Traffic-R1: Reinforced LLMs Bring Human-Like Reasoning to Traffic Signal Control Systems

Anonymous ACL submission

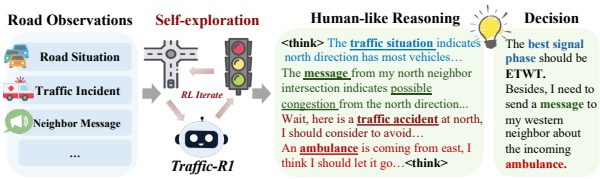
Abstract

We introduce Traffic-R1, a 3B-parameter foundation model with human-like reasoning for Traffic signal control (TSC), developed via self-exploration and iterative reinforcement of LLM with expert guidance in a simulated traffic environment. Compared with traditional reinforcement learning and recent LLM-based methods, Traffic-R1 offers three main advantages: zero-shot generalization, transferring unchanged to new road networks and out-of-distribution incidents by leveraging internal traffic-control policies and reasoning; a compact 3B-parameter design that supports real-time inference on mobile-class chips for edge deployment; and an explainable TSC process that enables multi-intersection coordination through communication and an asynchronous communication network. Extensive benchmarks show Traffic-R1 outperforms strong baselines and training-intensive RL controllers. In production, the model now manages signals affecting over 55,000 drivers daily, reduces average queue lengths by more than 5%, and halves operator workload. We will open source our checkpoint and code to foster further research.

1 Introduction

Rapid urbanization and surging vehicle ownership intensify congestion, wasting billions of productive hours, burning vast fuel reserves, and driving nearly a quarter of urban greenhouse emissions. Prolonged delays raise crash rates, slow emergency response, exacerbate cardiopulmonary pollution, and unfairly burden transit-poor communities. Within this broad societal context, traffic-signal control (TSC), which coordinates phase sequences and durations at signalized intersections, remains a principal lever for mitigating congestion and improving network throughput (Yau et al., 2017; Ye et al., 2019; Wei et al., 2021; Zhang et al., 2024).

Traditional controllers like FixedTime (Koonce et al., 2008) and MaxPressure (Varaiya, 2013) rely on fixed heuristics and thus adapt poorly to fluctuating demand. Reinforcement learning (RL) replaces these hand-crafted rules with a data-driven policy: each cycle observes lane queues, delays, and neighboring signal states, selects a phase (or duration) as the action, and receives a reward linked to delay reduction or throughput gain. Deep RL further augments this paradigm by learning the policy end-to-end with expressive function approximators. These advances achieve impressive benchmark scores in simulation (Srinivasan et al., 2006; Zhao et al., 2011). However, field deployment remains rare as existing methods still suffer from (i) cross-region generalization: policies tuned on one city transfer poorly to another (Zhao et al., 2024; Wei et al., 2021); (ii) interpretability: nontransparent decisions undermine practitioner trust (Vouros, 2022; Glanois et al., 2024); and (iii) robustness to out-of-distribution (OOD) events: models falter during incidents or emergency-vehicle priority scenarios (Glanois et al., 2024; Wei et al., 2018).



Features	Fixed Time	Max Pressure	Traditional RL	LLM as Explainer	LLM as Assistant	Traffic-R1
Zero-shot Deploy	✓	✓	✗	✗	✓	✓
Incident Dealing	✗	✗	✗	✗	✓	✓
Efficiency	✓	✓	✓	✗	✗	✓
Performance	✗	✗	✓	✓	✗	✓
Explainability	✗	✗	✗	✓	✗	✓
Multi-Intersection Coordination	✗	✗	✓	✗	✗	✓

Figure 1: Introduction of Traffic-R1, a foundation (covering six features) reinforced LLM for TSC systems.

Recently, large language models (LLMs) have been enlisted to alleviate these shortcomings. Two integration paradigms dominate. In the **LLM Explainer** paradigm (Lai et al., 2023; Yuan et al., 2025), an LLM is trained to verbalize the policy of an RL controller, translating opaque action choices

into natural language rationales. In contrast, the **LLM assistant** framework (Wang et al., 2025a, 2024) keeps the RL agent in charge of routine control and consults an LLM only when OOD incidents arise. *Both paradigms are promising yet remain distant from large-scale deployment:* LLM Explainers inherit the coverage and performance ceiling of the underlying RL policy, and their post-hoc narratives can diverge from the controller's true internal logic (Wu et al., 2025; Glanois et al., 2024; Wei et al., 2018; Agarwal et al., 2024; Malin et al., 2025); LLM Assistants introduce additional prompt engineering and repeated LLM queries, inflating latency and computation while providing limited benefit for everyday signal timing. Hence practitioners remain cautious about adopting current LLM-enhanced TSC solutions.

To date, operational TSC systems still rely on heuristic rule sets and substantial human oversight to cope with routine flow and unexpected incidents (Mandhare et al., 2018; Müller et al., 2021; Mahavar and Juremalani, 2018). Bridging this research-deployment gap requires a *foundational model* for TSC systems, i.e., a single, versatile agent capable of (1) **zero-shot generalization** to unseen traffic networks and OOD incidents, (2) **resource-efficient inference** on edge hardware such as mobile platforms, and (3) **human-like & transparent reasoning** that supports explainable decision-making and multi-intersection coordination. Figure 1) schematically illustrates these three desiderata in the envisioned foundational agent.

We answer this call with **Traffic-R1**, a lightweight reinforced LLM with human-like reasoning, designed as a foundational traffic controllers incorporating the six key features from Figure 1. Built on the efficient Qwen2.5-3B, Traffic-R1 employs a two-stage agentic RL finetuning approach: an offline stage integrates human expert knowledge from TSC recordings, and an online stage adapts the model to dynamic scenarios through exploration in simulated environments. Inspired by recent studies (Bai et al., 2022; Liu et al., 2024; Yu et al., 2025; Zhou et al., 2023), our RL framework trains Traffic-R1 to generate Chain-of-Thought (CoT) reasoning along with actions via self-iteration. The learning process is guided by a policy-based reward model that incorporates rewards for both output format and action correctness. In the offline stage, action rewards are calculated based on alignment with human expert decisions; in the online stage, they are derived from simulated traffic feedback. This training scheme enables the lightweight model to develop robust reasoning through self-exploration, leading to strong zero-shot TSC performance and generalization to out-of-distribution (OOD) scenarios, all while maintaining resource-efficient inference.

By generating its own training samples within our RL framework, Traffic-R1 mitigates the catastrophic forgetting often observed when smaller models are finetuned on synthetic data from larger ones. As a result, it retains strong general language communication skills and explainable reasoning. o leverage these language skills for multi-agent traffic coordination, we introduce an asynchronous communication network that enables LLM agents to coordinate across multiple intersections via message-passing, mimicking the collaboration of human traffic agents.

In summary, our contributions are:

- **Foundation model for TSC systems**: We present the first LLM-based, general-purpose controller that can operate at any interaction without additional training, handling routine signaling, incident management, and emergency-vehicle prioritization with human-level reasoning.
- **Lightweight yet high-performing**: Our two-stage RL training yields a 3B-parameter LLM that matches or exceeds much larger models (e.g., GPT-4o) and strong expert baselines, while remaining suitable for inference on edge devices.
- **Human-like reasoning and communication**: We achieve human-like reasoning for explainable TSC through self-iteration of the reinforced LLM. This reasoning, combined with language capabilities, is further utilized in our asynchronous communication network for effective coordination across multiple intersections.
- **Extensive validation and field deployment**: Evaluated on standard TSC benchmarks and out-of-distribution incident tasks, achieving stable state-of-the-art performance. In live deployment serving over 55,000 drivers daily, trials show a >5% reduction in average queues and a >50% reduction in operator workload for phase planning and incident response.

2 Related Works

Traffic Signal Control. TSC is essential for traffic management. The FixedTime method, one of the earliest and most widely used approaches, relies on predetermined cycle lengths and phase allocations set by human experts for each intersection (Yau

et al., 2017; Serafini and Ukovich, 1989). It is simple and stable but inefficient under dynamic traffic and requires substantial human effort (Serafini and Ukovich, 1989; Thunig et al., 2019). Max-pressure introduces adaptive control by prioritizing movements based on pressure, defined from queue lengths per direction (Varaiya, 2013; Mercader et al., 2020), yet it and FixedTime remain constrained by fixed rules that struggle with varying scenarios (Rasheed et al., 2020; Wei et al., 2021; Abdulhai et al., 2003). Machine learning has produced RL-based TSC methods such as CoLight (Wei et al., 2019), CosLight (Ruan et al., 2024), and MPLight (Chen et al., 2020), which improve performance in simulation but often fall short for real-world deployment (Chen et al., 2022; Wei et al., 2021; Qadri et al., 2020). Recent work explores integrating large language models into TSC, but significant challenges remain before these approaches are practical for field use (Lai et al., 2023; Yuan et al., 2025; Wang et al., 2025a, 2024; Wen et al., 2023; Zhao et al., 2025; Feng et al., 2025).

