Language-Informed Traffic Signal Control: What a Zero-Shot LLM Gains from a Structured Incident Report

1st Shashank Kafle
Sirindhorn International Institute of Technology
Thammasat University
Pathum Thani, Thailand
shashankkafle0@gmail.com

2nd Teerayut Horanont
Sirindhorn International Institute of Technology
Thammasat University
Pathum Thani, Thailand
teerayut@siit.tu.ac.th

***Abstract*—Urban traffic congestion is a major global problem, and the difficulty of adapting to non-recurrent events such as lane blockages is a key contributor. Reinforcement learning (RL)-based controllers perform well under recurrent traffic patterns but cannot ingest incident information at inference time: they must either have encountered similar disruptions in training or infer the disruption indirectly from the queue it creates. Large language models (LLMs), by contrast, can interpret a structured incident report supplied in their prompt. We isolate the contribution of that report through a matched ablation: two identical zero-shot LLM signal controllers face the same severe lane blockage on a 4×4 network in the simulation of urban mobility (SUMO), each selecting phases from intersection state, differing only in whether the report appears in the prompt. The informed controller outperforms its event-blind counterpart by a paired margin that well exceeds our noise criterion, while both zero-shot arms trail a trained RL baseline. The text channel works, but most of the gap to RL is already present under normal traffic, so it is not explained by missing event information.**

***Index Terms*—traffic signal control, large language models, zero-shot control, incident management, intelligent transportation systems**

I. Introduction

Congestion-related delay, emissions, and safety risks remain central challenges for urban mobility, and adaptive traffic signal control (ATSC) is a primary tool for addressing them [1]. Learning-based ATSC methods perform well under recurrent conditions, but non-recurring events such as lane blockages shift traffic away from the patterns these controllers were trained on [2], [3]. Recent RL approaches to adaptive traffic signal control have achieved state-of-the-art performance, though evaluations are typically conducted under recurrent, historically recorded demand [4], [5].

RL approaches to traffic signal control are typically trained on recurrent, undisrupted demand, and their performance has been shown to degrade when conditions at deployment fall outside the training distribution [1], [2]. Moreover, the state representations standard in this literature (lane occupancies, queue lengths, and phase information) contain no explicit incident signal [2]. Recent large language model (LLM)-based controllers have begun to handle incident scenarios, including lane blockages [6] and long-tail events such as emergency vehicles and degraded sensing [7]; in these systems, incident information is supplied to the LLM alongside the traffic state, but its contribution is not isolated: no controlled comparison measures how the same controller behaves on the identical event with and without the incident description. This ablation is the contribution of this paper. Without this separation, an LLM-based controller's performance cannot be attributed to its control capability or to its use of the event information, leaving unclear which ability limits performance and what role an LLM should play.

In this paper, we evaluate a zero-shot LLM as a direct signal controller under a lane blockage, comparing otherwise-identical prompts with and without a structured incident report describing the event. Across three matched seeds, the report reduces average travel time by $8.61 \pm 0.72\,\mathrm{s}$ relative to the event-blind arm, twelve times the SD of the per-seed differences, above our pre-specified 5× criterion. It is not obvious in advance that the report should help: the model may ignore it, misapply it, or already cope through the state dynamics alone. Each outcome carries a different implication for language-informed control, which the ablation is designed to separate. Our contributions are threefold: (1) we design an internally controlled ablation that isolates the contribution of a structured incident report to a zero-shot LLM signal controller, holding the network, demand, seeds, and all other prompt content fixed, implemented in a SUMO/TraCI pipeline; (2) we show that the report measurably improves performance under the blockage relative to the matched event-blind arm, while both LLM arms trail a trained RL controller, mostly by a gap already present without the blockage; (3) we situate both LLM arms against fixed-time, MaxPressure, and a trained CoLight baseline under the identical scenario, as reference context for the ablation.

This work was supported by the Excellent Foreign Students (EFS) scholarship of Sirindhorn International Institute of Technology (SIIT), Thammasat University.

