Incorporating vision-based artificial intelligence and large language model for smart traffic light control

Yao, Jiarong; Li, Jiangpeng; Xu, Xiaoyu; Tan, Chaopeng; Yap, Kim Hui; Su, Rong

DOI
10.1016/j.asoc.2025.113333

Publication date
2025

Document Version
Final published version

Published in
Applied Soft Computing

Citation (APA)
Yao, J., Li, J., Xu, X., Tan, C., Yap, K. H., & Su, R. (2025). Incorporating vision-based artificial intelligence and large language model for smart traffic light control. *Applied Soft Computing*, *179*, Article 113333. https://doi.org/10.1016/j.asoc.2025.113333

Applied Soft Computing Journal 179 (2025) 113333

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Incorporating vision-based artificial intelligence and large language model for smart traffic light control

Jiarong Yao [a], Jiangpeng Li [a,*], Xiaoyu Xu [a], Chaopeng Tan [b], Kim Hui Yap [a], Rong Su [a,*]

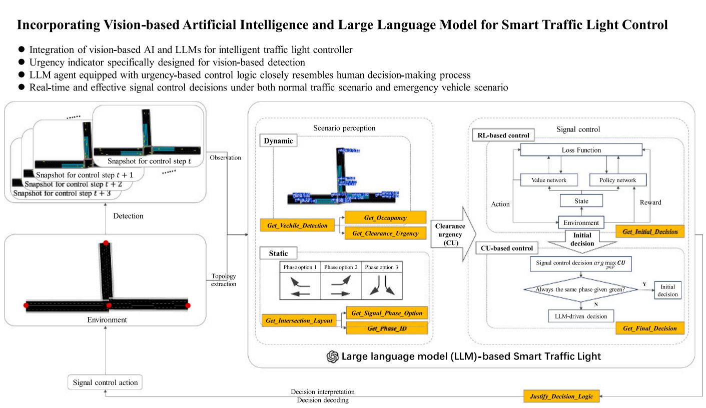
[a] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

[b] Department of Transport and Planning, Delft University of Technology, Gebouw 23, Stevinweg 1, Delft 2628 CN, the Netherlands

HIGHLIGHTS

- This is the first exploration to incorporate vision-based perception and AI techniques into large language model-based traffic light control.
- A vision-based perception tool is designed to extract real-time image features of an intersection from a top-down view using YOLO algorithm.
- An LLM agent is augmented with a toolkit incorporating data-driven insights, domain knowledge and logical reasoning akin to human awareness.
- A clearance urgency index is defined to quantify the need for green time regarding phase-level traffic demands and vehicle progression priority.
- An integrated control framework is proposed aiming at better flexibility of signal control optimization under diverse scenarios.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Adaptive Traffic Signal Control
Computer Vision
Large Language Models
YOLO Object Detection
Reinforcement Learning

ABSTRACT

The increasingly complicated urban traffic patterns lead traffic signal control to a new trend of higher flexibility and quicker response, which becomes possible with advances in both sensor technology and artificial intelligence. Though in its early stage, existing intelligent signal controllers equipped with reinforcement learning (RL)-based feature extractor and large language model (LLM)-driven scenario understanding and decision support already demonstrate powerful data digesting ability. This study thus proposes a smart traffic light control system integrating a vision-based perception tool to extract traffic state from real-time snapshot image of the intersection, and an LLM agent controller for signal phase switching upon scenario analysis. An indicator describing the urgency for green time at phase level is defined to abstract the contextual information regarding the competition of multiple approaching traffic flows, which augments the LLM with domain-specific logical reasoning for signal control action generation, aimed at assigning green time to the flows with the most compelling needs. With a RL-based controller providing initial control decision as backup, the proposed method

* Corresponding authors.
E-mail addresses: jiarong.yao@ntu.edu.sg (J. Yao), jiangpen001@e.ntu.edu.sg (J. Li), xuxi0019@e.ntu.edu.sg (X. Xu), c.tan-2@tudelft.nl (C. Tan), ekhyap@ntu.edu.sg (K.H. Yap), rsu@ntu.edu.sg (R. Su).

https://doi.org/10.1016/j.asoc.2025.113333
Received 1 January 2025; Received in revised form 25 April 2025; Accepted 14 May 2025
Available online 22 May 2025

is able to handle both pre-trained and out-of-distribution scenarios through real-time traffic state diagnosis and knowledgeable reasoning. Simulation evaluation on different intersection layouts and vehicle compositions is conducted with horizontal comparison of five benchmarks. A decrease in average waiting time was realized by more than 5 % under normal traffic scenario and 20 % under emergency vehicle scenario, respectively. Further, comprehensive analysis was conducted to explore the applicability of the proposed method and feasibility for real-world application in unmanned aerial vehicle (UAV)-based intelligent traffic management.

1. Introduction

As one of the most important means of traffic management, traffic signal control is in essence aimed at the optimal matching between temporal right-of-way supply and incoming demands of different movements within the scale of the intersection. The evolution of signal control methods from fixed time, actuated to adaptive mode is not only attributed to the advance of detection techniques, but also reflects the needs for a more flexible and responsive solution. Recent decades have witnessed the burgeoning of learning-based signal control methodology, which leverages artificial intelligence (AI) techniques like deep learning (DL), reinforcement learning (RL) and deep reinforcement learning (DRL) in either state perception or policy generation to support decision making [1–3].

The application of learning-based algorithms in traffic signal control actually presents a paradigm shift in both control strategy and methodology as compared with the traditional rule-based or model-based methods [4]. Deep learning algorithms are widely used for traffic state estimation and prediction, which serve as input for signal control optimization [5–8]. RL-based methods generally adopt a Markov decision process framework to model the signal control problem as a goal-oriented and sequential decision-making problem based on environment interaction to cope with the uncertainty and complexity of traffic flow [9]. By generating action distributions directly through a policy network, rewards based on environmental feedback will guide the directional exploration, deriving adaptive traffic signal controllers considering the nonlinear and complicated relationship between traffic demand and infrastructure supply [10]. However, pure RL is still not well resolved in the face of a high-dimensional solution space, which gives rise to the combination of RL and DL. DRL shows effectiveness in solving the dimensional catastrophe problem which traditional RL encounters in high-dimensional state spaces [11–13]. Used for adaptive traffic signal control with varying degrees of success, DRL-based methods are mostly aimed at single intersection scenario, while the studies on large-scale network application show a diversified trend, including integrating cooperative control strategies at the global level by regarding the network as a single agent [14–16], and designing multi-agent reinforcement learning (MARL) framework with elaborated agent autonomy and interaction modeling [17–19].

Despite satisfactory performance in control performance and data processing, learning-based methods are faced with inevitable shortcomings like lack of generality or dependence on pre-training, which necessitates the evolvement of signal controllers featuring more human-like awareness reacting to the volatility of urban traffic demand scenarios. Large language models (LLMs), such as the generative pre-trained Transformer (GPT) series [20], bidirectional encoder representations from transformers (BERT) [21], and text-to-text Transfer transformer (t5) [22], have thus come to light and gained significant attention due to their ability to learn from application behavior and assist in the optimization of existing systems [23,24]. Starting to infiltrate various applications in intelligent transportation system (ITS), including traffic prediction, sentiment analysis of social media data for traffic insights, emergency response, disaster management, and multi-modal transportation planning, LLMs are also well regarded in terms of signal control, with a promise of flexibly transferring engineering experience or expert knowledge in various scenarios some of which we have even never seen [25,26].

Thus, this study is an innovative attempt to propose a LLM-based signal controller by integrating the emerging UAV detection input, the AI techniques for traffic state evaluation corresponding to the image processing of data source, and the AI-driven control framework using LLM for signal control optimization.

The rest of this paper is organized as follows. Section 2 gives a detailed review on the current LLM-based signal control methods and vision-based AI techniques in ITS application. The methodology section presents the integrated control framework of the smart traffic light, with details on prompt construction of related sub-tasks for signal control optimization and the corresponding tools, especially the vision-based perception and the derivative urgency indicator as well as the control method. To evaluate the proposed method, a simulation case study with comprehensive horizontal comparison and result analysis is given in Section 4. Conclusions and discussions of this paper are provided in the last section with future interest of relevant issues.

2. Literature review

2.1. Large language model (LLM)-based traffic signal control

LLMs' excellence at understanding, generating, and translating human-like text, which is in line with the call for smarter signal controllers, spawns a series of applications or implementations regarding urban signal control [27]. Tang et al. directly used LLM as the bridge between computers and human engineers for linguistic interaction and strategy generation, where LLM played a role of scenario interpretation and recommendation of corresponding control strategies, while the mapping between scenarios and control strategy was predefined and updated manually [28]. Similarly, Dai et al. established a digital traffic engineer framework equipped with LLM for interpretation of human language and the automation of traffic control strategy generation. Through chain-of-thought (CoT)-driven prompts, LLMs took over some human-dominated tasks like scenario parsing, control method selection, control execution and report generation [29]. A more comprehensive and integrated framework, Open-TI (traffic intelligence) was proposed by Da et al., involving traffic situation extraction, origin-destination demand estimation, traffic flow prediction and signal control, which is stated as the first attempt of using LLM to conduct exhaustive traffic analysis from scratch [30]. Specifically, in [30], LLM was augmented with a tool pool consisting of multiple external traffic analysis packages to support various task-specific embodiments, ranging from downloading map to running simulation. Among them, control-related tasks included not only selection of given control models but also training and fine-tuning of control policies through an agent-agent communication, realizing meta-control featuring open-ended design with great scalability.

Different from the above research where LLM serves more as a subsidiary, a larger canvas for LLM's capacity in signal control optimization is further explored. The LLMLight proposed by Lai et al. was the first exploration to adopt LLM directly as a decision-making agent, which equipped the LLM with knowledgeable prompts detailing real-time traffic conditions and reasoning the optimization requirements akin to human intuition. A specialized backbone LLM, LightGPT is designed tailored for signal control tasks using imitation learning for critic-guided control policy fine-tuning [31]. Starting from Open-TI, Da et al. continued focusing on the incorporation of LLM in signal control task. A prompt-based grounded action transformation (GAT) framework to fine tune control policies of RL-based signal control methods is further proposed under out-of-distribution scenarios [32]. With real-time data, LLM provided valuable insights about the weather, road condition for prompt learning combining implicit human knowledge, so that the RL-based control models learned from simulation dataset can better adapt to complicated practical scenarios. Pang et al. and Wang et al. integrated RL and LLM for human-mimetic signal control specially for unfamiliar or uncommon scenarios like emergency vehicles, detection failure, or link blockage caused by traffic incidents, etc. [33,34]. Both studies adopted a RL-based method for initial signal control action generation while the difference lies in that the former method, iLLM-TSC, which is short for integrated large language model-based traffic signal controller, built an assessment task based on LLM to evaluate the reasonableness of the initial decision based on real-time data and provide the final decision through policy refinement realized through delicate prompts designed by chain-of-thought and tree-of-thought techniques [33]. The latter, LLM-Assisted Light, on the other hand, focused on LLM augmentation through a series of perception and decision tools with a more comprehensive human-like judgement logic to recommend the most suitable tools considering their causal relationships, which demonstrated a trend of toolkit modularization to make RL-based controller less dependent on reward function [34]. Movahedi and Choi proposed a novel LLM-based agent development framework including two types of Generally Capable Agents (GCAs): the actor agent and the critic agent, for adaptive traffic signal control considering the similarities of LLM-based controllers to RL agents in learning from past experiences to optimize outcomes through continuous interaction with their environments. A zero-shot chain of thought method solely using actor agent, made real-time decisions like switching to new traffic phases or extending the current ones by leveraging its extensive knowledge base and advanced reasoning and planning capabilities, resembling an open-loop feedback control system. Another dual-agent controller using both actor agent and critic agent, operated more like a closed-loop feedback control system where evaluation on the action outcomes was further conducted to integrates new knowledge from environmental interactions and enhance the system's overall learning performance [35,36].