RL finetuning for LLM. LLMs often require finetuning to adapt to specific tasks or human preferences (Wu et al., 2025; Minaee et al., 2024; Ding et al., 2025; Li et al., 2025). Instruction finetuning (Chung et al., 2024; Dettmers et al., 2023) is widely used because it is simple and effective for task- and format-specific adaptation. RL finetuning, particularly RL from Human Feedback (Ouyang et al., 2022), has seen less adoption for downstream use due to its complex pipeline and high compute cost(Gao et al., 2023; Bai et al., 2022; Ramamurthy et al., 2022). Recent work shows LLMs can develop reasoning through interaction with RL environments, and Liu et al. (2024) reduces compute needs by using group policy-based optimization. RL finetuning offers practical advantages: (1) self-exploration reduces dependence on large labeled datasets, (2) it guides models to acquire capabilities rather than imitate or memorize outputs, and (3) KL-divergence constraints on updates help mitigate catastrophic forgetting by anchoring the policy to the pretrained distribution(Cao et al., 2024; Xu et al., 2025; Ke et al., 2025).

3 Preliminaries

Definition 3.1. ***Road Network.*** *The road network is modeled as a directed graph with intersections $\mathcal{V}$ and lanes $\mathcal{L}$. Lanes are classified into three types: (1) go-through lanes ($\mathcal{L}_{go}$), (2) left-turn lanes ($\mathcal{L}_{left}$), and (3) right-turn lanes ($\mathcal{L}_{right}$). Each lane connects to neighboring intersections and is divided into segments $S = \{s_1, \ldots, s_n\}$ based on their distance from the intersection.*

Definition 3.2. ***Signal Phase.*** *At each signal-switching time step, the model assigned to an intersection selects a signal phase from a predefined set $\mathcal{A} = \{a_1, \ldots, a_m\}$. A signal phase is defined as $a = set(\mathcal{L}_{allow})$, where $\mathcal{L}_{allow}$ represents the set of lanes permitted to proceed without conflicting movements (i.e., green light for $\mathcal{L}_{allow}$ and red light for conflicting lanes).*

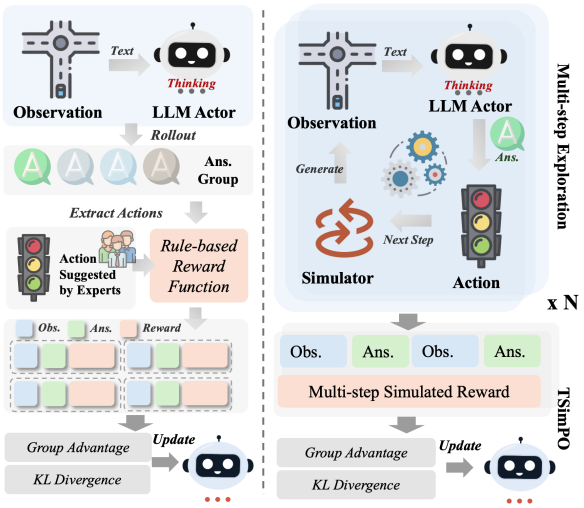
Definition 3.3. ***Traffic Signal Control System.*** *The traffic signal control system comprises multiple agents $\Pi = \{\pi_1, \ldots, \pi_n\}$, each managing signal control at one of n intersections in a road network. Each agent π_i collaborates with neighboring agents through traffic observations and message passing at signal-switching time steps to coordinate multi-intersection operations, such as green wave synchronization and emergency response.*

4 Methodology

In this section, we describe the training pipeline for Traffic-R1. As shown in Figure 2, we utilize a two-stage RL framework comprising offline human-informed RL and online open-world RL to finetune the LLM. For each stage, we define distinct training datasets, environments, policy optimization methods, and reward designs to support their functionality. Besides, we propose an asynchronous communication network for reinforced LLM to support efficient multi-intersection coordination.

4.1 Human-Informed RL

Existing LLMs for TSC are typically finetuned to imitate traditional RL models. This approach has



(a) Offline human-informed RL (b) Online Open-world RL

Figure 2: Introduction of the two-stage RL framework

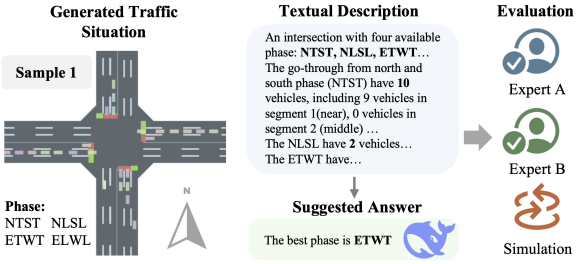


Figure 3: Expert-Collaborative Dataset Construction

two major flaws: first, finetuning LLMs to imitate traditional RL models confines their performance and generalization to the capabilities of the teacher RL model. Second, action trajectories produced by RL models through iterative optimization in simulated environments may not provide clear reasoning and logic for LLMs, as some RL decision policies and actions can be suboptimal or impractical for general TSC, focusing instead on narrow performance metrics. To address these limitations, we propose a human-informed RL finetuning stage that incorporates an expert-collaborative TSC dataset and offline policy optimization. This approach replaces the RL model teacher with real human traffic experts to guide the finetuning of the LLM.

4.1.1 Expert-Collaborative TSC Dataset

Our RL finetuning approach reduces the training data requirement from hundreds of thousands of samples typically needed for instruction finetuning to just thousands. This enables the creation of a TSC dataset with actions provided by human experts for each traffic scenario. We developed a 3,000-sample dataset using a human-in-the-loop pipeline. For each traffic scenario, a base model suggests an action. This action is then validated for effectiveness using the SUMO simulator and reviewed by two human traffic control experts. If an action is rejected, experts provide the correct one. This process, involving 11 experts, resulted in our final dataset. Crucially, we only include the final, validated actions, excluding any reasoning steps. This design encourages the model to develop its own reasoning capabilities during RL finetuning, rather than imitating an external thought process.

4.1.2 Offline Policy Optimization

Inspired by the work of (Liu et al., 2024), which effectively finetunes LLMs on mathematics and coding datasets containing only final answers or actions without reasoning text, our approach uses the same offline RL framework to finetune the LLM on our expert-collaborative TSC dataset to promote self-thinking in traffic control. The core process involves an LLM policy model interacting with an offline environment, guided by a rule-based reward derived from expert-provided answers. Given a textual prompt x describing a specific traffic scenario, the LLM, parameterized by θ, generates an output sequence y autoregressively according to its policy $\pi_\theta(y \mid x) = \prod_{t=1}^{|y|} \pi_\theta(y_t \mid x, y_{<t})$.

Input Template for Rollout. Rollout is a critical component of RL iterations, which involves using the original LLM π_θ to produce a variety of structured responses. To guide the interaction process, the LLM is prompted with specific templates that include format instructions, ensuring the generated sequence y contains both task-specific reasoning and answer components in a structured, extractable format for the offline reward policy. The input prompt template is presented in Appendix A.1.

Offline Rule-based Reward. By treating the offline TSC task as a math-like problem-solving process during our offline RL stage, the rule-based reward is required to be clear and simple, minimizing computational complexity and preventing reward hacking. The reward function R is a weighted combination of an accuracy reward R_{acc} and a format reward R_{format}, defined as: $R(x, y) \in [0, 1] = [w_{\text{acc}}, w_{\text{format}}] \cdot [R_{\text{acc}}, R_{\text{format}}]^T$. Since signal actions are mutually exclusive, $R_{\text{acc}} = 1$ only when the generated action exactly matches the expert-suggested action. Similarly, $R_{\text{format}} = 1$ only when the generated sequence y fully adheres to the specified reasoning and answer format instructions.

