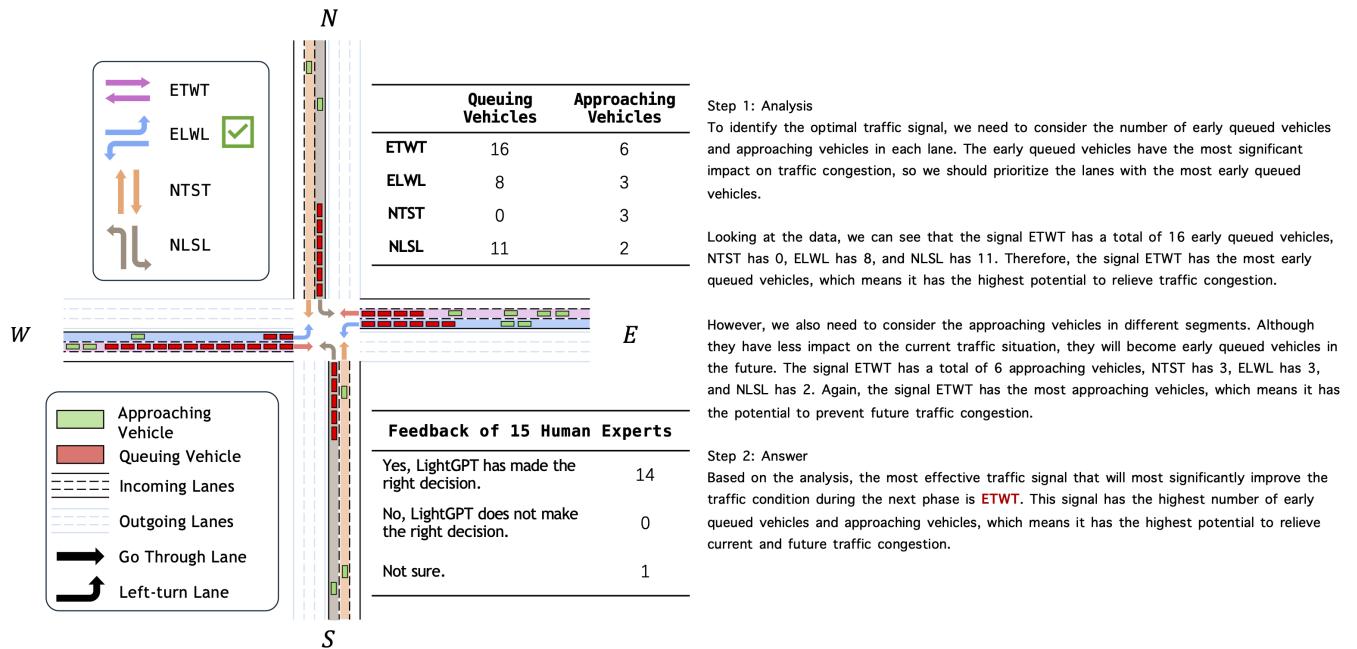


Figure 16: Human expert evaluation: Case 5.

**Reasoning of Llama-2-13B:**

...Therefore, my chosen signal is `<signal>NTST</signal>`. This signal will relieve the high demand for northbound traffic, which is currently experiencing a high number of approaching vehicles. By allowing northbound traffic to proceed before southbound traffic, we can improve the traffic condition and reduce congestion in all directions.

**Reasoning of Llama-2-70B:**

...Based on the analysis, the optimal traffic signal is `<signal>NTST</signal>`. It addresses the bottleneck in the through lanes and maximizes the throughput of vehicles.

Explanation:

While **ELWL** does not address the bottleneck, it can help to clear out the left-turn lanes, which can improve the overall traffic condition. However, **NTST** addresses the bottleneck directly and can maximize the throughput of vehicles. Therefore, NTST is the optimal traffic signal in this scenario.

**Reasoning of ChatGPT-3.5:**

...Based on the analysis, the signal with the lowest total number of vehicles and early queued vehicles is the most effective in improving traffic conditions.

Step 2: Answer your chosen signal.

`<signal>ELWL</signal>`

Figure 17: The reasoning of Llama-2 models and ChatGPT-3.5.