It is noted that such smart signal controllers equipped with LLMs all target at the control parameter optimization of isolated intersections with adaptive signal control mode, which mostly determines the next phase or controlled movements to assign green time. Studies on extending LLM-integrated single intersection control to the control of an intersection cluster, arterial or even network coordination control are still in their infancy. Based on Lai's LLMLight, Tislenko and Kisilev regarded each intersection in the network as an individual decision-making agent featuring integration of RL and LLM, meanwhile considering the heterogeneity of different agents in terms of their environments and states [36]. The increase in the number of control objects actually motivated the aggregation of diverse decision recommendations for a larger action space, and the optimal action was determined using a majority voting mechanism, with an eventual goal for global system performance improvement whose degree is related to task difficulty. By contrast, Tang et al. proposed an LLM-assisted green wave control approach tailored for arterial coordination control. Domain-specific tasks like network parsing, knowledge integration and green wave evaluation were set up as the component of the LLM workflow, and the final control policy was given to optimize the offset for fixed-time coordination in an interactively way with detailed interpretation [37].

Though the infiltration of LLM in signal control is not so widespread as that in autonomous driving [38–41], the existing studies have already showed good signs of its potential with different degrees of participation. Despite such promising development, limitations and challenges remain, among which one direction is to improve real-time data processing and feedback capability as well as accuracy especially when multi-step action embodiment is designed. It thus provokes a thought of how to guarantee the alignment between the expected and actual output after several intermediate sub-tasks concerning feature extraction and interpretation. From this perspective, incorporating vision-based models capable of directly processing visual information of the traffic state is expected to perform better in scenario perception than that based on non-vision-based sensor data.

2.2. State-of-the-art vision-based AI implementation in traffic signal control

Recent breakthroughs on computer vision and pattern recognition actually pave the way for such idea, which can be exemplified by the emerging topic of combination of UAV images and computer vision techniques in urban traffic management [42,43]. Unlike loop detector or LIDAR sensor data, image or video data provide the information of a larger array of partakers in a traffic scenario, which is also closer to the recognition channels of traffic engineers [44]. Considering the signal control domain-specific knowledge that green time should be given to the movement with more incoming demand or higher urgency to be served, signal control using vision input either from simulation or field application has carved out a place since the past decade. Jeon et al. were the first to adopt an image-based RL algorithm for traffic signal control, where aerial images or animation shoots were used to derive the waiting and approaching vehicles to represent state and reward during reinforcement learning [45]. In [46], traffic simulator snapshots were input into a deep convolutional network to estimate velocity and travel direction of vehicles, which served as state presentation for both policy- and value-based RL methods to optimize the phase switching of isolated intersections. Using cameras to monitor real-time traffic information, Willy et al. proposed an adaptive phase-free distributed algorithm using the conflict matrix of different movements, considering the balance between the queue with heavy traffic and low traffic to increase the green time utilization [47]. Different from the top-down perspective of simulation snapshots, surveillance cameras or license plate recognition detectors providing entry-based vehicle arrival from a squint angle are mostly commonly used in practice. According to different utilization degrees of video detection, methodologies span from using AI-based algorithms like YOLO (You Only Look Once) series to extract traffic flow indexes like flow or density to determine the phase switching or green phase duration [48–52], to adopting DRL for near-real-time signal control in more diverse scenarios through a fully autonomous and end-to-end pattern [53,54]. Though research using drone-based image data as input are only beginning to emerge, Aatmaj has integrated drone-captured images as a data source in Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) machine learning architectures for more feature extraction and more accurate prediction of traffic flow patterns in enhancing urban traffic management through intelligent traffic light control, presenting potential of drone detection to revolutionize traffic management systems [55].

It can be concluded from the reviewed literature that vision input with a top-down bird-eye perspective at the intersection does record richer information about the dynamics of both individual participants and the interaction, either cooperative or competitive, between different control objects. However, the information usage for signal control optimization is mostly confined to detecting, classifying, vehicle counting and tracking which can also be obtained by fixed detectors or connected vehicle trajectory data, making vision input seemingly just a low-cost alternative detection means, leaving other valuable information unrevealed [56,57]. Motivated by such underutilization of vision input in signal control and the above-mentioned trend of adopting vision-based human-mimetic understanding and reasoning in LLM-based signal controllers, this study proposes to incorporate vision-based AI techniques and LLM into a signal control system to explore higher intelligence degree of smart traffic lights based on the existing LLM-driven signal control methods [31,33–34]. By processing real-time images of the intersection with a top-down perspective using

Fig. 1. A sample of Signalized INtersection Dataset (SIND) [58]. (Green label refers to the car class, light green label refers to the pedestrian class, yellow label refers to the motorcycle class, green label refers to the bicycle class, red label refers to the specific object used in [58]).

the YOLO algorithm, an urgency indicator is defined to quantify the need for green time regarding the incoming demands of different phases, which is then developed as a perception tool. Together with other perception tools like layout extraction and phase structure parsing, a signal control method based on the urgency indicator is developed as the decision-making tool of an LLM agent through domain-specific prompt design to realize adaptive signal control optimization of an isolated intersection. The proposed GPT-assisted signal controller is expected to simplify the process for human operators by providing clearer insights into what is observed and which phase should be immediately given green time as compared with current LLM-based controller using non-vision-based detection data. Even under uncommon scenarios like emergency vehicles which are seldom used for pre-training for RL-based control methods, informed decisions ensuring safety and special vehicle preemption can be obtained.

2.3. Contribution of this study

Positioned in the fields of computer vision, artificial intelligence and traffic signal control, this work is an interdisciplinary exploration attempting to improve the flexibility of signal control to handle the priorities of different passing demands under diverse scenarios, which can be summarized into the following key contributions.

- With a comprehensive review on the state-of-the-art LLM-based signal control methods, this work explores potential of integrating vision-based AI and LLMs to enhance the perception and decision-making processes of an intelligent traffic light controller, especially in dynamic or complex environments requiring fast decisions. Comprehensive experimentation showed the remarkable learning and adaptation capabilities of the proposed controller within deployed environments, with great prospect to provide automated decisions in real-world settings.
- Unlike the delay-related measure adopted in the RL-based model for initial signal control action determination, the proposed urgency indicator is more intuitive and informative, specifically tailored for vision-based detection. Equipped with such urgency-based control logic for the determination of the next phase to be switched to, the LLM closely resembles human decision-making processes, amalgamating domain knowledge and data-driven insights for a robust predictive control method capable of capturing intricate traffic dynamics.
- With customized prompts connecting the vision-based traffic state representation with the LLM, real-time decisions can be made about when to capture video snapshots or images for perception in every control step and what areas to focus on based on the scene understanding capability of LLM, which is less resource-consuming than building an integrated DRL-based controller consisting of a perception system, a separate control system, and other adaptive and pre-processing tools for pre-training and fine-tuning.

3. Methodology

Aimed at an isolated intersection, the research problem is to determine which phase should be operated in the next control step considering an adaptive signal control mode using real-time image data as input, with the goal of improving the progression efficiency of the traffic flows to be served. The input images are collected from a top-down perspective to have a panoramic view of the intersection and the

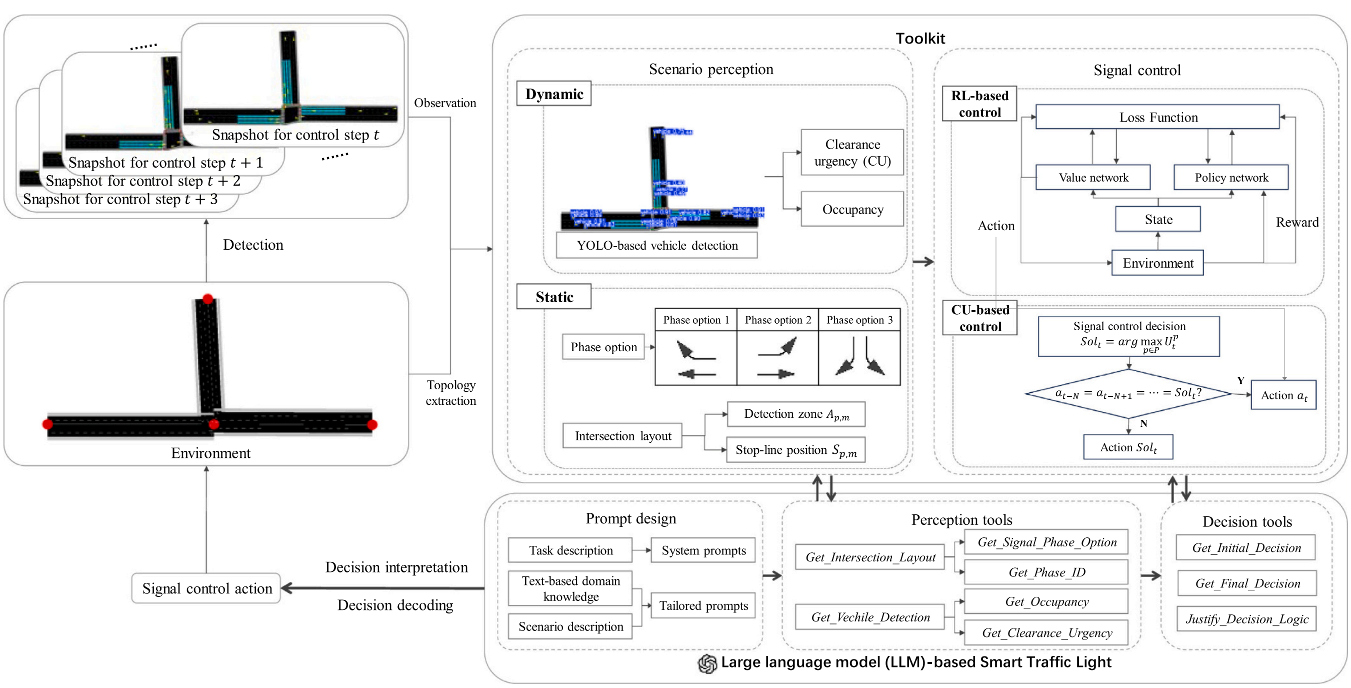


Fig. 2. The unified architecture of the LLM-based smart traffic light.

Fig. 3. A sketch of the structured prompts for the LLM-based smart traffic light.

vehicles running in its entries and exits, which is feasible considering the wide application of UAV detection and rich datasets in monitoring and surveillance tasks in the field of smart cities, as exemplified by a dataset sample shown in Fig. 1.

The core idea of the proposed method is to integrate vision-based AI and LLM into one unified control framework, including a vision-based perception tool to extract real-time image features using YOLO series algorithm, and an LLM agent augmented with a toolkit consisting of both perception and decision support tools incorporating domain knowledge of signal control and the logical reasoning as well as generalization capabilities of LLM.

As shown in the architecture in Fig. 2, the LLM agent is customized with both general system prompts and domain-specific prompts for the specific signal control task, enabling a scalable function set involving scenario understanding, information extraction and signal control recommendation as well as interpretation. Specifically, the proposed vision-based perception is realized by capturing images to a pre-trained YOLO model for vehicle detection, whose outputs include vehicle type,

Table 1
Symbol definition.

Indices and sets	
i	Index of vehicles
j	Index of vertexes
p	Index of phases
m	Index of movements
P	Set of feasible phases
M_p	Set of movements of phase p
$D_{p,m}$	Set of vehicles in movement m of phase p
Variable	
$X_{i,t}^{p,m}$	Property vector of identified vehicle i in movement m of phase p at control step t
$A_{p,m}$	Vertex vector of area of movement m of phase p
$S_{p,m}$	Vertex vector of the stop-line of movement m of phase p
$L_{p,m}$	Length of the lane group of movement m of phase p in the snapshot
$lane_{p,m}$	Number of lanes of the lane group of movement m of phase p
$O_t^{p,m}$	Occupancy of the lane group of movement m of phase p at control step t
$Q_t^{p,m}$	Vehicle count of the lane group of movement m of phase p at control step t
U_t^p	Clearance urgency of phase p at control step t
x_j, y_j	Coordinate of vertex j
v_j	Area vertex point j
c_i	Identified class of vehicle i
$conf_i$	Confidence of object detection for vehicle i
w_i	Weight of vehicle i
$d_{i,t}$	Distance between vehicle i and the stop-line of the entry it is approaching
l_i	Average vehicle length of vehicle i
a_t	Action of RL agent at control step t
s_t	Environmental state of RL agent at control step t
θ, ϕ	Network parameter of RL agent
π_θ	Policy network
v_ϕ	Value network
$F()$	Loss function for RL agent training
$\mathscr{L}_p, \mathscr{L}_v$	Loss term of policy function and value function
λ	Hyperparameter used to balance the policy loss and value loss
N	The number of control steps of the maximal green time
Sol_t	The final signal control solution at control step t
Parameters	
l_v	Average vehicle length of the general vehicles
l_e	Average vehicle length of the emergency vehicles

phase-wise incoming flow and vehicle location. Such extracted traffic state indexes are then used as input to obtain signal control solution through two methods, one acting as the initial control action using a universal RL-based control method and the other serving real-time signal control using an urgency-based control logic by defining a clearance urgency (CU) indicator to quantify the urgency of the phase-level traffic flow for the green time. With the final decision, the signal control solution is implemented with comprehensive decision interpretation through the response of LLM agent, regarding whether the current phase should be continued or switched to a different one for the coming control step, where the traffic evolution at the last timepoint will then become the perception object of the next control step.