Reinforcing Reasoning via Policy Optimization. To optimize the policy π_θ with parameters θ in an offline environment, we employ Group Relative Policy Optimization (GRPO) (Liu et al., 2024) to ensure stable gradient updates. Let π_θ represent the policy and $\{a_i^{(j)}\}_{j=1}^k$ denote each input prompt x_i is paired with k candidate completions sampled from the current policy. The reward function R assigns a score $r_i^{(j)}$ to each completion $a_i^{(j)}$. To address high variance in policy gradients, GRPO computes group-normalized advantages for each completion $a_i^{(j)}$ generated from the same input x_i, as shown in Equation 1. This approach centers the rewards within each group, mitigating the impact of absolute reward magnitudes:

$$A_i^{(j)} = r_i^{(j)} - \frac{1}{k} \sum_{l=1}^{k} r_i^{(l)}, \tag{1}$$

The policy is updated by maximizing the clipped surrogate objective:

$$\mathcal{L}(\theta) = \mathbb{E}\Bigg[\min\Bigg(\rho_i^{(j)} A_i^{(j)}, \text{clip}(\rho_i^{(j)}, 1-\epsilon, 1+\epsilon) * A_i^{(j)}\Bigg) - \beta \mathbb{D}_{\text{KL}}\left[\pi_\theta(\cdot|x) \parallel \pi_{\theta_{\text{ref}}}(\cdot|x)\right]\Bigg], \tag{2}$$

where $\rho_i^{(j)} = \frac{\pi_\theta(a_i^{(j)}|x_i)}{\pi_{\theta_{\text{ref}}}(a_i^{(j)}|x_i)}$ is the likelihood ratio between the current policy π_θ and the reference policy $\pi_{\theta_{\text{ref}}}$, and ϵ is the clipping threshold. The expectation $\mathbb{E}[\cdot]$ is computed over $(x_i, a_i^{(j)})$ drawn from $\pi_{\theta_{\text{ref}}}$.The coefficient β determines the strength of the Kullback-Leibler divergence penalty $\mathbb{D}_{\text{KL}}\left[\pi_\theta \parallel \pi_{\theta_{\text{ref}}}\right]$. In practice, reference policy $\pi_{\theta_{\text{ref}}}$ is typically set to a snapshot of the previous policy, which stabilizes training to inspiring deep thinking instead of imitation by constraining policy updates.

4.2 Open-World Reinforcement Learning

Although the human-informed offline RL has fine-tuned LLM to learn from human experts for stable performance on TSC tasks, the model's capacity is limited to the expert knowledge extracted from the dataset. In this section, we propose an open-world online RL to inspire LLM explore multi-step and multi-intersection TSC networks. This approach allows LLM to interact with the online dynamic simulated traffic environment and update its policy based on online reward for better performance.

Online Traffic Simulation To simulate the multi-intersection and multi-step dynamics of real-world traffic flow, we constructed a 4×4 simulated road network with 300-meter roads between each intersection. The 16 positions in the network represent most typical road scenarios encountered at real-world intersections. Traffic flow within the network is randomly generated, allowing for up to 8,000 vehicles over the course of one hour. For efficiency in iterations, we utilize CityFlow (Zhang et al., 2019) as the simulator to model the traffic dynamics resulting from the actions of the LLM. The online multi-step rewards R_{traj} are quantified based on the cumulative average queue length and waiting time caused by a series of multi-intersection actions. We use group advantage as final rewards to mitigate random variations in environment during online training through group mean comparison, ensuring stable optimization gradients.

Multi-step Policy Optimization Existing RL fine-tuning typically uses offline, single-turn settings (e.g., math). Traffic online RL requires LLMs to act in interactive, multi-step environments with stochastic feedback. Methods (Wang et al., 2025b) that concatenate observations and model responses into a single trajectory with a trajectory-level reward are impractical for TSC: context lengths become very large, raising compute and memory costs and diluting token-level attention on critical observations and actions. Besides, traffic is also partially observable and not well modeled as a perfect Markov decision process, so directly porting trajectory-concatenation approaches from digital games will likely produce brittle policies that generalize poorly and offer limited interpretability, similar to traditional RL controllers.

To address the chaotic dynamics in real-world traffic systems, where the link between actions and subsequent states is not strictly continuous, we propose *Stepwise Trajectory Policy Optimization* (STPO). This method assigns the total trajectory reward to each individual observation-action pair (o_t, a_t), decomposing the trajectory to provide denser, step-level reward signals and reduce computational overhead. The step-wise reward r_t is uniformly distributed from the total trajectory reward R_{traj} over T steps:

$$r_t = R_{\text{traj}}(o_{1:T}, a_{1:T}) \cdot T^{-1}, \tag{3}$$

where R_{traj} is the reward over the full sequence of observations $o_{1:T}$ and actions $a_{1:T}$. The policy $\pi_\theta(a_t|o_t)$ is then optimized with the objective:

$$\mathcal{L}_{\text{STPO}}(\theta) = \mathbb{E}_{(o_t, a_t \sim \pi_{\theta_{\text{ref}}})}\left[\log \pi_\theta(a_t|o_t) A_t\right], \tag{4}$$

The advantage calculation (A_t), Kullback-Leibler divergence penalty, and advantage clipping are analogous to those in the original GRPO formulation (see Equation 2).

4.3 Asynchronous Communication Network

Most TSC research assumes a synchronized parallel workflow, where all intersection agents act simultaneously. This simplification hinders real-world multi-intersection coordination, which is inherently asynchronous. Consequently, implementations in synchronized frameworks are often inefficient, requiring extensive shared global data, or ineffective, leading to incompatible decisions. While some reinforcement learning methods incorporate neighborhood observations, they typically suffer from poor generalization and scalability.

To enable natural communication and unlock the potential of our LLM-based model, we propose an asynchronous communication network. As illustrated in Figure 4, we partition intersections into two groups based on positional parity. Each TSC step is split into two inner steps: the first activates group one, and the second activates group

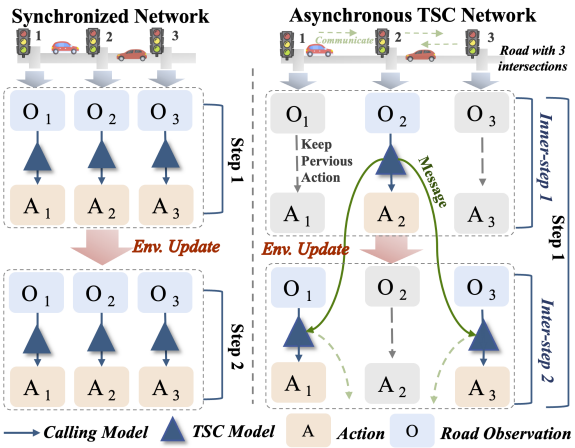


Figure 4: Asynchronous communication network compared with conventional synchronized network

two. This structure allows models activated in the first inner step to generate messages for their neighbors, which are then received and processed in the subsequent inner step. This framework fully utilizes the language capabilities of LLMs to enhance multi-intersection coordination. The pseudocode for this network is provided in Algorithm 1.

5 Experiments

In this section, we evaluate our proposed Traffic-R1 to address the following research questions:

- **RQ1**: Can Traffic-R1 outperform other TSC expert models and LLMs on public datasets and in zero-shot settings?
- **RQ2**: How does Traffic-R1 perform in handling OOD incidents through its reasoning?
- **RQ3**: What's the advantage of our RL-based finetuning over traditional paradigms for LLMs in traffic control tasks?
- **RQ4**: How effective are the designs of Traffic-R1 under various ablation settings?

5.1 Experimental Settings

5.1.1 Dataset

Our experiments were primarily conducted on two public traffic flow datasets (Mei et al., 2024) to ensure fair comparison, as detailed in Table 1. For out-of-distribution scenarios, we collect traffic emergency incident recordings from traffic management departments and summarize them into 200 representative textual examples, such as passages running onto roads, vehicle accidents, and school times, along with the action records implemented by traffic managers as correct responses (presented in Appendix A.3). Besides, for , we modify the Hangzhou1 datasets by incorporating a 5% proportion of emergency vehicle flow to simulate emergency vehicle coordination scenarios.

Table 1: Statistics of traffic flow datasets.