II. Related Work

Recent works employ LLMs as signal controllers via textual state prompts, either zero-shot [8] or with task-specific fine-tuning [9]. Extensions target heterogeneous intersection configurations via non-textual state encodings [10], vision-based perception feeding an LLM agent [11], and emission-aware objectives [12]. Yet evaluation remains confined to recurrent or volume-scaled flows, leaving incident-induced disruptions largely untested.

Handling irregular events such as accidents and lane blockages is a recognized challenge for signal control, since such events invalidate the conditions that adaptive controllers assume [6]. LLM-based approaches have addressed this by wrapping an LLM around an existing policy: [6] feeds real-time incident descriptions to an LLM that retunes the parameters of an underlying controller, [7] use the LLM to review and correct the decisions of an RL policy under anomalies, and [13] guides the LLM through retrieval over a database of historical emergency cases. Closest to our setting, [14] provides traffic state, including special-event flags, as natural-language prompts for an LLM controller; however, the only event evaluated is emergency-vehicle passage at a single intersection. In all of these, the incident information is either absent or present in every condition, so none isolates its contribution. Whether the same standalone zero-shot controller decides differently with and without the event description remains untested.

RL-based signal controllers are brittle under disruptions: trained policies degrade under incidents, demand surges, and sensor failures [2], [3]. One line of work accelerates re-adaptation when conditions change, through transfer and meta-learning [15]–[17] or drift-triggered relearning and online re-optimization [18]. A second line designs for the disruption: [19] coordinates neighboring signals around a malfunctioning one, [20] uses cell-level occupancy rather than aggregate features so a trained agent responds to accidents at locations unseen in training, and others build disruption handling into the reward or coordination structure [5], [21], [22]. Even where external event data is integrated, it enters as structured feeds triggering adaptation mechanisms [23], not as a description the controller itself interprets. Across this literature, incident information cannot be supplied to a trained policy at inference time in natural language; accommodating an unforeseen event requires retraining, transfer, or architectural provision, a limitation that a language interface could remove, although whether the language model itself is needed to exploit such information is a separate question.

III. Methodology

A. Environment and Network

All experiments run in SUMO 1.27.1, controlled through TraCI at the default step length of 1.0 s, with each episode lasting 3600 s. We use the Hangzhou 4×4 network (`hangzhou_4x4_gudang_18041610_1h`) from the traffic-signal-control benchmark. [4], [24], [25]: 16 intersections in a 4×4 grid, three lanes per approach (left-turn, through, right-turn), with demand derived from camera data. The dataset is distributed in CityFlow format; we converted the network and routes to SUMO inputs using LibSignal's converter [26] and verified that the conversion preserves all 2,983 trips with their departure times and routes. We chose SUMO over CityFlow because SUMO's traffic control interface (TraCI) permits runtime modification of the simulation state, which the mid-episode blockage injection requires. The converted route file is used unmodified and each experimental cell is run under three random seeds (1–3). SUMO's default behavior teleports any vehicle stationary for 300 s to the next free edge on its route; we disable this (`--time-to-teleport -1`), since teleportation would dissolve exactly the queues the blockage scenario is designed to create. Each intersection cycles through the four signal phases shown in Fig. 1 under a fixed timing scheme: the controller selects a phase, which holds green for 30 s, followed by 3 s yellow and 2 s all-red on transition. Right-turn movements are permissively green throughout and are not signal-controlled, following the benchmark's convention.

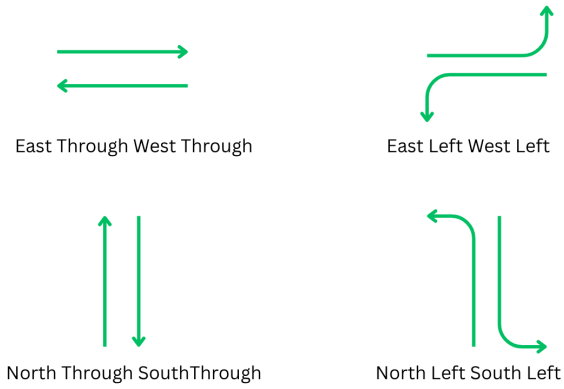


Fig. 1. The four signal phases. Each phase releases one pair of non-conflicting movements; right turns are permissive throughout.