Acting as a bridge to translate user intentions into specific tasks, prompts play a pivotal role in shaping the outputs of LLMs. The formulation of a prompt can significantly influence the style, length, and format of the system's response, directing the capabilities of LLMs to meet specific user needs and objectives. In order to properly utilize LLM features, we created prompts utilizing chain-of-thoughts method, enabling LLM to undertake a situational analysis and give appropriate advice. For the proposed signal control task, the prompts were structured to have details on traffic situation, vehicle information and control objective through a series of queries which are passed to the LLM agent, thus generating context-specific information and making decisions with human-like judgment logic, as shown in an example of the prompt structure in Fig. 3. Five components are included, (i) task description, defining the role of LLM as a smart traffic light; (ii) tools synopsis, listing the toolkit for perception and decision; (iii) attention points, outlining the guidelines for tool implementation regarding standards or constraints of signal control parameters; (iv) format declaration, specifying the output format of LLM; and (v) logical chain of dialogue, assisting

Table 2
Prompt components for static perception tools.

Tool	*Get_Intersection_Layout*	*Get_Signal_Phase_Option*
Function	This tool delineates the intersection's configuration, detailing the number and function of lanes associated with each direction.	This tool offers a detailed description of the traffic signal phases at the intersection, outlining the associated traffic movements for each phase.
Input	Intersection index	Intersection index
Output	A dictionary where each key represents a traffic movement index at the intersection, indicating the entry index and the direction index, where 's' is for straight, 'l' is for left-turn, and 'r' is for right-turn. The corresponding value is another dictionary with the following keys: "direction": A string indicating the lane direction, including 'Through', 'Left-turn' and 'Right turn'. "number of lanes": An integer representing the number of lanes for the specified direction.	A dictionary where each key represents a signal phase index in the given signal phase option set. The corresponding value is another dictionary with the following keys: "movements": A list of the movements controlled by the phase of interest.
Example	Call the function as follows: Layout = *Get_Intersection_Layout ('J1')* Output: Layout = {"-E1-s":{"direction": 'Through', 'number of lanes':2}, ...}	Call the function as follows: Phase_set = *Get_Signal_Phase_Option('J1')* Output: Phase_set = {"Phase 0": {"movements":['-E1-s', 'E0-s', '-E1-r', 'E0-r']},......}

LLM in reasoning according to a human-like logic of scenario perception, traffic state evaluation as well as analysis, and signal control optimization.

In the following sub-sections, detailed introductions of the proposed control method will be elaborated.

3.1. Preliminaries

The proposed method is based on the following assumptions.

a) An on-road camera is used to capture the road condition of the intersection. The camera angle should be wide enough to cover all the connected roads at the intersection. Additionally, the camera is considered to remain in an almost fixed position above the intersection, with relatively low amounts of shaking.
b) The input images are assumed to be available already, and their means of capture is outside the scope of the research.
c) Traffic demands are undersaturated.
d) The available phase set is given as known.

The symbols used in the following text are declared in Table 1.

3.2. Perception featuring vision-based artificial intelligence

Although data gathered in the real world will provide precise signals about the dynamics of the traffic environment, it may suffer from lack of visual diversity as it is costly to gather comprehensive data covering various junction configurations and demand scenarios in the real world. Thus, simulation is used as a safe, cost-effective and controlled tool for dataset establishment and method validation. In this study, snapshots of the intersection through the graphical user interface (GUI) of SUMO simulator are used as an alternative of putting a camera or UAV over an intersection to collect the image data of the whole intersection. A pre-trained YOLO v11 model is then used for re-training using the desired dataset consisting of the extracted snapshots, in order to adapt to certain classes in the target intersection scenario while containing the features

Table 3
Prompt components for dynamic perception tools.

Tool	*Get_Phase_ID*	*Get_Vehicle_Detection*
Function	This tool identifies the currently active traffic signal phase at the intersection, which is crucial for the LLM to understand the traffic flows of which movements are being allowed to pass.	This tool identifies the information of the approaching vehicles in each movement from the snapshot captured at the current timepoint.
Input	Intersection index	Intersection index
Output	An integer representing the phase index under operation for the specified intersection.	A dictionary where each key represents a traffic movement index at the intersection. The corresponding value is another dictionary with the following keys: "vehicle count": An integer representing the number of identified vehicles in the specified movement. "vehicle information": A dictionary where each key represents the vehicle index, the corresponding value is another dictionary with the following keys: "vehicle type": indicating the type of vehicles, including 'general vehicle' and 'emergency vehicle'. "vehicle location": A tuple including two floating numbers with four decimal places representing the 2-dimensional coordinate of the vehicle centroid. "confindence": A floating number with two decimal places indicating the identification confidence of the specified vehicle
Example	Call the function as follows: Phase_id = *Get_Phase_ID ('J1')* Output: Phase_id = 1	Call the function as follows: Vehicle_set = *Get_Vehicle_Detection ('J1')* Output: Vehicle_set= {"-E1-s":{'vehicle count': 4, 'vehicle information': {'-E1_ego.0':{'vehicle type': 'general vehicle', 'vehicle location': (0.0436, 0.8609)}}},……}

learned from the source dataset [59]. Thus, detection of different types of vehicles is realized by YOLO v11 through customized re-training and parameter fine-tuning, which can be processed into more quantitative indicators in the form of a sensory tool, enabling LLM to provide accurate traffic state perception, together with other cognitive tools.

3.2.1. Intersection scenario perception

Through domain-specific prompts, two perception tools, *Get_Intersection_Layout* and *Get_Signal_Phase_Option* are designed, as shown in Table 2, aiming to enable LLM to collect the environmental properties of the studied intersection.

3.2.2. YOLO-based vehicle objection

YOLO is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility. As a single regression based on a convolutional neural network, YOLO is an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks [60]. Released in September 2024, YOLO version 11, short for YOLOv11, brings improvements in the feature extractor using more convolutions with a modified Cross Stage Partial DenseNet backbone containing enhanced attention mechanisms and data augmentation techniques, which is proved to be more efficient with lower reported latency and computational complexity [61]. Thus, this section

Table 4
Prompt components for traffic operation perception tools.

Tool	*Get_Current_Occupancy*	*Get_Clearance_Urgency*
Function	This tool calculates the spatial occupancy of the lane group in each movement based on the vehicle detection results.	This tool calculates the clearance urgency indicator of the movements in each phase based on the vehicle detection results.
Input	Intersection index	Intersection index
Output	A dictionary where each key represents a traffic movement index at the intersection. The corresponding value is the spatial occupancy in the form of percentages.	A dictionary where each key represents a phase index at the intersection. The corresponding value is the clearance urgency in the form of a floating number.
Example	Call the function as follows: Occupancy = *Get_Current_Occupancy ('J1')* Output: Occupancy = {"-E1-s":31.52 %, "-E1-l":9.00 %, ……}	Call the function as follows: Urgency = *Get_ Clearance_Urgency ('J1')* Output: Urgency = {"Phase 0": 0.002516666323997673, "Phase 1": 0.0033701980933799774, ……}

introduces the YOLOv11 pre-trained using the COCO dataset [83] and fine-tuned using the custom dataset of the studied intersection to detect the approaching vehicle at signalized intersections for control purposes.

With the RL-based control method embedded for initial signal timing decision, the training process can also serve as the establishment of the custom dataset to fine tune YOLO v11. For the decision making at every control step, snapshot is captured in sync with the state extraction of RL, thus forming a custom dataset. For the signal control tasks aimed at normal traffic scenario and emergency vehicle scenarios in this study, the YOLO model is trained to detect two types of vehicles: cars and emergency vehicles. The custom dataset is split into a training set, validation set, and test set at an 18:1:1 ratio, with each image labelled with the vehicle type. With the fine-tuned YOLO model, the real-time image is fed as input for vehicle detection, and the output are the vehicles identified with bounding box and a confidence score on the image, which are represented using Eq. (1). In order to further determine to which phase an identified vehicle belongs within the intersection layout, the area of lane groups for each phase, $A_{p,m}$ and the stop-lines of each entry, $S_{p,m}$ are calibrated based on the centroid coordinates of the identified vehicles, as given by Eqs. (2) - (3).

$$X_{i,t}^{p,m} = (c_i, \quad conf_i, \quad x_i, y_i)_t \tag{1}$$

$$A_{p,m} = (v_1, \quad v_2, \quad v_3, \quad v_4), \quad v_j = \left(x_j, y_j\right), \quad j \in \{1,2,3,4\} \tag{2}$$

$$S_{p,m} = (v_1, \quad v_2), \quad v_j = \left(x_j, y_j\right), \quad j \in \{1,2\} \tag{3}$$

Thus, perception of the traffic dynamics at the intersection is realized through two tools, *Get_Phase_ID* and *Get_Vehicle_Detection*, whose prompt design is shown in Table 3.

3.2.3. Dynamic traffic operation evaluation

By vehicle detection, the traffic demand to be served can be obtained as the number of vehicles in different movements, which can be used to calculate the occupancy index used in the state representation of RL-based control method as the alternative of non-vision-based detection data, as given by Eqs. (4) - (5). The prompt detail of the corresponding tool, *Get_Current_Occupancy*, is shown in Table 4.

$$O_t^{p,m} = \frac{\sum_{i=1}^{Q_t^{p,m}} l_i}{L_{p,m} \times lane_{p,m}} \tag{4}$$

$$l_i = \begin{cases} l_v, & c_i = general\ vehicle \\ l_e, & c_i = emergency\ vehicle \end{cases} \tag{5}$$

However, for phases with similarly sized demands, non-vision-based detectors fail to provide further reference to distinguish their priorities. Additionally, the dependence of occupancy index on the link length may be affected by the heterogeneous topology of different links, which may sometimes mislead the signal control decision. By contrast, such issue is actually easy to handle for a human engineer who would assign the green time to the phase with more vehicle queuing or closer to the stop-line intuitively from the image of the intersection. Under emergency vehicle scenarios, traffic engineers will also offer signal preemption to emergency vehicles by default considering that emergency vehicles enjoy far higher urgency than other vehicle types. Such practitioners' control strategies under these two scenarios actually can be integrated in a unified control logic with the image data input.

Thus, leveraging the rich information carried by the real-time snapshot, a new indicator, **clearance urgency**, is defined as shown in Eq. (6). The subscript t of U_t^p refers to the control step t, while the superscript p refers to the phase index, thus the clearance urgency index describes the urgency for the green time of phase p at control step t. Defined as weighted sum of the reciprocals of the distance between the vehicles and the stop-line considering all the vehicles (indexed by i) controlled by a specific phase which consists of several controlled movement (indexed by m), this index provides human-mimetic viewpoint to the consequent reasoning and decision-making of LLM. The weight is a function of the vehicle class, $w_i = f(c_i)$, reflecting the contribution in the phase-level urgency for the green time. For example, the weight of emergency vehicle is far larger than the passenger car. Based on this indicator, an assessment tool, *Get_Clearance_Urgency*, is designed to comprehensively quantify the demand for right-of-way of all phases regarding volume, spatial distribution and preemption of incoming vehicles, with details of prompt design shown in Table 4.

$$U_t^p = \sum\nolimits_{m=1}^{M_p} \sum\nolimits_{i=1}^{D_{p,m}} \frac{w_i}{d_{i,t}} \tag{6}$$

The clearance urgency indicator not only considers the absolute number of incoming vehicles, but also quantifies their spatial distribution and service priority simulating human-like awareness. This indicator enjoys better applicability in adaptive signal control than queue length which is often used in the reward or objective function in RL-based signal control, as the control step is usually a short-term period not enough for queue forming, which may to some degree limit the space of optimization. Moreover, vehicle detection obtained from a top-down view feasible in UAV detection actually guarantees the comparability of the clearance urgency indicator which is cumbersome to realize through data fusion of multi-view license plate recognition (LPR) video detection.