Dataset	Structure	Vehicles	Arrival rate (vehicles/5min)			
			Mean	Std	Max	Min
Jinan1	3×4	6295	523.67	98.52	671	255
Jinan2		4365	362.83	74.81	493	236
Jinan3		5494	456.92	160.87	569	362
Hangzhou1	4×4	2983	247.68	40.44	332	211
Hangzhou2		6984	581.08	318.43	1145	202

5.1.2 Implementation Details

Traffic-R1 was trained based on Qwen2.5-3B-base on a device with 4 H100 GPUs and deployed for inference on a single Tesla T40 GPU, utilizing the Verl framework (Sheng et al., 2024). We evaluated all models using the CityFlow traffic simulator (Zhang et al., 2019). The experimental setup features a standard four-phase action space: north-south through (NTST), east-west through (ETWT), east-west left-turn (ELWL), and north-south left-turn (NLSL). Each green phase lasts 15 seconds, followed by a 3-second yellow and a 2-second red transition period, consistent with real-world TSC systems (Zhang et al., 2022; Mei et al., 2024). Right turns are permitted at all times. All traffic flow datasets are simulated for one-hour periods.

5.1.3 Baseline Methods

We incorporate a range of baseline models from various research areas to ensure a comprehensive comparison. For traditional TSC methods, we include FixedTime (Koonce et al., 2008) and Maxpressure (Varaiya, 2013). For RL-based methods, we evaluate five effective approaches: MPLight (Chen et al., 2020), AttendLight (Oroojlooy et al., 2020), CoLight (Wei et al., 2019), Efficient-CoLight (Wu et al., 2021), and Advanced-CoLight (Zhang et al., 2022), along with the state-of-the-art LLM-based method, LLMLight (Lai et al., 2023). For zero-shot methods, we assess the performance of CoLLMLight (Yuan et al., 2025) and general LLM models, which include Llama 3.3 (70B), Qwen 2.5 (72B), GPT 3.5-turbo, GPT-4o, and DeepSeek-R1 (671B and distilled to 7B). All learning-based baselines are trained on the same 4x4 simulated road network and traffic flow dataset as Traffic-R1 during the open-world RL stage. Notably, for LLMLight and CoLLMLight, we also incorporate our expert-collaborative dataset into the training instructions to ensure a fair comparison

5.1.4 Evaluation Protocols

We adopt the commonly used Average Travel Time (ATT) and Average Waiting Time (AWT) to evaluate the performance of models. Lower values in ATT and AWT indicate better traffic efficiency.

Table 2: Zero-shot performance comparison on conventional traffic signal control tasks (the smaller the better). The best results are in **bold** and second-best results are underlined.

Models	Jinan1		Jinan2		Jinan3		Hangzhou1		Hangzhou2		Paradigm
	ATT↓	AWT↓	ATT↓	AWT↓	ATT↓	AWT↓	ATT↓	AWT↓	ATT↓	AWT↓	
FixedTime	453.41	51.32	370.34	35.15	384.53	36.95	497.54	36.41	408.53	53.94	*Traditional Methods*
Maxpressure	274.34	32.04	246.35	22.56	245.66	24.31	289.55	21.52	349.53	67.52	
LLMLight-7B [KDD'24]	274.47	33.66	286.53	28.66	271.11	28.27	299.31	25.53	331.38	51.79	*RL-based Methods*
MPLight [AAAI'20]	455.34	72.45	471.14	78.03	427.37	64.91	491.32	64.05	425.42	69.85	
AttendLight [NeurIPS'20]	381.11	67.59	305.53	64.72	331.34	66.42	318.94	67.84	348.41	65.58	
CoLight [CIKM'19]	472.44	91.09	450.41	78.59	498.84	89.94	494.61	72.18	435.32	81.11	
Efficient-Colight [Arxiv'21]	663.16	98.98	640.34	91.32	638.23	80.34	701.45	103.43	534.94	87.19	
Advanced-CoLight [ICML'22]	347.31	56.54	345.78	35.96	342.56	37.55	485.32	54.11	523.19	72.56	
CoLLMLight-8B [Arxiv'25]	281.12	33.23	269.34	25.51	268.32	34.36	298.42	24.45	336.92	45.43	*Zero-shot Methods*
Llama3.3-70B [Meta'24]	272.41	33.53	244.55	22.04	243.53	25.43	281.44	17.65	326.42	45.56	
Qwen 2.5-72B [Alibaba'24]	275.42	33.15	251.41	25.49	264.21	24.54	282.13	17.54	329.34	39.34	
GPT 3.5-turbo [OpenAI'23]	337.32	39.98	328.19	37.08	343.19	34.35	293.42	23.45	348.59	33.45	
GPT-4o [OpenAI'24]	281.58	30.11	259.61	24.71	258.85	24.17	280.48	16.32	325.48	32.26	
DeepSeek-R1-671B [DeepSeek'25]	279.11	31.85	258.43	21.67	262.21	27.87	278.565	17.81	335.53	**30.19**	
DeepSeek-R1-Distill-7B [DeepSeek'25]	331.45	38.91	311.43	31.43	288.42	29.23	291.32	19.56	344.73	33.72	
Traffic-R1-3B (Ours)	**270.34**	**27.95**	**239.53**	**21.11**	**238.03**	**23.17**	**277.83**	**15.51**	**324.11**	33.14	

5.2 Conventional TSC Evaluation (RQ1)

We evaluate the performance of Traffic-R1 on conventional TSC tasks using public datasets that are widely adopted in TSC research. All learning-based methods are trained in the same simulated traffic environment as Traffic-R1 to fairly assessment. As shown in Table 2, our model significantly outperforms all baselines, demonstrating strong generalization for real-world deployment. RL-based methods perform poorly in zero-shot scenarios, even lagging behind traditional methods, which questions their real-world applicability without iterative training. While large LLMs like DeepSeek-R1-671B achieve impressive zero-shot results, their performance degrades when distilled into smaller models. For completeness, Appendix A.4 presents full-shot results, where RL models are trained on each test dataset. Although RL methods then surpass traditional ones, our zero-shot model still outperforms them, highlighting its superior internal reasoning and control policies.

5.3 OOD Incident Dealing (RQ2)

Evaluation only in ideal simulations is insufficient for deployment. RQ2 asks whether Traffic-R1 can handle out-of-distribution (OOD) traffic incidents via human-like reasoning. Incident handling requires internal logic, commonsense, and domain knowledge, which challenge both traditional RL and standard LLM approaches.

We split OOD tasks into two types: **local intersection incidents** (single-intersection events, e.g., an local accident) and **network-wide incidents** (events affecting multiple intersections, e.g., emergency-vehicle routing). For local incidents we use Emergency Action Accuracy (EAA). For network-wide incidents we use Average Emergency Travel Time (AETT) and Average Emergency Waiting Time (AEWT) adapted from Wang et al. (2024). Results are reported in Table 3. The findings highlight two key strengths of Traffic-R1:

- **Stable OOD generalization.** Traffic-R1 maintains consistent performance across diverse OOD scenarios and outperforms larger general LLMs and LLMLight by over 30% across metrics, indicating it applies traffic knowledge instead of merely imitating training examples.
- **Lightweight and efficient.** At 3B parameters, Traffic-R1 matches or exceeds traditional baselines and advanced LLMs while using roughly 1% of their parameters and much lower deployment cost. It performs particularly well on network-wide incidents, showing that the asynchronous communication design enables coordinated multi-intersection control.

5.4 Discussion of RL-based Finetuning (RQ3)

To validate our proposed RL-based two-stage finetuning paradigm, we compare it against traditional instruction finetuning, represented by the LLMLight framework (Lai et al., 2023). For a fair comparison, both methods use the same Qwen2.5 base model. We built the instruction dataset for LLMLight using a similar pipeline: RL-generated action trajectories, GPT-4 explanations, and expert-collaborative QA samples. After training, we evaluate the models in two dimensions: *1. TSC perfor-*

Table 3: OOD Incident Dealing Evaluation. "-" indicates the method is entirely inadequate for the task.