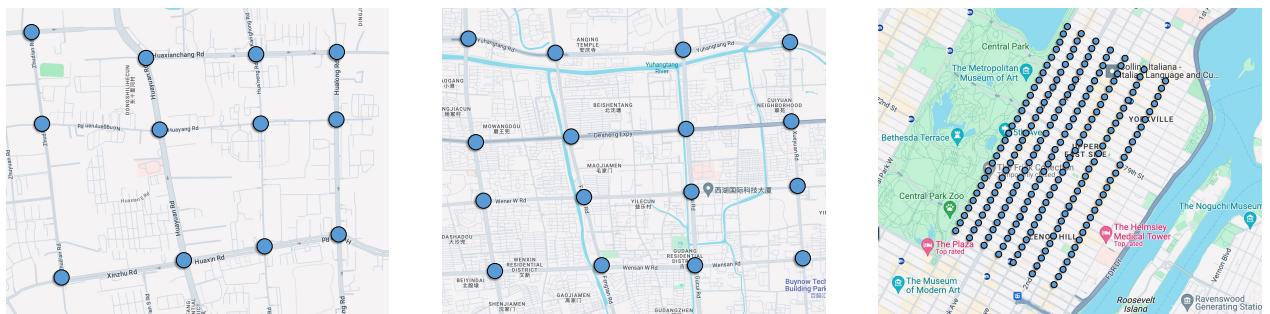


Figure 18: The road network systems of the Jinan, Hangzhou, and New York datasets.

**Table 11: The prompt template.**


---

A traffic light regulates a four-section intersection with northern, southern, eastern, and western sections, each containing two lanes: one for through traffic and one for left-turns. Each lane is further divided into three segments. Segment 1 is the closest to the intersection. Segment 2 is in the middle. Segment 3 is the farthest. In a lane, there may be early queued vehicles and approaching vehicles traveling in different segments. Early queued vehicles have arrived at the intersection and await passage permission. Approaching vehicles will arrive at the intersection in the future.

The traffic light has 4 signal phases. Each signal relieves vehicles' flow in the group of two specific lanes. The state of the intersection is listed below. It describes:

- The group of lanes relieving vehicles' flow under each traffic light phase.
- The number of early queued vehicles of the allowed lanes of each signal.
- The number of approaching vehicles in different segments of the allowed lanes of each signal.

Signal: ETWT

Allowed lanes: Eastern and western through lanes

- Early queued: 1 (East), 4 (West), 5 (Total)
- Segment 1: 0 (East), 0 (West), 0 (Total)
- Segment 2: 0 (East), 2 (West), 2 (Total)
- Segment 3: 2 (East), 1 (West), 3 (Total)

Signal: NTST

Allowed lanes: Northern and southern through lanes

- Early queued: 2 (North), 0 (South), 2 (Total)
- Segment 1: 1 (North), 0 (South), 1 (Total)
- Segment 2: 1 (North), 0 (South), 1 (Total)
- Segment 3: 4 (North), 1 (South), 5 (Total)

Signal: ELWL

Allowed lanes: Eastern and western left-turn lanes

- Early queued: 0 (East), 0 (West), 0 (Total)
- Segment 1: 0 (East), 0 (West), 0 (Total)
- Segment 2: 0 (East), 0 (West), 0 (Total)
- Segment 3: 0 (East), 1 (West), 1 (Total)

Signal: NLSL

Allowed lanes: Northern and southern left-turn lanes

- Early queued: 4 (North), 3 (South), 7 (Total)
- Segment 1: 0 (North), 0 (South), 0 (Total)
- Segment 2: 0 (North), 0 (South), 0 (Total)
- Segment 3: 1 (North), 2 (South), 3 (Total)

Please answer:

Which is the most effective traffic signal that will most significantly improve the traffic condition during the next phase?

Note:

The traffic congestion is primarily dictated by the early queued vehicles, with the MOST significant impact. You MUST pay the MOST attention to lanes with long queue lengths. It is NOT URGENT to consider vehicles in distant segments since they are unlikely to reach the intersection soon.

Requirements:

- Let's think step by step.
  - You can only choose one of the signals listed above.
  - You must follow the following steps to provide your analysis: Step 1: Provide your analysis for identifying the optimal traffic signal. Step 2: Answer your chosen signal.
  - Your choice can only be given after finishing the analysis.
  - Your choice must be identified by the tag: <signal>YOUR\_CHOICE</signal>.
-