3.3. LLM-driven reasoning for signal control optimization

The signal control of the intersection consists of two parts, the initial signal control action based on environmental observations by modelling the intersection as a RL agent, and the final signal control decision after thoughtful reasoning of the LLM agent given the real-time urgency-based operation evaluation.

3.3.1. RL-based control for initial signal control decision

The control of traffic lights at an intersection is modelled as a Markov Decision Process (MDP) and a RL agent is developed for the intersection to act as an intelligent controller to execute signal control action and interact with the environment, with the expectation to maximize the reward. An existing RL-based control method is adopted here for the initial signal control decision making [62]. Three critical components, state, action and reward are defined to describe the RL-based signal control process of each control step.

- The state of each movement is represented by the occupancy of the last time step, namely the ratio between the total length of lane groups occupied by vehicles of the certain movement and the total length of lane groups, at time t, as denoted by s_t^m.
- The action a_t of the RL agent refers to selecting a phase from the given phase set to be executed in the next control step.
- The reward of the RL agent is the average waiting time of vehicles.

Based on the above RL agent design, the adaptive signal control of the intersection is realized by executing the signal phase switching between any two consecutive control steps according to the policy of the agent, in an expectation to reduce the average waiting time and ultimately, clear the intersection as soon as possible.

Training of the RL agent is based on the Proximal Policy Optimization (PPO) algorithm and adopts two neural network, one policy network and one value network, both using state s_t^m as input [63], as shown in Fig. 2. By minimizing a loss function incorporating both the policy loss and the value loss, the value network learns to predict the expected return for each state accurately, enabling the policy network to select better actions for the agent accordingly.

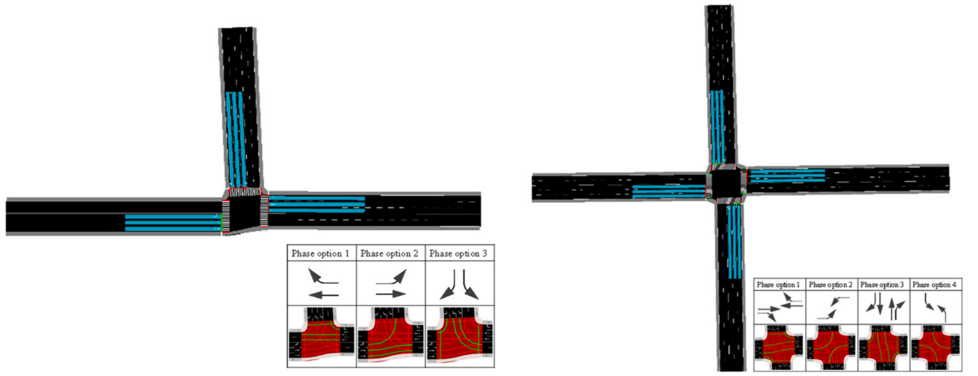
Using the RL-based controller, the initial signal control decision can be provided as one decision tool of LLM agent, *Get_Initial_Decision*, which is detailed in Table 5.

Table 5
Prompt components for initial signal control decision tools.

Tool	*Get_Initial_Decision*
Function	This tool is designed to use the RL-based controller to obtain the signal control action based on the state in the current control step, which is used as a backup choice for signal control decision during the reasoning of the LLM agent.
Input	Intersection index, Occupancy
Output	An integer representing the phase index as the initial signal control decision under the current environment for the specified intersection.
Example	Call the function as follows: Initial_decision = *Get_Initial_Decision ('J1')* Output: Initial_decision = 1

Table 6
Prompt components for final signal control decision tools.

Tool	*Get_Final_Decision*	*Justify_Decision_Logic*
Function	This tool is designed to use the CU -based control logic to obtain the signal control action based on the state in the current control step, which is used as the final signal control decision.	This tool identifies the currently active traffic signal phase at the intersection, which is crucial for the LLM to understand the traffic flows of which movements are being allowed to pass.
Input	Intersection index, Urgency	Intersection index, Final_decision
Output	An integer representing the phase index as the final signal control decision under the current environment for the specified intersection.	A string describing the available action set, the current traffic scenario based on perception, including the movement-level occupancy, phase-level clearance urgency and the existence of emergency vehicles, the initial signal control decision and the reasoning to obtain the final signal control decision
Example	Call the function as follows: Final_decision = *Get_Final_Decision ('J1')* Output: Final_decision = 1	Call the function as follows: Step_explanation = *Justify_Decision_Logic ('J1')* Output: Step_explanation = "The final decision is to set Phase 2 as the green signal in the next control step, which is suitable for the current standard situation. There are no emergency vehicles in any movements. This signal control action will help manage the traffic flow effectively, especially addressing the high urgency on movements '-E1–s' and 'E0–s'. "



(a) 3-way intersection **(b) 4-way intersection**

Fig. 4. Simulation models.

3.3.2. *LLM reasoning for final signal control decision*

In real-time signal control operation, the YOLO-based perception tool detects the vehicles of different movements and passes the details of these identified vehicles as well as the clearance urgency evaluation results to the LLM. From the definition of the clearance urgency, the corresponding signal control strategy would be to switch the phase with the largest value of U_t^p into the green phase for the next control step. Based on the idea of this control strategy, two decision tools, *Get_Final_Decision* and *Justify_Decision_Logic*, as shown in Table 6, are designed to output the signal control decisions based on both the initial RL-driven solution and the real-time LLM-generated decision. Output in the standard format, the decision is then passed to the signal controller for implementation, with detailed explanation on the reasoning process behind the final decision. It is noted that the LLM-generated control decision should also meet the general constraints like minimum and maximum green time. If an inappropriate response demonstrating LLM hallucination or timing parameter constraints violation, e.g., maximal green time, is obtained, the RL-driven control decision will be adopted for execution as backup.

Thus, the adaptive signal control optimization of the intersection is realized by the LLM augmented by the above perception, reasoning and decision tools with tailored prompts. Leveraging LLM's processing ability of natural language and the translation ability of human intention, the vision-based perception tool and the corresponding CU-based control strategy makes the full use of traffic information carried by real-time image and contributes in flexibly adjust the control decision in response to evolution of traffic flows and service priority heterogeneity.

4. Evaluation

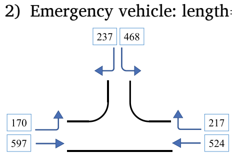
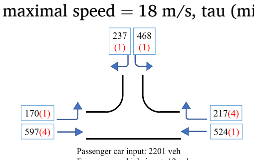
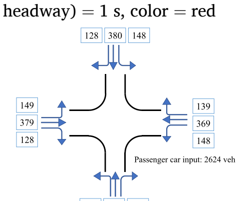
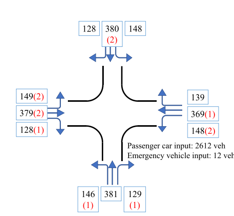
This section presents the simulation evaluation and discussion of the proposed method for its performance in intersection signal control optimization. Two kinds of intersection layout, three-way and four-way, and two types of vehicle composition, only passenger cars representing normal traffic and mixed vehicle input of cars and emergency vehicles, are adopted for the design of diversified scenarios. Comprehensive analysis regarding the control performance under normal traffic state and the progression efficiency of emergency vehicles considering traffic incident case is conducted, together with horizontal comparison with a fixed-time method [64], an adaptive signal control method [65–66], a purely RL-driven control method [62], a DRL-based method [9,67], and an integrated LLM-driven control method using non-vision-based detection, which is hereinafter referred to as LLM-assisted light [34].

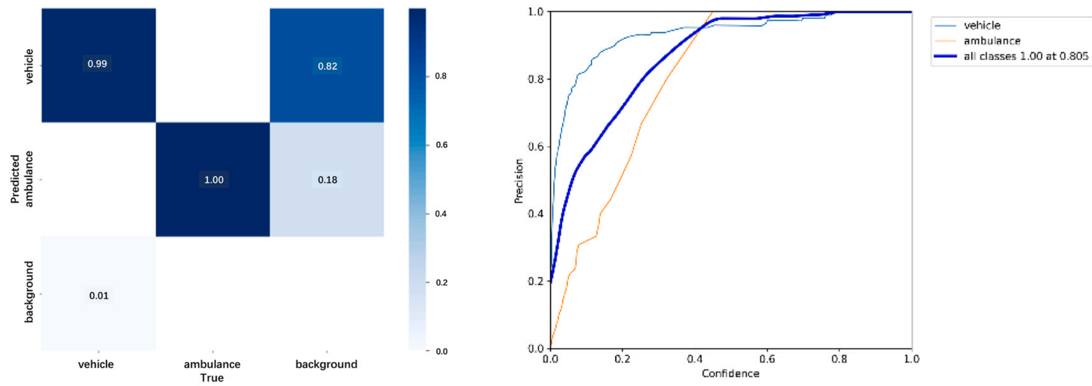
4.1. *Experimental settings*

The comprehensive experimental study was conducted using a PC with the following specifications: an AMD Ryzen 7 5800 H CPU @ 3.20 GHz, 16.0 GB of RAM, and one NVIDIA GeForce RTX 3060 GPU.

Two simulation models were built based on a widely used simulator, Simulation of Urban Mobility (SUMO) developed by employees of the Institute of Transportation Systems at the German Aerospace Center [68], as shown in Fig. 4, a 3-way intersection, and a 4-way standard intersection, with three-lane entry and two-lane exit composing each link of the intersection in both cases. As an open source, highly portable, microscopic and continuous traffic simulation package designed to handle large networks, SUMO allows for intermodal simulation with a large set of tools for scenario creation. For the selected benchmark methods using non-vision-based detector data as input, lane area detectors were set in each entry with a length of 50 m to obtain the occupancy data, which is denoted as the blue area in Fig. 4. According to assumption (d), the available phase sets, with three phases for the 3-way

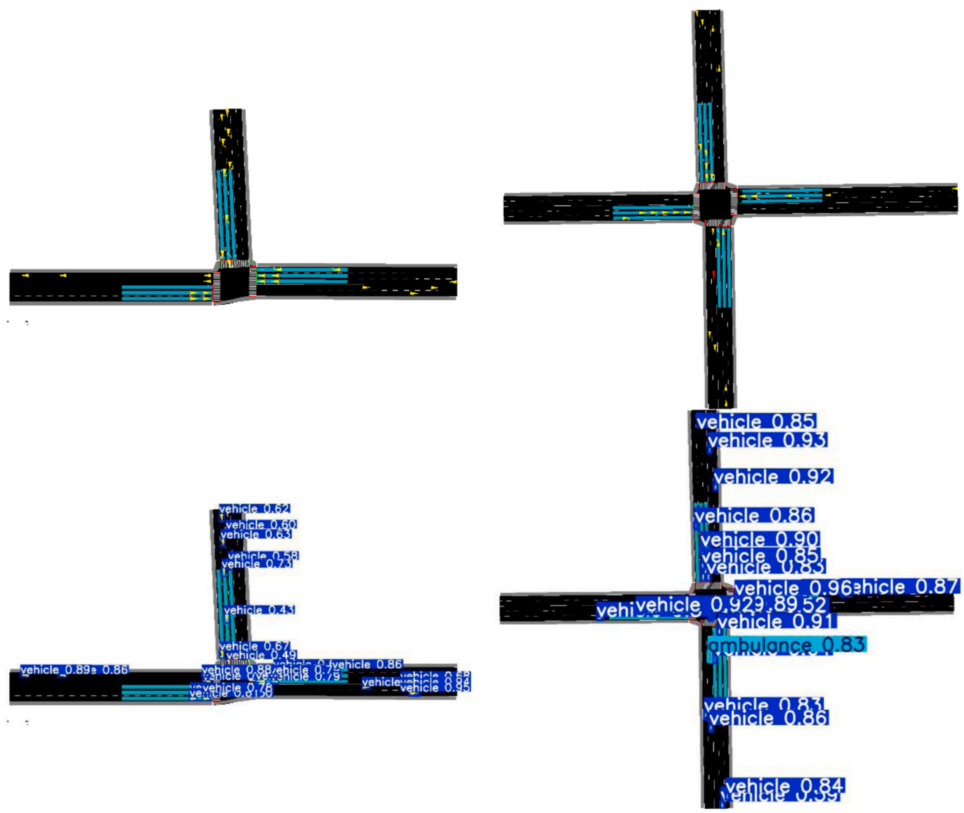
Table 7
Traffic demand scenarios of simulation case study.