Method	Parameter Size	Local	Network-wide	
		EAA↑	AETT↓	AEWT↓
Random	n.a.	0.25	614.45	97.42
MaxPressure	n.a.	-	287.94	21.87
Advanced-CoLight	n.a.	-	286.32	24.53
LALight	72B	0.82	234.42	12.32
LLMLight	7B	0.42	273.55	15.21
Qwen2.5 (large)	72B	0.88	232.54	10.53
DeepSeek-R1	672B	**0.93**	223.19	10.14
Traffic-R1	3B	0.85	**215.58**	**7.98**

mance: we assess zero-shot capabilities, quantifying results based on the average improvement in ATT and AWT across all datasets, with Qwen2.5-3B's performance normalized to 0.5. *2. General capabilities:* we use public benchmarks (Team, 2024) to evaluate reasoning, instruction following, and commonsense to gauge the models' stability and potential for real-world deployment. The results in Figure 5, highlight two key findings regarding RL finetuning for LLMs in traffic tasks:

- **From imitation to reasoning.** Traffic-R1 significantly outperforms the instruction-finetuned model on zero-shot TSC tasks. This performance gap, despite identical training data, shows that instruction finetuning encourages imitation of synthetic data. In contrast, our RL-based paradigm fosters internal reasoning through self-exploration and iteration.
- **Complete general capabilities.** Instruction finetuning often causes "catastrophic forgetting," degrading LLM's general abilities. The SFT model exemplifies this, performing worse than the base model on general benchmarks. Traffic-R1 mitigates this issue using self-rollout samples and KL divergence penalty. This approach constrains the policy update to a controlled space, promoting deeper optimization over memorization and preserving the model's comprehensive capabilities.

5.5 Ablation Study (RQ4)

To assess the contribution of each component of Traffic-R1 to its performance, we developed the following model variants for our ablation study:

- **(-) Expert.** This variant excludes the human-informed RL stage and is trained solely using open-world exploration.
- **(-) Open-world.** This variant excludes the open-world RL stage during training.
- **(-) Communicate.** This variant removes the asynchronous communication mechanism and operates without communication.

We present the ablation results for Traffic-R1 and its variants on both conventional traffic scenarios and OOD scenarios in Figure 6. Our findings are summarized as follows: 1) The human-informed RL stage is necessary to establish foundational TSC knowledge, enabling stable exploration in the subsequent open-world RL stage. 2) The open-world RL stage is effective in unlocking the model's potential to achieve superior performance. 3) Asynchronous communication is critical for multi-intersection coordination tasks but is not determinative for conventional traffic scenarios.

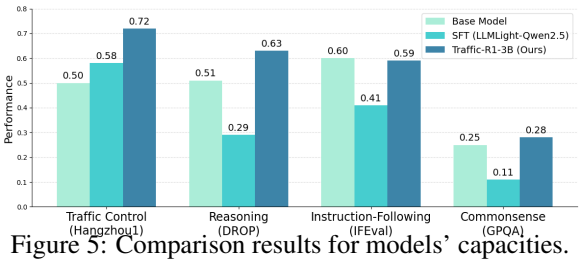


Figure 5: Comparison results for models' capacities.

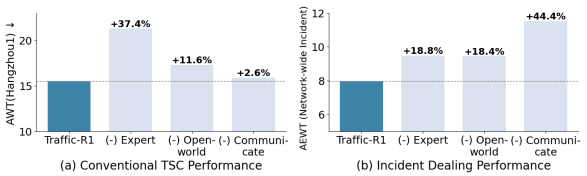


Figure 6: Ablation results on TSC and OOD tasks.

6 Real-world deployment

Figure 7: In-suit traffic sensing devices for Traffic-R1

As shown in Figure 7, Traffic-R1 is deployed on a online traffic platform in a Chinese city (name withheld for privacy), controlling 10 key intersections and serving over 55,000 drivers daily. A/B tests for 6 weeks (Table 6) show significant improvements in real-world traffic control—more efficient decision-making, superior performance compared with human experts, and consistent operation with reduced labor. We place details and discussion in Appendix A.6 due to space limit.

Table 4: A/B test results spanning 6 weeks.

Method	Average Queue↓	Maximum Queue↓	Working Hours↓
Manual	34.5	50.3	2+
Traffic-R1	31.3	48.1	0.5+
#Improve	**9.3%**	**4.4%**	**∼75%**

7 Conclusion and Future Work

In this paper, we introduce a two-stage reinforcement learning strategy and a network communication framework to convert the LLM into a foundational traffic-control model that operates like a human traffic agent. Evaluation on standard TSC benchmarks and traffic-incident handling shows improvement over prior methods, and real-world tests demonstrate value for industrial deployment. Future work includes reinforced VLMs for direct road-vision reasoning without reliance on structured textual traffic data, enhancing efficiency and deployment convenience.

8 Limitations

While Traffic-R1 demonstrates strong performance in both standard and out-of-distribution traffic control tasks with impressive zero-shot generalization, it has several limitations. First, the base model, training, and inference are highly reliant on structured data in a single textual modality: traffic observations and incident information must be converted into structured text before being input to the model. This requirement causes inconvenience and potential information loss in dense real-world deployments, and we will address it in future work by developing reinforced vision-language models and multimodal input pipelines. Second, due to limitations in deployment scale and practical constraints imposed by traffic management agencies, we cannot provide a detailed analysis of LLM hallucination-related instability across large-scale traffic scenarios or assess drivers' feedback to the agent-based traffic network. Access restrictions, privacy requirements, safety constraints, and the cost and complexity of long-term field trials prevented extended real-world evaluations and systematic collection of driver response data.

References

Baher Abdulhai, Rob Pringle, and Grigoris J Karakoulas. 2003. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3):278–285.

Chirag Agarwal, Sree Harsha Tanneru, and Himabindu Lakkaraju. 2024. Faithfulness vs. plausibility: On the (un) reliability of explanations from large language models. *arXiv preprint arXiv:2402.04614*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2024. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*.

Chacha Chen, Hua Wei, Nan Xu, Guanjie Zheng, Ming Yang, Yuanhao Xiong, Kai Xu, and Zhenhui Li. 2020. Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3414–3421.

Rex Chen, Fei Fang, and Norman Sadeh. 2022. The real deal: A review of challenges and opportunities in moving reinforcement learning-based traffic signal control systems towards reality. *arXiv preprint arXiv:2206.11996*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, and 1 others. 2025. Understanding world or predicting future? a comprehensive survey of world models. *ACM Computing Surveys*, 58(3):1–38.

Jie Feng, Jun Zhang, Tianhui Liu, Xin Zhang, Tianjian Ouyang, Junbo Yan, Yuwei Du, Siqi Guo, and Yong Li. 2025. Citybench: Evaluating the capabilities of large language models for urban tasks. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 5413–5424.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Claire Glanois, Paul Weng, Matthieu Zimmer, Dong Li, Tianpei Yang, Jianye Hao, and Wulong Liu. 2024. A survey on interpretable reinforcement learning. *Machine Learning*, 113(8):5847–5890.

Zixuan Ke, Fangkai Jiao, Yifei Ming, Xuan-Phi Nguyen, Austin Xu, Do Xuan Long, Minzhi Li, Chengwei Qin, Peifeng Wang, Silvio Savarese, and 1 others. 2025. A survey of frontiers in llm reasoning: Inference scaling, learning to reason, and agentic systems. *arXiv preprint arXiv:2504.09037*.

Peter Koonce and 1 others. 2008. Traffic signal timing manual. Technical report, United States. Federal Highway Administration.

Siqi Lai, Zhao Xu, Weijia Zhang, Hao Liu, and Hui Xiong. 2023. Llmlight: Large language models as traffic signal control agents. *arXiv preprint arXiv:2312.16044*.

Zhonghang Li, Lianghao Xia, Xubin Ren, Jiabin Tang, Tianyi Chen, Yong Xu, and Chao Huang. 2025. Urban computing in the era of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Vaishali Mahavar and Jayesh Juremalani. 2018. Literature review on traffic control systems used worldwide. *Journal of Emerging Technologies and Innovative Research*, 5(5):77–79.

Ben Malin, Tatiana Kalganova, and Nikolaos Boulgouris. 2025. A review of faithfulness metrics for hallucination assessment in large language models. *IEEE Journal of Selected Topics in Signal Processing*.

Pallavi A Mandhare, Vilas Kharat, and CY Patil. 2018. Intelligent road traffic control system for traffic congestion: a perspective. *International Journal of Computer Sciences and Engineering*, 6(07):2018.

Hao Mei, Xiaoliang Lei, Longchao Da, Bin Shi, and Hua Wei. 2024. Libsignal: An open library for traffic signal control. *Machine Learning*, 113(8):5235–5271.

Pedro Mercader, Wasim Uwayid, and Jack Haddad. 2020. Max-pressure traffic controller based on travel times: An experimental analysis. *Transportation Research Part C: Emerging Technologies*, 110:275–290.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Arthur Müller, Vishal Rangras, Tobias Ferfers, Florian Hufen, Lukas Schreckenberg, Jürgen Jasperneite, Georg Schnittker, Michael Waldmann, Maxim Friesen, and Marco Wiering. 2021. Towards real-world deployment of reinforcement learning for traffic signal control. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 507–514. IEEE.