B. Blockage Location and Scenario Selection

The blockage site was fixed before any simulation, from a static analysis of the network and route files. For each of the 16 approaches into the four interior intersections, we computed the ratio of demand to through-lane storage, taking storage as lane length divided by 7.5 m per queued vehicle (5.0 m vehicle length plus 2.5 m minimum gap). The North approach of intersection_2_3 (road_2_4_3_1; 572.8 m, storage 76.4 vehicles) ranks first under both denominators — 2.44 on total demand (186 vehicles) and 1.49 on through demand (114) — in each case roughly 20% above the next approach. Blockage duration was set to 1200 s (steps 500–1700), leaving 1900 s of recovery within the 3600 s horizon; onset precedes the peak arrival period on the blocked approach. [27] report, for a dataset of 472 urban accidents at 75 intersections, a mean impact duration of 31.8 min and a median of 25 min, where impact duration comprises discovery, response, clearance and

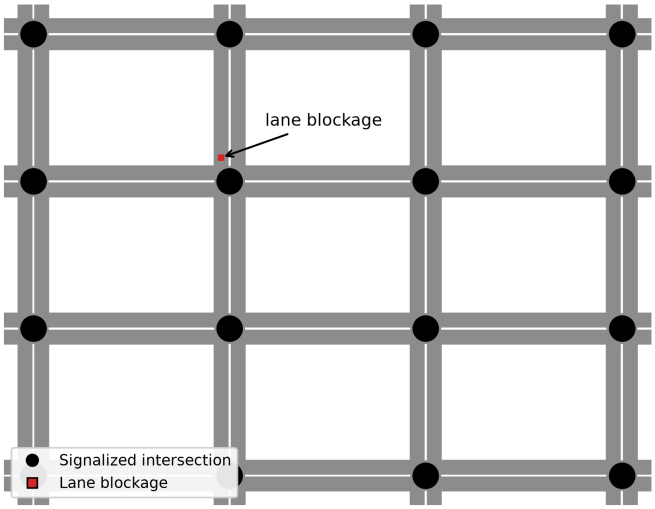


Fig. 2. The Hangzhou 4×4 network (north is up). Intersections are named column_row. The blockage site (red) is on the through lane of the North approach into intersection 2_3, the intersection directly below the marker; intersection 2_4, directly above it, is the intersection whose exit road is blocked.

recovery. The physical blockage is strictly shorter than this interval, so 1200 s falls below the empirical ceiling. The same study places 95% of impacted intersections within 1.8 km of the accident, which motivates measuring network-wide over the full horizon rather than locally.

Acceptance criteria were fixed before these runs: the paired change in average travel time (ATT) under MaxPressure had to reach approximately five times the clean-run seed spread, completion had to stay above 0.70 without gridlock, and queues had to recover inside the horizon. The scenario met all three. Paired ΔATT was 27.67 ± 0.83 s against a clean-run seed spread of 1.60 s ($17.3\times$), and completion under the blockage was 0.897–0.900, matching the normal-condition value of 0.898. The blockage is realized as a single obstacle vehicle stopped via TraCI on the blocked lane identified above, 10 m upstream of the stop line. It is inserted at step 500 and removed at step 1700. At this position no vehicle can merge around it, so the lane's full through movement is cut off; at larger offsets, following vehicles were observed to bypass the obstacle by changing lanes. Vehicle teleporting is disabled so that queues trapped behind the obstacle are preserved (Section III-A). The obstacle vehicle itself is excluded from all metrics: it appears in no per-vehicle accounting, no queue counts, and not in the completion denominator.