Scenario No.	1	2	3	4
Intersection layout	3-way		4-way	
Scenario	Normal traffic	Emergency vehicle	Normal traffic	Emergency vehicle
Simulation period	1 hour			
Control step	5 s			
Vehicle composition	1) General passenger car: length= 4.5 m, width= 1.8 m, maximal speed = 15 m/s, tau (minimal headway) = 1 s, color = yellow 2) Emergency vehicle: length= 7 m, maximal speed = 18 m/s, tau (minimal headway) = 1 s, color = red			
Vehicle input				



(a) Normalized Confusion Matrix **(b)Precision-Confidence Curve**

Fig. 5. Training results of YOLO v11.



(a) 3-way normal traffic scenario sample **(b) 4-way emergency scenario sample**

Fig. 6. Demo images of YOLO v11 detection. (The vehicle label refers to general passenger car while the ambulance label refers to the emergency vehicle. Then values after class label refers to the confidence of the detection using the customized YOLOv11 detector).

intersection and four phases for the 4-way intersection, are also defined as shown in the figure, and input as the phase option for signal control action generation. In addition, a yellow light of 3 seconds is set following every green phase.

Both normal traffic and emergency vehicle scenario are designed as given in Table 7 [82], where the text in red means the input load of emergency vehicles. It is noted that for the emergency vehicle scenario, the input timepoints of the emergency vehicles are randomly determined.

For the vision-based perception tool, the customized dataset used to train and fine tune the YOLO model was established using the “Traci” interface to capture the screenshots of the intersection during the simulation run. A total of 200 images, with a resolution of 2550 * 1295 pixels, were collected. The training process is configured with the following parameters: resized image resolution *imgsz* $= [2550, 1295]$, training epochs $= 200$, number of images per batch $= 2$, number of worker threads $= 1$.

The LLM used in the experiments, including the LLM of the proposed method and the LLM of LLM-assisted light was GPT-4 API released by OpenAI [69], and Fig. 3 actually presents an example of the log showing how the LLM complete the signal control of one control step through the task-specific prompts under the proposed control framework.

4.2. YOLO-based vehicle detection evaluation

The custom dataset was split into three sets: training, testing, and

Table 8
Signal control models used for horizontal comparison.

Model	Description	Parameters	
Fixed-time	Cycle-based, uniform green time duration for all phases	Phase length (s)	20
Self-organizing traffic light (SOTL)	Cycle-based, dynamic phase length based on self-organizing principles, where a "… elements of the self-organizing system are designed to dynamically and autonomously solve a problem or perform a function at the system level." [66]	Minimal green time (s)	5
		Platoon size threshold (veh*s)	3
RL-based	Non-cycle-based, adaptive signal controller realized by executing signal phase switching between any two consecutive control steps according to the policy of the RL agent, aiming to reduce the average waiting time. (Details are introduced in Subsection 3.3.1)	Batch size	60
		Number of steps for each environment per update	300
		Number of epoch	5
		Learning rate	0.001
		Number of training time steps	1000000
Deep Q-learning network (DQN)-based	Non-cycle-based, adaptive signal controller using a deep Q-learning network for traffic condition extraction, and executing phase extension or truncation of the current phase aiming to reduce the total vehicular delay.	Number of episodes	2000
		Action space size	3
		Learning rate	0.1
		Reward decay rate	0.9
		Replay memory size	200
		Target network update frequency (iteration)	50
		Exploration policy parameter of e-greedy	0.9
LLM-assisted light	Integrated RL-based control and LLM-driven decision-making using non-vision-based detection (occupancy of lane groups) [16]	Control step size (s)	5
Proposed method	Integrated RL-based control and LLM-driven decision-making using vision-based detection (clearance urgency of lane groups, as introduced in Section 3.2)	Control step size (s)	5
		Confidence threshold of vehicle detection	0.4

validation, with a ratio of 18:1:1, respectively. The evaluation of the vehicle detection accuracy is based on the mean Average Precision (mAP) metric, which considers both the precision and recall of the detected objects as well as the model's confidence in generating bounding boxes. With training results shown in Fig. 5, the YOLO v11 model obtained a mAP50 of 98.8 % and a mAP50 of 99.5 % for the class of general passenger car and emergency vehicle in the training subset (180 images), respectively. The index "mAP50" refers to the mean average precision with a IoU (Intersection over Union) threshold of 0.50. For the validation subset (20 images), the customized YOLO detector obtained a mAP50 of 99.5 % and a mAP50 of 91.9 % for the class of general passenger car and emergency vehicle, respectively. For the testing subset (20 images), the customized YOLO detector obtained a mAP50 of 99.1 % and a mAP50 of 96.5 % for the class of general passenger car and emergency vehicle, respectively. As shown in Fig. 6 where the upper two image samples are the screenshots of the simulation junctions while the lower two images demonstrate the detection results obtained by the customized YOLO detector, the two image samples indicate that the YOLOv11 model is satisfactorily accurate in detecting different types of vehicles. Moreover, it takes only approximately 12.3 ms to process each image, enabling it to function effectively in real-time for traffic state perception. As the precision in Fig. 5(b) equals the amount of truly detected vehicles divided by the total number of detected objects retrieved by the algorithm, thus for the integrated vision-based perception tool in signal control implementation, the thresholds for identification of general passenger cars and emergency vehicles are both set as 0.4.

4.3. Signal control performance evaluation

To quantify the signal control performance of all three methods for horizontal comparison, three indicators, average speed (AS), average waiting time (AW) and total travel time (TT), were selected. Each method was tested under the specific scenario and the specific intersection layout for three times, and the average values of the evaluation indexes of these three parallel tests were used for result analysis, in order to guarantee the generality of results.

4.3.1. Signal control performance under normal traffic scenario

For horizontal comparison under normal traffic scenario, five baseline methods were selected, as shown in Table 8, hereinafter SOTL is used to represent the Self-organizing traffic light method for short, and DQN is used to represent the Deep Q-learning network-based method.

Among these five benchmark methods, fixed-time method and SOTL method are non-learning traffic signal controllers. For the remaining learning-based methods, RL-based method and DQN-based method are traditional learning-based method, while LLM-Assisted light method and our proposed method are integrated LLM and learning-based methods. It is noted that the RL-based method and DRL-based method are different in terms of control logic, learning algorithm and elements of RL (namely state, action and state) although they both model the intersection as a RL through discrete traffic state encoding. Both the LLM-Assisted light method and our proposed method use the RL-based method as the initial signal control solution, as mentioned in the methodology section.

Table 9 demonstrates the comparison of signal control performance of all six methods. An evident gap can be seen between the fixed-time method the others by more than 7.92 %, 37.86 % and 13.77 % in the indicators of AS, AW and TT respectively for the 3-way intersection case while the difference is even larger in the 4-way intersection case, with an improvement of more 39.95 %, 67.19 % and 47.85 % in terms of AS, AW and TT, respectively. Such difference is due to the fact that the fixed-time method adopts a fixed-time control type while all the others belong to adaptive control type, thus great delay may be generated by longer red phases without regard of the actual traffic demand fluctuation when the intersection uses fixed timing scheme.

Take the five adaptive control methods separately for comparison, the control performance can be sorted in such an order from good to bad: the proposed method, LLM-assisted light method, RL-based method, DRL-based method and SOTL, with only one exception that the total travel time of DRL-based method is slightly smaller than SOTL in the 4-way intersection case. The reason for DRL-based method may result from action space defined as the extension duration of the current phase and the control logic of executing based on the fixed phase structure. Based on a basis timing scheme, the phase sequence is fixed and a phase cannot be skipped even if the given minimum green phase time presents an oversupply considering the actual incoming vehicles.

As for the RL-based method and the two LLM-based methods using the RL-based controller as initial solution, it is obvious that the proposed method outperformed the other two with improvements of up to 12.23 %, 15.11 % and 9.68 % in terms of average speed, average waiting time and total travel time, respectively, using the RL-based method

Table 9
Signal control performance comparison under normal traffic scenario. (The values in bold refer to the best one among all the methods and the values in parentheses denote the improvement as compared with the fixed-time method).

Method \ Evaluation index	AS (m/s)	AW (s)	TT (s)
(a) 3-way intersection scenario (Scenario No. 1)			
Fixed time	6.19	14.79	92637
SOTL	6.86(+10.82%)	8.18(-44.69%)	78231(-15.55%)
RL	6.95(+12.28%)	7.54(-49.02%)	76568(-17.35%)
DQN	6.68(+7.92%)	9.19(-37.86%)	79883(-13.77%)
LLM-assisted light	7.06 (+14.05%)	7.16 (-51.59%)	75274 (-18.74%)
Proposed method	**7.12 (+15.02%)**	**7.14 (-51.72%)**	**74053 (-20.06%)**
(b) 4-way intersection scenario (Scenario No. 3)			
Fixed time	3.88	72.13	271420
SOTL	5.77(+48.71%)	22.44(-68.89%)	145814(-46.28%)
RL	5.97(+53.87%)	19.52(-72.94%)	138246(-49.07%)
DQN	5.43(+39.95%)	23.66(-67.19%)	141541(-47.85%)
LLM-assisted light	6.14 (+58.25%)	17.55 (-75.67%)	130650 (-51.86%)
Proposed method	**6.70 (+72.68%)**	**16.57 (-77.03%)**	**124863 (-53.99%)**

(a) 3-way scenario case **(b) 4-way scenario case**

Fig. 7. Snapshot of the intersection state with different control decisions. (Yellow triangles refers to the simulation vehicles in the SUMO simulator).

as baseline. As compared with the 3-way intersection scenario, the superiority of both two methods integrating RL and LLM over the purely RL-driven controller is obviously larger under the 4-way intersection scenario. The reason may be that the traffic pattern complexity of 4-way layout is greater than that of 3-way layout, leaving more room for optimization by leveraging the scenario understanding and reasoning capacity of the LLM agent. As for the gap between the LLM-based signal control methods with non-vision-based perception and with vision-based perception, Fig. 7 provides two cases to demonstrate the difference. In the snapshot of the 3-way intersection case, the decision of the proposed method is phase option 2 as given in Fig. 3(a), while the decision of LLM-assisted light is phase option 1, which shows that for phases with similar demand levels, the green time is assigned to the movement with more urgency. As for the 4-way intersection case, the decision of the proposed method is phase option 1 while the decision obtained by LLM-assisted light is phase option 3, which also justifies the control idea of clearance urgency-based decision tool.

4.3.2. Signal control performance under emergency vehicle scenario

Under emergency vehicle scenario, the weights of general passenger car and emergency vehicle were set as 1 and 10 respectively, considering the priority of the emergency vehicles in practical. In addition to the overall signal control performance, the progression efficiency of the emergency vehicles is also compared through three indicators, average control delay (AD), average number of stops (ANS) and average travel time (ATT), as shown in Table 10. Based on the comparison in Section 3.3.1, only the best three methods, RL-based method, LLM-assisted light method and our proposed method are tested under the emergency vehicle scenario. Horizontally, the gain in traffic efficiency of the emergency vehicles is obviously larger than that of general passenger cars, especially in terms of average delay, which demonstrates the significance of signal control with emergency preemption.