Afshin Oroojlooy, Mohammadreza Nazari, Davood Hajinezhad, and Jorge Silva. 2020. Attendlight: Universal attention-based reinforcement learning model for traffic signal control. *Advances in Neural Information Processing Systems*, 33:4079–4090.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Syed Shah Sultan Mohiuddin Qadri, Mahmut Ali Gökçe, and Erdinç Öner. 2020. State-of-art review of traffic signal control methods: challenges and opportunities. *European transport research review*, 12(1):55.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.

Faizan Rasheed, Kok-Lim Alvin Yau, Rafidah Md Noor, Celimuge Wu, and Yeh-Ching Low. 2020. Deep reinforcement learning for traffic signal control: A review. *IEEE Access*, 8:208016–208044.

Jingqing Ruan, Ziyue Li, Hua Wei, Haoyuan Jiang, Jiaming Lu, Xuantang Xiong, Hangyu Mao, and Rui Zhao. 2024. Coslight: Co-optimizing collaborator selection and decision-making to enhance traffic signal control. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2500–2511.

Paolo Serafini and Walter Ukovich. 1989. A mathematical model for the fixed-time traffic control problem. *European Journal of Operational Research*, 42(2):152–165.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Dipti Srinivasan, Min Chee Choy, and Ruey Long Cheu. 2006. Neural networks for real-time traffic signal control. *IEEE Transactions on intelligent transportation systems*, 7(3):261–272.

ModelScope Team. 2024. EvalScope: Evaluation framework for large models.

Theresa Thunig, Robert Scheffler, Martin Strehler, and Kai Nagel. 2019. Optimization and simulation of fixed-time traffic signal control in real-world applications. *Procedia Computer Science*, 151:826–833.

Pravin Varaiya. 2013. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195.

George A Vouros. 2022. Explainable deep reinforcement learning: state of the art and challenges. *ACM Computing Surveys*, 55(5):1–39.

Maonan Wang, Yirong Chen, Aoyu Pang, Yuxin Cai, Chung Shue Chen, Yuheng Kan, and Man-On Pun. 2025a. Vlmlight: Traffic signal control via vision-language meta-control and dual-branch reasoning. *arXiv preprint arXiv:2505.19486*.

Maonan Wang, Aoyu Pang, Yuheng Kan, Man-On Pun, Chung Shue Chen, and Bo Huang. 2024. Llm-assisted light: Leveraging large language model capabilities for human-mimetic traffic signal control in complex urban environments. *arXiv preprint arXiv:2403.08337*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb,

Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025b. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *Preprint*, arXiv:2504.20073.

Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. 2019. Colight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1913–1922.

Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2021. Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD explorations newsletter*, 22(2):12–18.

Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2496–2505.

Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. 2023. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*.

Qiang Wu, Liang Zhang, Jun Shen, Linyuan Lü, Bo Du, and Jianqing Wu. 2021. Efficient pressure: Improving efficiency for signalized intersections. *arXiv preprint arXiv:2112.02336*.

Xiao-Kun Wu, Min Chen, Wanyi Li, Rui Wang, Limeng Lu, Jia Liu, Kai Hwang, Yixue Hao, Yanru Pan, Qingguo Meng, and 1 others. 2025. Llm fine-tuning: Concepts, opportunities, and challenges. *Big Data and Cognitive Computing*, 9(4):87.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Kok-Lim Alvin Yau, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk. 2017. A survey on reinforcement learning models and algorithms for traffic signal control. *ACM Computing Surveys (CSUR)*, 50(3):1–38.

Bao-Lin Ye, Weimin Wu, Keyu Ruan, Lingxi Li, Tehuan Chen, Huimin Gao, and Yaobin Chen. 2019. A survey of model predictive control methods for traffic signal control. *IEEE/CAA Journal of Automatica Sinica*, 6(3):623–640.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zirui Yuan, Siqi Lai, and Hao Liu. 2025. Collmlight: Cooperative large language model agents for network-wide traffic signal control. *arXiv preprint arXiv:2503.11739*.

Huichu Zhang, Siyuan Feng, Chang Liu, Yaoyao Ding, Yichen Zhu, Zihan Zhou, Weinan Zhang, Yong Yu, Haiming Jin, and Zhenhui Li. 2019. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The world wide web conference*, pages 3620–3624.

Liang Zhang, Qiang Wu, Jun Shen, Linyuan Lü, Bo Du, and Jianqing Wu. 2022. Expression might be enough: Representing pressure and demand for reinforcement learning based traffic signal control. In *International Conference on Machine Learning*, pages 26645–26654. PMLR.

Siyao Zhang, Daocheng Fu, Wenzhe Liang, Zhao Zhang, Bin Yu, Pinlong Cai, and Baozhen Yao. 2024. Trafficgpt: Viewing, processing and interacting with traffic foundation models. *Transport Policy*, 150:95–105.

Baining Zhao, Jianjie Fang, Zichao Dai, Ziyou Wang, Jirong Zha, Weichen Zhang, Chen Gao, Yue Wang, Jinqiang Cui, Xinlei Chen, and Yong Li. 2025. UrbanVideo-bench: Benchmarking vision-language models on embodied intelligence with video data in urban spaces. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32400–32423, Vienna, Austria. Association for Computational Linguistics.

Dongbin Zhao, Yujie Dai, and Zhen Zhang. 2011. Computational intelligence in urban traffic signal control: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):485–494.

Guoyang Zhao, Fulong Ma, Weiqing Qi, Chenguang Zhang, Yuxuan Liu, Ming Liu, and Jun Ma. 2024. Tsclip: Robust clip fine-tuning for worldwide cross-regional traffic sign recognition. *arXiv preprint arXiv:2409.15077*.

X Zhou, M Liu, BL Zagar, E Yurtsever, and AC Knoll. 2023. Vision language models in autonomous driving and intelligent transportation systems. arxiv 2023. *arXiv preprint arXiv:2310.14414*.

A Appendix

A.1 Input Template for Traffic-R1

Input Templates for LLM Rollout

System: You are a helpful traffic control agent.
Task Description: The crossroad connects two roads: north-south and east-west, with the traffic light at their intersection. Each road is divided into two sections (e.g., north and south for the north-south road) and each section has two lanes: a through lane and a left-turn lane...
Structured Traffic Observation:
Signal: ETWT
Allowed lanes: Eastern and western through lanes
- Early queued: 2 (East), 1 (West), 3 (Total)
- Segment 1: 0 (East), 0 (West), 0 (Total)
- Segment 2: 1 (East), 0 (West), 1 (Total)
- Segment 3: 1 (East), 2 (West), 3 (Total)
Signal: ELWL
Allowed lanes: Eastern and western left lanes
- Early queued: 0 (East), 1 (West), 1 (Total)
- Segment 1: 0 (East), 0 (West), 0 (Total)
- Segment 2: 3 (East), 0 (West), 3 (Total)
- Segment 3: 1 (East), 0 (West), 1 (Total)
Signal: NTST
Allowed lanes: North and south through lanes
- Early queued: 0 (North), 0 (South), 0 (Total)
- Segment 1: 0 (North), 0 (South), 0 (Total)
- Segment 2: 1 (North), 0 (South), 1 (Total)
- Segment 3: 1 (East), 1 (West), 2 (Total)
Signal: NLSL
Allowed lanes: North and south left lanes
- Early queued: 1 (North), 0 (South), 1 (Total)
- Segment 1: 0 (North), 0 (South), 0 (Total)
- Segment 2: 0 (North), 0 (South),0 (Total)
- Segment 3: 1 (East), 0 (West), 1 (Total)
Incident Information (optional):
(at training stage we do not use this part)
Refer Appendix A.3...
Format Instruction: You can only choose one of the signals listed above. You FIRST think about the reasoning process for your choice as an internal monologue and then provide the final answer. Your think process MUST BE put in <think>...</think> tags. The final choice MUST BE put in \boxed{ }.

A.2 Introduction of baselines

We compare our method against three categories of approaches for traffic control. Below is detailed information about these methods:

- **Traditional Methods:** This category includes conventional traffic signal control (TSC) methods, which are straightforward and widely adopted in real-world traffic systems.
  - **FixedTime (Koonce et al., 2008):** A policy that assigns a fixed cycle length with predefined phase splits across all phases.
  - **MaxPressure (Varaiya, 2013):** A control strategy that selects the phase with the highest pressure to optimize traffic flow.
- **RL-based Methods:** These methods normally require training and interaction with their policies on each evaluation dataset.
  - **LLMLight-7B (Lai et al., 2023):** A SOTA LLM-based TSC method that employs the Advanced-CoLight framework to interact and generate action policies for each dataset. It utilizes GPT-4 to generate explanations for each action, which, along with Advanced-CoLight generated actions, are used for instruction fine-tuning to enable the LLM base model to emulate TSC capabilities.
  - **MPLight (Chen et al., 2020):** A method based on the FRAP model that uses pressure as both observation and reward to optimize TSC.
  - **AttendLight (Oroojlooy et al., 2020):** A method that employs attention mechanism to construct phase features and predict its transition probabilities.
  - **CoLight (Wei et al., 2019):** A method that uses a graph attention network to represent inter-intersection communication within a RL framework.
  - **Efficient-CoLight (Wu et al., 2021):** An enhanced version of the CoLight model that incorporates efficient pressure as an observation to improve decision-making in TSC.
  - **Advanced-CoLight (Zhang et al., 2022):** A SOTA RL-based method that enhances CoLight by integrating efficient pressure and advanced traffic state features, such as effective running vehicles, to optimize decision-making capabilities.
- **Zero-shot Methods:** These methods (including Traffic-R1) are represented by their zero-shot working ability on every dataset. In our experiment, the models are not trained on any TSC evaluation dataset and carry out traffic control based on their inner policies and knowledge.
  - **CoLLMLight-8B (Yuan et al., 2025):** An enhanced version of LLMLight (Lai et al., 2023) that incorporates neighbor information into consideration. The model is trained using an instruction fine-tuning framework on a synthetic dataset and demonstrates improved zero-shot performance compared to LLMLight.
  - **Llama3.3-70B:** A LLM developed by Meta AI, featuring 70 billion parameters and a 128K

Table 5: Performance comparison on conventional traffic signal control tasks (the smaller the better). The best results are in **bold** and second-best results are underlined.

Models	Jinan1		Jinan2		Jinan3		Hangzhou1		Hangzhou2		Paradigm
	ATT	AWT	ATT	AWT	ATT	AWT	ATT	AWT	ATT	AWT	
FixedTime	453.41	51.32	370.34	35.15	384.53	36.95	497.54	36.41	408.53	53.94	*Traditional Methods*
Maxpressure	274.34	32.04	246.35	22.56	245.66	24.31	289.55	21.52	349.53	67.52	
LLMLight-7B	274.47	33.66	256.53	28.66	247.11	28.27	289.31	25.53	331.38	51.79	*RL-based Methods*
MPLight	310.54	50.45	270.14	48.03	272.37	42.91	319.32	44.05	365.42	69.85	
AttendLight	280.11	47.59	250.53	34.72	251.34	36.42	288.94	27.84	338.41	55.58	
CoLight	272.44	41.09	250.41	38.59	248.84	39.94	294.61	42.18	335.32	61.11	
Efficient-Colight	263.16	28.98	240.34	21.32	238.23	**20.34**	301.45	33.43	334.94	47.19	
Advanced-CoLight	**247.31**	32.54	**235.78**	25.96	242.56	27.55	285.32	24.11	323.19	52.56	
Llama3.3-70B	272.41	33.53	244.55	22.04	243.53	25.43	281.44	17.65	326.42	45.56	*Zero-shot Methods*
Qwen 2.5-72B	275.42	33.15	251.41	25.49	264.21	24.54	282.13	17.54	329.34	39.34	
GPT 3.5-turbo	337.32	39.98	328.19	37.08	343.19	34.35	293.42	23.45	348.59	33.45	
GPT-4o	281.58	30.11	259.61	24.71	258.85	24.17	280.48	16.32	325.48	32.26	
DeepSeek-R1-671B	279.11	31.85	258.43	21.67	262.21	27.87	278.565	17.81	335.53	**30.19**	
DeepSeek-R1-Distill-7B	331.45	38.91	311.43	31.43	288.42	29.23	291.32	19.56	344.73	33.72	
Traffic-R1-3B	270.34	**27.95**	239.53	**21.11**	**238.03**	23.17	**277.83**	**15.51**	**324.11**	33.14	

token context window. It offers performance comparable to much larger models in zero-shot tasks, with enhanced capabilities in tool calling and multilingual support.

- **Qwen2.5-72B:** A LLM developed by Alibaba Cloud, with 72 billion parameters. It is designed for superior performance in zero-shot learning, particularly in coding, mathematics, and following complex instructions.
- **GPT-3.5-turbo:** A variant of OpenAI's GPT-3.5 model, finetuned for conversational applications. It excels in zero-shot dialogue systems and natural language processing tasks.
- **GPT-4o:** OpenAI's advancing multimodal language model that processes text, audio, and images. It provides advanced zero-shot capabilities across multiple tasks, with fast response times and cost efficiency.
- **DeepSeek-R1-671B:** A massive language model with 671 billion parameters, developed by DeepSeek AI. It specializes in zero-shot reasoning, particularly in mathematics and coding, and is available as an open-source model.
- **DeepSeek-R1-Distill-7B:** A Qwen2.5-based distilled version of DeepSeek-R1, featuring 7 billion parameters. It maintains competitive zero-shot performance in reasoning tasks through efficient distillation methods.

A.3 Traffic Incident Data

In this work, we evaluate the models' incident handling capabilities by collecting traffic emergency incident recordings from traffic management departments. We summarize these recordings into 200 representative textual examples to simulate the occurrence of out-of-distribution (OOD) traffic incidents. Below, we present selected anonymized examples of these incident recordings:

Examples for testing traffic incidents

ID 1: At this intersection, a traffic accident in the eastbound lane is causing significant congestion. **Response Action:** ETWT

ID 2: At this intersection, a pedestrian was struck in the northbound crosswalk. **Response Action:** ETWT/NLSL

ID 3: Report from the nearby intersection to the north: Heavy southbound traffic is approaching. **Response Action:** NTST/NLSL

ID 4: At this intersection, a school bus is stopped in the eastbound lane, loading students. **Response Action:** NTST

ID 5: At this intersection, a group of pedestrians is blocking the westbound crosswalk. **Response Action:** NTST

ID 6: Report from the nearby intersection to the east: A fire hydrant crew is slowing westbound traffic. **Response Action:** NTST/ELWL

ID 7: At this intersection, vehicles spun out in the westbound lane. **Response Action:** NTST

ID 8: At this intersection, for the XXX event, a marathon is passing through the eastbound lane. **Response Action:** ETWT

ID 9: At this intersection, a road rage incident was reported in the northbound lane due to a heavy traffic jam. **Response Action:** NTST

ID 10: At this intersection, it is 5:30 PM, the nearby western school is scheduled to dismiss, leading to increased vehicular traffic for student pick-up. **Response Action:** ETWT

A.4 More Experiment Result

We evaluate the performance of Traffic-R1 on standard signal control tasks in full-shot settings. We adopt the original deployment configurations of all learning-based baseline models for training in the

Responses of Traffic-R1 in different scenarios

Case 1: <think>Based on the current state of the intersection, Signal: ETWT (Eastern and Western through lanes) is the one that most significantly improves traffic conditions during the next phase. This is because this lane has **18 early queued vehicles and 13 approaching vehicles in Segment 3**, indicating a significant number of vehicles waiting to pass through and **incoming huge traffic pressure**. Approving this signal will allow these vehicles to proceed, easing the traffic flow.</think>

Case 2: <think>Based on the current state of the intersection, the signal phase that will most significantly improve traffic conditions during the next phase is NTST (Northern and southern through lanes). This is because **only the southern through lane has approaching vehicles in Segment 1**, which indicates that there are vehicles waiting to enter the intersection. By allowing vehicles in the northern and southern through lanes to flow, **this signal phase will help avoid the queue of approaching vehicles** and improve overall traffic flow.</think>

Case 3: <think>Based on the current state of the intersection, the signal phase that will most significantly improve traffic conditions during the next phase is ETWT (Eastern and Western through lanes). Although the **northern through lane has 20 early queued vehicles in Segment 2**, an ambulance is currently approaching from the east, heading west. Prioritizing ETWT is critical to ensure **the ambulance passes through the intersection without delay**, maintaining emergency response efficiency. This signal will also alleviate some traffic pressure by allowing eastern and western vehicles to proceed.</think>

Case 4: <think>Based on the current state of the intersection, the signal phase that will most significantly improve traffic conditions during the next phase is NTST (Northern and Southern pedestrian crossing). This is because **a large number of students are waiting to cross from the northern to southern direction**, indicating significant pedestrian demand. Activating this signal will allow these students to cross safely, reducing congestion at the crosswalk.</think>

Case 5: <think>Based on the current state of the intersection and the urgent message of an emergency fire truck approaching from the west, the signal phase that will most significantly improve traffic conditions during the next phase is ETWT (Eastern and Western through lanes). This is because **both ETWT and NTST have comparable queues, with approximately 15 early queued vehicles each in Segment 2, but prioritizing ETWT ensures the fire truck's swift passage**. Activating this signal will clear the western lane, facilitating emergency access while also easing general traffic flow.</think>

test environments, while Traffic-R1 is assessed directly in a zero-shot setting. The results, presented in Table 5, show that Traffic-R1, in its zero-shot configuration, outperforms all baselines, including RL-based methods that underwent extensive training and optimization on the test datasets. Notably, some advanced LLMs achieve impressive performance comparable to SOTA RL-based methods, demonstrating the potential of LLMs to fully replace traditional RL-based methods in traffic control tasks. Furthermore, Traffic-R1 achieves better performance than SOTA LLMs (e.g., DeepSeek-R1-671B) while utilizing only 1% of their parameter size.