The superiority of the proposed method regarding the overall control performance still retains under the traffic incident scenario, meanwhile leads in the progression efficiency of the emergency vehicles. LLM-assisted light also manages to prioritize the emergency vehicles over the general passenger cars at the cost of a little decreasing of the progression quality of the general passenger cars. The compromise between the benefits of two different types of vehicles in LLM-assisted light is mainly realized by setting the traffic light of the lane group where the emergency vehicle runs as green at once when the emergency vehicle enters the detection zone. However, such preemption mechanism may

Table 10
Signal control performance comparison under emergency vehicle scenario. (The values in bold refer to the best one among all the methods and the values in parentheses denote the improvement as compared with the RL-based method).

(a) 3-way intersection scenario (Scenario No. 2)

Evaluation index	General passenger car			Emergency vehicle		
Method	AS (m/s)	AW (s)	TT (s)	AD(s)	ANS	ATT(s)
RL-based method	7.01	7.17	77940	16.14	0.64	28.83
LLM-assisted light	6.9 (−1.57 %)	7.66 (+6.83 %)	76290 (−2.12 %)	12.53 (−22.37 %)	0.58 (−9.38 %)	25.33 (−12.14 %)
Proposed method	**7.08** **(+0.99 %)**	**7.05** **(−1.67 %)**	**73152** **(−6.14 %)**	**10.02** **(−37.92 %)**	**0.16** **(−75 %)**	**22.89** **(−20.60 %)**

(b) 4-way intersection scenario (Scenario No. 4)

Evaluation index	General passenger car			Emergency vehicle		
Method	AS (m/s)	AW (s)	TT (s)	AD(s)	ANS	ATT(s)
RL-based method	5.91	18.56	135350	34.62	1.19	49.14
LLM-assisted light	5.81 (−1.69 %)	20.14 (+8.51 %)	137072 (−1.27 %)	21.56 (−37.72 %)	0.96 (−19.33 %)	36.46 (−25.80 %)
Proposed method	**6.78** **(+12.83 %)**	**15.87** **(−14.49 %)**	**122060** **(−9.82 %)**	**13.42** **(−61.24 %)**	**0.28** **(−76.47 %)**	**27.97** **(−43.08 %)**

Table 11
Signal control performance comparison under different data inputs. (The values in bold refer to the best one among all the methods).

LLM-based controller	Evaluation index	AS (m/s)	AW (s)	TT (s)
3-way (Scenario No. 1)	LLM-assisted light using occupancy data	7.06	7.16	75274
	LLM-assisted light using flow data	7.00	8.09	77314
	Proposed method	**7.12**	**7.14**	**74053**
4-way (Scenario No. 3)	LLM-assisted light using occupancy data	6.14	17.55	130650
	LLM-assisted light using flow data	6.54	17.35	128731
	Proposed method	**6.70**	**16.57**	**124863**

be aggressive as it may take more than one control step for the emergency vehicle to pass the stop-line, thus posing more control delay to the vehicles in other phases. In contrast, the clearance urgency index considers both the spatial distribution and priorities of different vehicle types, which is more flexible than the single criterion of existence in LLM-assisted light. Regarding the performance indicators of emergency vehicles, the decrease of average number of stops of the proposed method is far larger than the others, which is also a reflection of the effectiveness of using clearance urgency to provide right-of-way on customized demand.

4.3.3. Signal control performance using different data source inputs

As different data sources and detection indicators indeed provides the portrait of the traffic situation in different degrees, the contribution to signal control optimization is also different. As stated earlier, LLM-assisted light method uses the lane group detector to collect the vehicular occupancy data as input, while our proposed method uses the image data to extract the vehicle positions for calculation of the clearance urgency indicator as the input of signal control decision. Here in this section, another detector, event-triggered fixed detector installed at the entry of each lane group is further tested, with horizontal comparison of LLM-assisted light and the proposed method to explore the impact of different data sources or detection conditions. By setting up the fixed section detector at the starting point of each lane group, the detector will be triggered to collect the entry time when a vehicle is loaded on the specific lane group in the SUMO simulator. Based on the LLM-assisted light method, the incoming flow of each phase was used as the alternative of the occupancy indicator, thus the signal control decision was changed to assign the phase with the largest incoming vehicles with green time for the next control step.

Under the normal scenarios, the alternative LLM-based controller using flow data as input was tested for three parallel experiments, as shown in Table 11. From the results, the proposed method still shows advantages over the other two regarding all the control performance indicators, while the LLM-based controller using flow data and LLM-assisted light using occupancy data have one victory apiece, although the gap doesn't seem like a lot. What is certain is that phase-level incoming flow obtained by fixed detectors is effective to be used in signal control decision-making, while the advantage of the vision-based input and the urgency indicator of the proposed method can be further proved.

The different performance in 3-way case and 4-way case may be related to the phase structure design. As shown in Fig. 4, three phase options are available in the 3-way intersection case, with each phase assigning right-of-way to the movements of the same link. Thus, the occupancy data of the detection zone of three links is more suitable than the incoming flow of three links for the signal control decision to deal with immediate demand for green time. As for the 4-way intersection, the four phases in the phase structure in Fig. 4 is arranged by the directions of movements, instead of the movements of one link. Considering the different lane numbers of different movements, the occupancy of left-turn lanes with a smaller vehicle flow sometimes may be larger than that of the straight lanes, which may eventually drive the LLM-assisted light to give a prior green time to the left-turn movements, although the incoming flow of straight movements is larger. Such issue can be solved by the LLM-based controller using flow data, which

Table 12
Signal control performance comparison under different control steps. (The values in bold refer to the best one among all the methods and the values in parentheses denote the improvement as compared with the proposed method using a control step of 5 s).

Control step size (s) \ Evaluation index		AS (m/s)	AW (s)	TT (s)
3-way (Scenario No. 1)	5	7.12	7.14	74053
	7	**7.20 (+1.12%)**	**6.65 (-6.86%)**	**72490 (-2.11%)**
	9	7.00(-1.68%)	9.29(+30.11%)	79667(+7.58%)
	11	6.83(-4.07%)	11.11(+55.60%)	84186(+13.68%)
4-way (Scenario No. 3)	5	6.70	16.57	124863
	7	**6.79 (+1.33%)**	**14.86 (-10.32%)**	**117965 (-5.52%)**
	9	6.64(-0.89%)	15.86(-4.28%)	124012(-0.68%)
	11	6.54(-2.39%)	16.50(-0.42%)	125855(+0.79%)

Table 13
LLM models used for comparison.

LLM model	Description
GPT 4	OpenAI's most advanced generative artificial intelligence chatbot released in 2023, focusing on improving accuracy, efficiency, and contextual understanding based on a more extensive and diverse training dataset
GPT 4o	Released in May 2024, further refined OpenAI's language models by delivering more concise and disciplined responses, with structured explanations for scientific and technical contexts as well as enhanced creative writing and detailed literary analysis
GPT 4o-mini	A lighter, cost-efficient variant and scaled-down version of GPT−4o in July 2024 supporting the same range of languages as GPT−4o [70,71].
Qwen2.5:72b Qwen2.5:32b Qwen2.5:14b Qwen2.5:7b Qwen2.5:3b Qwen2.5:1.5b Qwen2.5:0.5b	Qwen2.5 is the latest series of Qwen large language models, with a number of base language models and instruction-tuned language models ranging from 0.5 to 72 billion parameters. Upon Qwen2, Qwen2.5 has greatly improved capabilities in coding and mathematics, instruction following, and is more resilient to the diversity of system prompts [72,73].
QwQ:32b	QwQ is the medium-sized reasoning model of the Qwen series. Compared with conventional instruction-tuned models, QwQ, which is capable of thinking and reasoning, can achieve significantly enhanced performance in downstream tasks, especially hard problems [74].
Llama3.3:70b	The Meta Llama 3.3 multilingual large language model (LLM) is an instruction tuned generative model in 70B (text in/text out), which is optimized for multilingual dialogue use cases [75].
Llama3.2:3b Llama3.2:1b	Llama 3.2 collection of multilingual large language models is a collection of pretrained and instruction-tuned generative models in 1B and 3B sizes (text in/text out). The Llama 3.2 instruction-tuned text only models are optimized for multilingual dialogue use cases, including agentic retrieval and summarization tasks [76].
Deepseek-r1:70b Deepseek-r1:32b Deepseek-r1:7b	DeepSeek's first-generation of reasoning models, with comparable performance to OpenAI-o1, including six dense models distilled from DeepSeek-R1 based on Llama and Qwen [77].
Mistral:7b	The Mistral Large Language Model (LLM), released by Mistral AI team, is a pretrained generative text model with 7 billion parameters [78,79].

provides the absolute vehicular number in a larger detection scale, which can improve the control performance through a more forward planning. This comparison also implies the applicability of a certain traffic evaluation indicator or signal controller is related to quite a number of factors, like the data availability, topological layout, signal timing design and actual traffic demand.

4.3.4. Sensitivity analysis of control step size

Considering the potential relationship between the detection scale and the control step size, three more step sizes, 7 s, 9 s and 11 s were further tested to explore whether there exists an optimal control step corresponding to the detection scale of the intersection to realize the best control performance of the proposed method. As shown in Table 12, the control performance first improves when the control step increases to 7 s and then degrades as the control step size continues to increase for both the 3-way and 4-way intersection scenarios, especially obvious in terms of the average waiting time.

As the real-time snapshots capture more information of the intersection entries, thus a larger control step is able to clear the involved vehicles considered in the control objective. However, the increase of the control step also means the increased delay for the queuing vehicles and increased vehicles joining the queue during the red phase. Thus, from the results, a control step of 7 s can be regarded as the optimal control step size for the proposed method and the two test scenarios, reaching a balance between the control benefit brought by a better demand-supply matching and the delay caused by longer red phase.

4.4. Computation performance analysis of LLM agent

Regarding the contribution of LLM agent's performance to the signal control optimization task, the case of 3-way intersection under normal traffic scenario (Scenario No. 1) is selected for a further test using two more different versions of GPT 4, GPT 4o and GPT 4o-mini, and other LLM models as shown in Table 13.

Testing under Scenario No. 1, Table 14 gives the comparison of control performance and computation cost. The results of GPT series are obtained through cloud computing by calling the API using the device mentioned in Section 4.1. As for the other LLM models, their tests are realized through local computing by a AMD Ryzen Threadripper PRO 7975WX workstation with 4.00 GHz 755.0 GB of RAM, and four NVIDIA GeForce RTX 4090 GPU, due to the high configuration requirement. The total computation cost is the time spent for the signal control of the whole 3600 s simulation period, while the step cost refers to the time spent on one decision making cycle including perception, urgency evaluation and signal control solution.

For the comparison within GPT series, the control performance of GPT 4o is slightly worse than that of GPT 4, yet still surpasses that of purely RL-based method and evenly matches that of LLM-assisted light in Table 9(b), while the degradation becomes more significant when the LLM agent changes to GPT 4o-mini, especially in terms of the average waiting time, which is also justified in existing research that the 4o model is more reliable and accurate compared to the 4o-mini model [80]. However, the saving of computation cost when using the GPT 4o-mini can reach nearly a half, which is more than two hours. Considering the balance between solution quality and computation cost, GPT 4o can be an acceptable and economical alternative.

Regarding the comparison within Qwen2.5 series together with QwQ, Qwen2.5 with 72b parameters obtained the highest average speed while the 32b version obtained the smallest average waiting time and total travel time. Considering the computation cost of these two versions, Qwen2.5 with 32b parameters is a cost-effective choice. As for the

Table 14
Performance comparison using different versions of GPT under Scenario No. 1. (The values in bold refer to the best one in the corresponding LLM series and the values in parentheses in GPT series denote the percentage difference as compared with GPT4).

GPT Version \ Evaluation index	Signal control performance			Computation performance	
	AS (m/s)	AW (s)	TT (s)	Total computation cost (s)	Step computation cost (s)
GPT 4	**7.12**	**7.14**	**74053**	15645	32.594
GPT 4o	7.07 (-0.70%)	7.43 (+4.06%)	76464 (+3.26%)	11645 (-25.57%)	24.311
GPT 4o-mini	6.8 (-4.49%)	8.89 (+24.51%)	79428 (+7.26%)	**8025 (-48.71%)**	**15.257**
Qwen2.5:72b	**6.79**	8.90	80432	37205	77.351
Qwen2.5:32b	6.74	**8.42**	**79455**	9943	21.110
Qwen2.5:14b	6.53	10.64	84887	5140	10.426
Qwen2.5:7b	6.48	187.49	295792	4134	5.856
Qwen2.5:3b	5.32	132.47	271261	2345	3.983
Qwen2.5:1.5b	5.40	119.92	259527	1323	2.132
Qwen2.5:0.5b	5.49	142.82	269784	**838**	**1.416**
QwQ:32b	6.06	14.27	104910	37506	88.596
Llama3.3:70b	**6.29**	**11.44**	**88898**	33138	72.756
Llama3.2:3b	5.04	73.74	211269	1926	3.684
Llama3.2:1b	4.58	96.39	243311	**1641**	**3.427**
Deepseek-r1:70b	**6.92**	**9.32**	**87381**	47764	99.716
Deepseek-r1:32b	6.14	19.13	116871	21909	45.645
Deepseek-r1:7b	5.10	33.34	131332	**6635**	**13.514**
Mistral:7b	**6.83**	**8.69**	**79757**	**2707**	**5.553**

performance of QwQ, its control performance lies somewhere between Qwen2.5:7b and Qwen2.5:14b but its computation cost is even larger than Qwen2.5b with 72b parameters. Although the average speed of Qwen2.5 with 7b parameters is larger than QwQ, the average waiting time and total travel time deteriorate sharply by one order of magnitude, which may the cost of less phase switching to remain relatively high vehicular speed. Compared with the results of Qwen2.5 series, QwQ is not a good option for signal control application.

The results of Llama series show an obvious gap between 3.3 version and 3.2 version, both in terms of control performance and computation cost by up to an order of magnitude. Although Llama 3.3 can obtain relatively good performance of signal control, it is time-consuming and not applicable to real-time decision-making. By contrast, the performance of three versions of Deepseek series is more even considering their parameter sizes. However, the computation cost of Deepseek-r1 with 70b parameters is larger than that of Llama with the same parameter size. Including Mistral model, the best models of these four open lightweight LLM series in terms of average speed are Qwen2.5:72b, Llama3.3:70b, Deepseek-r1:70b and Mistral:7b.

As compared with the GPT series, the ranking from high to low in terms of average speed is GPT, Deepseek, Mistral, Qwen and Llama, while the ranking from low to high in terms of average waiting time and total travel time is GPT, Qwen, Mistral, Deepseek and Llama. Considering the computation cost of such five LLM models as the best one in terms of control performance within the corresponding series, the ranking of computation cost from low to high is Mistral, Qwen, GPT, Llama and Deepseek. Among the lightweight models, Mistral is an economical option with its step cost superior to the others by an order of magnitude. With a similar control performance to GPT 4o-mini, the step cost of Mistral only accounts for about one-third of the step cost of GPT 4o-mini, which shows prospect in empirical application [81].

It is noted that the Deepseek series and QwQ both have a "chain-of-thought" mechanism to explicitly show the reasoning process, which leads to the two largest computation cost of Deepseek-r1:70b and QwQ:32b to some extent, and also makes them prone to output error. Regarding the performance of the fixed-time method in Table 9 as the baseline, the performance of some LLM models like Qwen2.5 with less than 7b parameters, Llama 3.2 series, Deepseek r1 with less than 70b parameters is even worse, which is manifested as long queue and even link overflow in the simulator. It comes to a conclusion that the error tolerance ability of the proposed method is related to the intelligence level of the LLM agent, which is quantified by the parameter quantity. Thus, to make sure the safe execution of the proposed method from the perspective of traffic operation management considering its application in real world, the LLM agent should be intelligent enough to solve the signal control problem based on the well-designed prompt. On the other hand, prompt engineering should be another way of risk management, such as enhancing and enriching the prompt system to handle the wrong decisions of the LLM agent through more tests in future work.

5. Conclusion

Leveraging the remarkable learning and adaptation capabilities of LLMs within deployed environments, this study explores the potential of integrating YOLOv11 object detection, RL and LLM for contextual analysis in signal control decision making of single intersections. By transforming the real-time image or video footage data of either simulation or field detection into insightful information of the different approaching flows within the intersection scale, the competition for the green time among different signal phases is quantified through the clearance urgency indicator, which provides a novel perspective of control strategy unlike the traditional delay-based ones. Equipped with such perception and reasoning tools, the LLM agent is developed to analyze the real-time traffic state and come up with signal control decision with human-level proficiency. Simulation experiments under both normal traffic scenario and emergency vehicle scenario demonstrate the capacities of the proposed smart traffic light in terms of the accuracy of grasping traffic demand pattern, as well as the flexibility and scalability of handling different control needs. Under normal traffic scenario, the proposed method can increase about 10 % in average speed and decrease about 5 % in average waiting time as compared with the benchmark LLM-based method using occupation detection as data input for LLM reasoning. Such gaps increase to more than 14 % and 20 % in average speed and average waiting time respectively under emergency vehicle scenario, with even larger gap reaching up to 21 % in terms of average delay of emergency vehicles, demonstrating the superiority of the proposed method in ensuring the traffic efficiency of both general vehicles and emergency vehicles. A comprehensive comparison regarding different control methods, different data source inputs, different control step sizes, and different LLM models also provides deep insights into the superiority of the proposed method, as well as its applicability and feasibility in real-world and real-time application.

Future work will focus on expanding the applicability of the proposed method through evaluation under more diverse scenarios such as more vehicle types like transit buses, more demand patterns like tidal commuting or dominated turning flows and larger control scale like arterials or networks considering the co-relationship between consecutive intersections. Furthermore, the scalability of the integration framework to augment LLM with different external tools can be explored to provide more evaluation indicators and control strategies with customized prompts, facilitating a more elaborated and responsive signal control system.

Author contributions

The authors confirm contribution to the paper as follows: study conception and design: Jiarong Yao, Kim Hui Yap, Rong Su; data collection: Jiarong Yao; analysis and interpretation of results: Jiarong Yao, Jiangpeng Li, Xiaoyu Xu; draft manuscript preparation: Jiarong Yao, Jiangpeng Li, Xiaoyu Xu, Chaopeng Tan, Kim Hui Yap, Rong Su. All authors reviewed the results and approved the final version of the manuscript.

CRediT authorship contribution statement

Chaopeng Tan: Writing – review & editing, Formal analysis. **Kim Hui Yap:** Supervision, Conceptualization. **Rong Su:** Supervision, Conceptualization. **Jiarong Yao:** Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Jiangpeng Li:** Writing – review & editing, Validation, Software, Formal analysis. **Xiaoyu Xu:** Writing – review & editing, Software, Formal analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Wan, K. Wu, T. Shi, J. Wang, Fair and Efficient Traffic Light Control with Reinforcement Learning. International Symposium on Intelligent Computing and Networking, Springer Nature, Cham, Switzerland, 2024, pp. 17–33.

[2] A. Wong, T. Bäck, A.V. Kononova, A. Plaat, Deep multiagent reinforcement learning: Challenges and directions, Artif. Intell. Rev. 56 (6) (2023) 5023–5056.

[3] R. Kumar, N.V.K. Sharma, V.K. Chaurasiya, Adaptive traffic light control using deep reinforcement learning technique, Multimed. Tools Appl. 83 (5) (2024) 13851–13872.

[4] W. Hua, G. Zheng, V. Gayah, Z. Li, Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation, ACM SIGKDD Explor. Newsl. 22 (2) (2021) 12–18.

[5] S. Abbasi, S.S. Mousavi, E. Farbod, M. Yousefi, M. Parvin, Hybrid data mining and data-driven algorithms for a green logistics transportation network in the post-COVID era: A case study in the USA, Syst. Soft Comput. 6 (2024) 200156.

[6] R. Luo, Y. Zhang, Y. Zhou, H. Chen, L. Yang, J. Yang, R. Su, Deep Learning Approach for Long- Term Prediction of Electric Vehicle (EV) Charging Station Availability, IEEE Int. Intell. Transp. Syst. Conf. (ITSC) 2021 (2021) 3334–3339, https://doi.org/10.1109/ITSC48978.2021.9564633.

[7] Q. Chen. Congestion estimation and turning ratio prediction based on machine learning with applications in urban traffic light control. Master's thesis, Nanyang Technological University, Singapore. 2019.

[8] K. Shaaban, A. Hamdi, M. Ghanim, K.B. Shaban, Machine learning-based multi-target regression to effectively predict turning movements at signalized intersections, Int. J. Transp. Sci. Technol. 12 (1) (2023) 245–257.

[9] S. Shen, G. Shen, Y. Shen, D. Liu, X. Yang, X. Kong, PGA: An efficient adaptive traffic signal timing optimization scheme using actor-critic reinforcement learning algorithm, KSII Trans. Internet Inf. Syst. (TIIS) 14 (11) (2020) 4268–4289.

[10] A. Baher, R. Pringle, G.J. Karakoulas, Reinforcement learning for true adaptive traffic signal control, J. Transp. Eng. 129 (3) (2003) 278–285.

[11] H. Ammar, Y.Y.ılmaz Deep, reinforcement learning for intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 23 (1) (2020) 11–32.

[12] L. Li, Y. Lv, F. Wang, Traffic signal timing via deep reinforcement learning, IEEE/CAA J. Autom. Sin. 3 (3) (2016) 247–254.

[13] X. Liu, M. Zhu, S. Borst, A. Walid, arXiv preprint, arXiv:2302.03669,, Deep Reinf. Learn. Traffic Light Control Intell. Transp. Syst. (2023). arXiv:2302.03669,.

[14] W. Tong, J. Cao, A. Hussain, Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning, Transp. Res. Part C: Emerg. Technol. 125 (2021) 103046.

[15] S. El-Tantawy, B. Abdulhai, H. Abdelgawad, Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1140–1150.

[16] E. Van der Pol, F.A. Olienhoek, Coordinated deep reinforcement learners for traffic light control, Proc. Learn., Inference Control multi-Agent Syst. (NIPS 2016 8 (2016) 21–38.

[17] C. Chen, W. Hua, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, Z. Li, Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control, Proc. AAAI Conf. Artif. Intell. 34 (4) (2020) 3414–3421.

[18] T. Chu, J. Wang, L. Codeca, Z. Li, Multi-agent deep ` reinforcement learning for large-scale traffic signal control, IEEE Trans. Intell. Transp. Syst. 21 (3) (2019) 1086–1095.

[19] Y. Gong, M. Abdel-Aty, Q. Cai, M.S. Rahman, Decentralized network level adaptive signal control by multi-agent deep reinforcement learning, Transp. Res. Interdiscip. Perspect. 1 (2019) 100020.

[20] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, X. Huang, A Compr. Capab. Anal. GPT-3 GPT- 3. 5 Ser. Models (2023). CoRR abs/2303.10420 (.

[21] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre training of deep bidirectional transformers for language understanding, in: in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[22] A. Mastropaolo, S. Scalabrino, N. Cooper, D.N. Palacio, D. Poshyvanyk, R. Oliveto, Studying the usage of text-to-text transfer transformer to support code-related tasks (Madrid, ES), IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE) 2021 (2021) 336–347, https://doi.org/10.1109/ICSE43902.2021.00041.

[23] M. Mohammadi, A. Al-Fuqaha, Enabling cognitive smart cities using big data and machine learning: Approaches and challenges, IEEE Commun. Mag. 56 (2) (2018) 94–101.

[24] J. Rasley, S. Rajbhandari, O. Ruwase, Y. He, Deep Speed: System optimizations enable training deep learning models with over 100 billion parameters. in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, United States, 2020, pp. 3505–3506.

[25] Z. Yuan, S. Lai, H. Liu, CoLLMLight: Coop. Large Lang. Model Agents Netw. -Wide Traffic Signal Control. arXiv Prepr. (2025) arXiv:2503.11739,.