A.5 Reasoning Cases of Traffic-R1

We provide several representative reasoning processes generated by Traffic-R1 during experiments across different scenarios. The results clearly demonstrate the model's comprehensive and logical reasoning ability in both conventional traffic situations and incident response tasks. The reasoning examples illustrate the model's understanding of traffic control, including its focus on queued vehicles and approaching vehicles. The model independently makes trade-offs to achieve better traffic efficiency. Furthermore, when an incident occurs, Traffic-R1 accurately analyzes the situation and identifies optimal actions to address it, while even striving to both maximize traffic efficiency and respond to the incident's needs. This showcases the model's ability to balance multiple objectives in dynamic traffic environments.

A.6 Deployment Details

A.6.1 Environment and System Adaption.

Our model, Traffic-R1, has been deployed on the official traffic management platform of a Tier-1 city in China, where it oversees 10 critical, interconnected intersections. This network serves over 55,000 vehicles daily and is situated in a central commercial district characterized by high-volume, dynamic traffic patterns. The intersections are configured as four-way junctions on arterial roads, with inter-intersection distances ranging from 600 to 800 meters. The traffic load is substantial, with peak daily throughput at a single intersection exceeding 21,000 vehicles.

To bridge the gap between simulation and real-

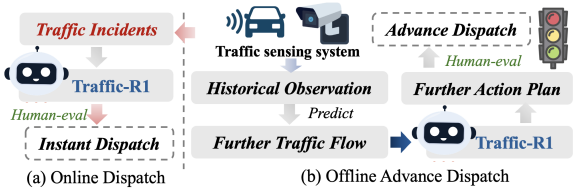


Figure 8: Online and offline dispatch framework

world complexity, two key adaptations were implemented. First, to manage the increased number of lanes typical of urban arterial roads, we aggregated lanes with identical traffic movements into a single logical input, thereby reducing state-space complexity for the model. Second, the system incorporates a mandatory pedestrian-crossing phase during morning and evening peak hours. This regulatory requirement introduces additional constraints to the signal timing plan, demanding greater adaptive capabilities from our model to maintain traffic efficiency. Real-world deployment necessitates two critical components: a multi-modal sensing system and a safety-compliant dispatch framework.

To provide the model with real-time, LLM-compatible inputs, we developed a multi-modal sensing suite. As illustrated in Figure 7, roadside cameras provide 2D visual data, which is processed by a fine-tuned Grounding DINO model for vehicle detection and classification. This is augmented by millimeter-wave radar data that provides precise vehicle-to-intersection distance measurements, creating a comprehensive traffic state representation.

Direct, fully online model control is infeasible in public infrastructure due to regulatory and safety mandates. To address this, we designed a hybrid online-offline dispatch framework (Figure 8). The dispatch framework is executed on a centralized server equipped with an NVIDIA Tesla T40 24G GPU with an average inference latency of less than 2 seconds (437 output tokens in average). This centralized deployment is optimal for our predominantly offline dispatch strategy, allowing for robust data aggregation and computation. For routine traffic management, the system operates offline. Historical traffic data spanning 30 days is used to train a LightGBM model with spatio-temporal embeddings to forecast traffic flow for the subsequent day. Based on these forecasts, Traffic-R1 generates a preliminary signal timing plan. Crucially, this plan is submitted to human experts at the traffic management authority for review and final approval before implementation. The online pipeline is reserved for real-time incident response, allowing the model to make immediate adjustments under expert supervision. This dual-pipeline approach ensures both operational efficiency and compliance with public safety standards. Furthermore, while the current deployment leverages a center server, the model is designed for future scalability. The 3B-parameter Qwen architecture of Traffic-R1 can be quantized for efficient execution on low-power edge devices that lack dedicated GPUs.

Table 6: Real-world A/B test results spanning 6 weeks.

Method	Average Queue↓	Maximum Queue↓	Working Hours↓
Manual	34.5	50.3	2+
Traffic-R1	31.3	48.1	0.5+
#Improve	**9.3%**	**4.4%**	**∼75%**

A.6.2 In-Situ Performance Evaluation

We conducted a six-week in-situ evaluation to quantify the performance of Traffic-R1. The experiment was structured as an A/B test, comparing our proposed system against the baseline, which consists of manual signal control by human operators. The two systems were deployed on an alternating weekly basis to mitigate temporal bias. Performance metrics, including average queue length, maximum queue length, and daily human operational hours, were recorded during weekday evening peak hours (16:00-19:30). Data from weekends and public holidays were excluded to ensure a consistent basis for comparison.

As detailed in Table 6, our system demonstrates marked improvements over the manual baseline. Specifically, Traffic-R1 reduced the average and maximum queue lengths by 9.3% and 4.4%, respectively. Furthermore, it achieved a substantial 75% reduction in the human labor required for daily operations. These results empirically validate the effectiveness and efficiency of our model in a real-world, dynamic traffic environment.

Algorithm 1 Asynchronous TSC Network with Symmetric Half-Step Communication

Input: Road network graph $G = (\mathcal{I}, E)$, partitioned intersection sets $(\mathcal{G}_1, \mathcal{G}_2)$.
Initialize: Environment Env, Agent policy π, message buffer $\mathcal{M}$.
for each time_step t in Total_Steps **do**
// — Half-step for agents in $\mathcal{G}_1$ —
  Create new empty buffer $\mathcal{M}_{next}$.
  for each intersection $i \in \mathcal{G}_1$ **do**
    $O_i, M_{in,i} \leftarrow Env.get_observation(i), \mathcal{M}[i]$ ▷ Read messages from $\mathcal{G}_2$ from prev. half-step
    $A_i, M_{out,i} \leftarrow \pi(O_i, M_{in,i})$
    if $M_{out,i} \neq$ null **then**
      for each neighbor $j \in \mathcal{N}(i) \cap \mathcal{G}_2$ **do**
        if $dist(i, j) \leq$ 2km **then**
          $\mathcal{M}_{next}.add(j, M_{out,i})$
        end if
      end for
    end if
  end for
  $\mathcal{M} \leftarrow \mathcal{M}_{next}$ ▷ Update message buffer for the next half-step (for $\mathcal{G}_2$)
  $Env.update_state(\{A_i\}_{i \in \mathcal{G}_1})$
// — Half-step for agents in $\mathcal{G}_2$ —
  Create new empty buffer $\mathcal{M}_{next}$.
  for each intersection $j \in \mathcal{G}_2$ **do**
    $O_j, M_{in,j} \leftarrow Env.get_observation(j), \mathcal{M}[j]$ ▷ Read messages from $\mathcal{G}_1$ from prev. half-step
    $A_j, M_{out,j} \leftarrow \pi(O_j, M_{in,j})$
    if $M_{out,j} \neq$ null **then**
      for each neighbor $i \in \mathcal{N}(j) \cap \mathcal{G}_1$ **do**
        if $dist(j, i) \leq$ 2km **then**
          $\mathcal{M}_{next}.add(i, M_{out,j})$
        end if
      end for
    end if
  end for
  $\mathcal{M} \leftarrow \mathcal{M}_{next}$ ▷ Update message buffer for the next half-step (for $\mathcal{G}_1$)
  $Env.update_state(\{A_j\}_{j \in \mathcal{G}_2})$
end for