[26] D. Mahmud, H. Hajmohamed, S. Almentheri, S. Alqaydi, L. Aldaheri, R.A. Khalil, N. Saeed, Integrating LLMs with ITS: Recent Advances, Potentials, Challenges, and Future Directions, IEEE Trans. Intell. Transp. Syst. (2025) 1–36.

[27] S. Javaid, H. Fahim, B. He, N. Saeed, Large language models for UAVs: Current state and pathways to the future, IEEE Open J. Veh. Technol. 5 (2024) 1166–1192.

[28] Y. Tang, X. Dai, C. Zhao, Q. Cheng, Y. Lv, Large language model-driven urban traffic signal control. 2024 Australian & New Zealand Control Conference (ANZCC, IEEE, Gold Coast, Australia, 2024, pp. 67–71.

[29] X. Dai, Y. Tang, Y. Chen, X. Zhang, Y. Lv, Large Language Model-Powered Digital Traffic Engineers: The Framework and Case Studies, IEEE J. Radio Freq. Identif. 8 (2024) 780–787.

[30] L. Da, K. Liou, T. Chen, X. Zhou, X. Luo, Y. Yang, H. Wei, Open-TI: Open traffic intelligence with augmented language model, Int. J. Mach. Learn. Cybern. 15 (2024) 4761–-4786.

[31] S. Lai, Z. Xu, W. Zhang, H. Liu, H. Xiong, Large Lang. Models Traffic Signal Control Agent.: Capacit. Oppor. (2023) arXiv:2312.16044.

[32] L. Da, M. Gao, H. Mei, H. Wei, Prompt to Transfer: Sim-to-Real Transfer for Traffic Signal Control with Prompt Learning, Proc. AAAI Conf. Artif. Intell. 38 (1) (2024) 82–90.

[33] A. Pang, M. Wang, M.O. Pun, C.S. Chen and X. Xiong, iLLM-TSC: Integration reinforcement learning and large language model for traffic signal control policy improvement. (2024) arXiv preprint arXiv:2407.06025.

[34] M. Wang, A. Pang, Y. Kan, M.O. Pun, C.S. Chen and B. Huang, LLM-assisted light: Leveraging large language model capabilities for human-mimetic traffic signal control in complex urban environments. (2024) arXiv preprint arXiv:2403.08337.

[35] M. Movahedi, J. Choi, The Crossroads of LLM and Traffic Control: A Study on Large Language Models in Adaptive Traffic Signal Control, IEEE Trans. Intell. Transp. Syst. (2024), https://doi.org/10.1109/TITS.2024.3498735.
[36] M. Tislenko, D. Kisilev, Single Agent. Multi-Agent.: Improv. Traffic Signal Control (2024) arXiv, abs/2406.13693.
[37] Y. Tang, X. Dai, Y. Lv, Large Language Model-Assisted Arterial Traffic Signal Control, IEEE J. Radio Freq. Identif. 8 (2024) 322–326.
[38] L. Wen, X. Yang, D. Fu, X. Wang, P. Cai, X. Li, T. Ma, Y. Li, L. Xu, D. Shang, Z. Zhu, S. Sun, Y. Bai, X. Cai, M. Dou, S. Hu, B. Shi, Y. Qiao, arXiv preprint, Road. gpt-4v (ision): Early Explor. Vis. -Lang. Model Auton. Driv. (2023) arXiv:2311.05332.
[39] S. Zhou, J. Li, T. Yamauchi, J. Cai, K. Tei, Evaluating Vision-Language Models in Visual Comprehension for Autonomous Driving. 2024 IEEE 4th International Conference on Software Engineering and Artificial Intelligence (SEAI), IEEE, Xiamen, China, 2024, pp. 205–209.
[40] H. Tian, K. Reddy, Y. Feng, M. Quddus, Y. Demiris, P. Angeloudis, Large (Vision) Language Models for Autonomous Vehicles: Current Trends and Future Directions, Author Prepr. (2024), https://doi.org/10.36227/techrxiv.172963218.80161917/v1.
[41] S. Masri, H.I. Ashqar, M. Elhenawy, Large Language Models (LLMs) as Traffic Control Systems at Urban Intersections: A New Paradigm, (2024) arXiv preprint arXiv:2411.10869.
[42] K. Kanistras, G. Martins, M.J. Rutherford, K.P. Valavanis, A survey of unmanned aerial vehicles (UAVs) for traffic monitoring, in: international conference on unmanned aircraft systems (ICUAS), 2013, IEEE, Atlanta, GA, USA, 2013, pp. 221–234.
[43] G.R. Lira, Z. Kokkinogenis, R.J.F. Rossetti, D.C. Moura, T. Rúbio, A computer vision approach to drone-based traffic analysis of road intersections, in: IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, IEEE, Rio de Janeiro, Brazil, 2016, pp. 47–53.
[44] N. Dilshad, J.Y. Hwang, J.S. Song, N.M. Sung, Applications and challenges in video surveillance via drone: A brief survey, in: International Conference on Information and Communication Technology Convergence (ICTC), 2020, IEEE, Jeju, Korea (South, 2020, pp. 728–732.
[45] H. Jeon, J. Lee, K. Sohn, Artificial intelligence for traffic signal control based solely on video images, J. Intell. Transp. Syst. 22 (5) (2018) 433–445.
[46] S.S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, IET Intell. Transp. Syst. 11 (7) (2017) 417–423.
[47] W.C. Tchuitcheu, C. Bobda, M.J.H. Pantho, Internet of smart-cameras for traffic lights optimization in smart cities, Internet Things 11 (2020) 100207.
[48] Z. Liang, Y.P. Flötteröd, H. Chen, A. Sohr, X. Bei, M. Bottazzi, J. Trumpold, Real-time microscopic traffic simulation and optimization at intersections with video traffic detection, Int. J. Electr. Electron. Eng. Telecommun. 1 (2018) 30–37.
[49] M.M. Gandhi, D.S. Solanki, R.S. Daptardar, N.S. Baloorkar, Smart control of traffic light using artificial intelligence, in: 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE), 2020, IEEE, Jaipur, India, 2020, pp. 1–6.
[50] B.R. Prathap, K.P. Kumar, C.R. Chowdary, J. Hussain, AI-Based Yolo V4 Intelligent Traffic Light Control System, J. Autom., Mob. Robot. Intell. Syst. 16 (4) (2022) 53–61.
[51] R. Cheng, Z. Qiao, J. Li, J. Huang, Traffic signal timing optimization model based on video surveillance data and snake optimization algorithm, Sensors 23 (11) (2023) 5157.
[52] Z. Fahrunnisa, R. Rahmadwati, R.A. Setyawan, Adaptive traffic light signal control using fuzzy logic based on real-time vehicle detection from video surveillance, J. Ilm. Tek. Elektro Komput. Dan. Inform. 10 (2) (2024) 235–251.
[53] D. Garg, M. Chli, G. Vogiatzis, Fully-autonomous, vision-based traffic signal control: From simulation to reality. in Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2022, pp. 454–462.
[54] K.F. Chu, A.Y.S. Lam, V.O.K. Li, Traffic signal control using end-to-end off-policy deep reinforcement learning, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 7184–7195.
[55] A.A. Salunke, Enhancing urban traffic management through predictive modelling and drone-captured image analysis for smart traffic lights, Int. Res. J. Mod. Eng. Technol. Sci. (2023).
[56] J. Chung, K. Sohn, Image-based learning to measure traffic density using a deep convolutional neural network (CNN), IEEE Trans. Intell. Transp. Syst. 19 (5) (2017) 1670–1675.
[57] S. Zhou, S.T. Ng, Y. Yang, J.F. Xu, Integrating computer vision and traffic modeling for near-real-time signal timing optimization of multiple intersections, Sustain. Cities Soc. 68 (2021) 102775.
[58] Y. Xu, W. Shao, J. Li, K. Yang, W. Wang, H. Huang, C. Lv, H. Wang, SinD: A drone dataset at signalized intersection in china, IEEE25th Int. Conf. Intell. Transp. Syst. (ITSC) 2022 (2022) 2471–2478, https://doi.org/10.1109/ITSC55140.2022.9921959.
[59] H. Gupta, O.P. Verma, Monitoring and surveillance of urban road traffic using low altitude drone images: a deep learning approach, Multimed. Tools Appl. 81 (2022) 19683–19703, https://doi.org/10.1007/s11042-021-11146-x.
[60] A. Sharma, V. Kumar, L. Longchamps, Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species, Smart Agric. Technol. 9 (2024) 100648.
[61] G.P.C.P. da Luz, G.M. Sato, L.F.G. Gonzalez, J.F. Borin, Smart Parking with Pixel-Wise ROI Selection for Vehicle Detection Using YOLOv8, YOLOv9, YOLOv10, and YOLOv11, arXiv preprint arXiv:2412.01983, 2024.
[62] M. Wang, X. Xiong, Y. Kan, C. Xu, M.-O. Pun, UniTSA: A universal reinforcement learning framework for v2x traffic signal control, IEEE Trans. Veh. Technol. (2024) 1–16.
[63] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347, 2017.
[64] W. Genders, S. Razavi, Open-Source Framew. Adapt. Traffic Signal Control. arXiv Prepr. (2019) arXiv:1909.00395,.
[65] C. Gershenson, Self-Organizing Traffic Lights, Complex Syst. 16 (2004) 29–53, https://doi.org/10.25088/ComplexSystems.16.1.29.
[66] S.-B. Cools, C. Gershenson, B. DHooghe, Self-organizing traffic lights: A realistic simulation. Advances in applied self-organizing systems, Springer, 2013, pp. 45–55.
[67] sumo_RL. ⟨https://github.com/MrLeedom/sumo_RL/tree/master⟩. Accessed: 2025-03-20.
[68] P.A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, E. Wießner, Microscopic traffic simulation using sumo. 21st international conference on intelligent transportation systems (ITSC), IEEE, Maui, HI, USA, 2018, pp. 2575–2582.
[69] K. Sanderson, GPT-4 is here: what scientists think, Nature 615 (7954) (2023) 773.
[70] J. Šimsová, Examining Cognitive Abilities and Multilingual Performance of Large Language Models: A Comparative Analysis of GPT-3 and GPT-4. Bachelor thesis, Charles University, 2024.
[71] J.C.M. Than, W.T. Vong, K.S.C. Yong, Comparison of Multi-Modal Large Language Models with Deep Learning Models for Medical Image Classification, in: IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA), 2024, IEEE, 2024, pp. 1–5.
[72] Qwen. ⟨https://huggingface.co/Qwen⟩. Accessed: 2025-04-22.
[73] Qwen2.5. ⟨https://ollama.com/library/qwen2.5:72b⟩. Accessed: 2025-04-22.
[74] QwQ. ⟨https://ollama.com/library/qwq⟩. Accessed: 2025-04-22.
[75] Llama3.3. ⟨https://ollama.com/library/llama3.3⟩. Accessed: 2025-04-22.
[76] meta-llama/Llama-3.3-70B-Instruct. ⟨https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct⟩. Accessed: 2025-04-22.
[77] Deepseek-r1. ⟨https://ollama.com/library/deepseek-r1⟩. Accessed: 2025-04-22.
[78] mistralai/Mistral-7B-v0.1⟨https://huggingface.co/mistralai/Mistral-7B-v0.1⟩. Accessed: 2025-04-22.
[79] Mistral 7B. ⟨https://mistral.ai/news/announcing-mistral-7b⟩. Accessed: 2025-04-22.
[80] D.B. Dangi, B.B. Dangi, O. Steinbock, Evaluation of GPT-4o & GPT-4o-mini's Vision Capabilities for Salt Evaporite Identification, arXiv preprint arXiv:2412.10587, 2024.
[81] Number of Parameters in GPT-4 (Latest Data). https://github.com/cognitivetech/llm-research-summaries/blob/main/models-review/Number-Parameters-in-GPT-4-Latest- Data.mdutm_source=chatgpt.com. Accessed: 2025-04-20.
[82] Transportation Research Board. Highway Capacity Manual 7th Edition-A Guide for Multimodal Mobility Analysis. The national academies of sciences, engineering and medicine. Washington, DC, 2022.
[83] yolov5/data/coco.yaml. ⟨https://github.com/ultralytics/yolov5/blob/master/data/coco.yaml⟩. Accessed: 2025-04-23.