C. Compared Methods

We evaluate five controllers under identical network, demand, and signal-timing rules. FixedTime and MaxPressure serve as classical baselines, and CoLight as a trained reinforcement-learning reference; these three provide context for interpreting the LLM results, as the paper's core comparison is internal to the LLM pair. The two LLM controllers, LLM+text and LLM−text, are identical in model, prompts,

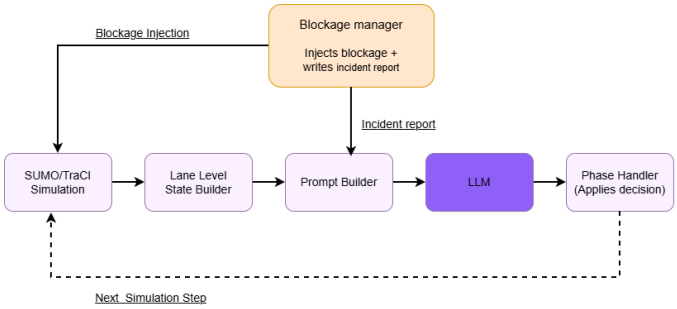


Fig. 3. Control loop and prompt composition; the incident-report section is the sole between-arm difference.

and world state except for one difference: when a blockage is active, the +text arm's prompt additionally contains a structured incident report rendered in the controller's observation vocabulary, while the −text arm's prompt omits these sections and is otherwise unchanged. Incident reports take two forms depending on the receiving intersection's relation to the blockage — one for the intersection whose approach lane is blocked and one for the intersection whose exit lane is blocked; report content and delivery rules are specified in Section III-D.

FixedTime cycles the four phases in a fixed order with equal green time, ignoring traffic state entirely. MaxPressure follows the max-pressure principle [28]: at each decision point, every phase is scored as the number of stopped vehicles on its protected incoming lanes minus those on its outgoing lanes, and the highest-scoring phase is activated. Our implementation computes stopped-vehicle counts (speed below 0.1 m/s) at the lane level without turn-ratio weighting, and holds the current phase on ties to avoid unnecessary transitions. CoLight [24] is a graph-attention reinforcement-learning controller that learns coordinated phase selection across neighboring intersections; it is trained once on normal traffic only, under a single training seed and the same route configuration used in evaluation, and its fixed weights are evaluated on seeds 1–3 in both conditions.

D. The LLM Controller

All LLM controllers use Qwen2.5-14B-Instruct [29] in a strictly zero-shot setting, with no fine-tuning or in-context examples. Inference uses greedy decoding (do_sample=False) for reproducibility: phase selection is a discrete choice with a fixed answer format, and sampling would introduce decision variance unrelated to the experimental treatment. Generation is capped at 1,024 new tokens; in practice, responses remain well below this limit, and all runs achieve a 100% valid-response rate. The model runs in float16 on a single NVIDIA A40 GPU using its native ChatML chat template. Prompts for all intersections requiring a decision at the same simulation step are submitted as a single batch; we note that batched float16 inference can introduce minor numerical nondeterminism relative to strictly serial execution, though decoding itself is greedy. Each intersection is controlled independently, with no inter-agent communication. A selected phase receives 30 s

of green, after which the controller is queried for the next phase. If the current phase is reselected, the next green interval begins immediately; otherwise a 3 s yellow and 2 s all-red clearance interval is inserted before the new phase's 30 s green. The inter-decision interval is therefore 30 s when the phase is held and 35 s when it changes.

Observation encoding. We adopt the prompt template of LLMLight [9] for observation verbalization and action formatting. At each decision point the prompt is rebuilt from the current intersection state. Each intersection has four approaches (north, south, east, west), each with a through lane and a left-turn lane, and four signal phases, each releasing one pair of lanes. Each lane is divided into three segments by vehicle position along the link: segment 1 covers the first tenth of the link, segment 2 extends from one tenth to one third, and segment 3 covers the remainder; these boundaries are set in our implementation. The prompt reports, per signal, the number of early-queued vehicles and the number of approaching vehicles in each segment. Segment labels convey ordinal proximity only: the prompt states neither the segment spans nor any metric distance. The observation contains no historical information and no state beyond the intersection's own approaches.

Decision protocol. The model is instructed to produce a step-by-step analysis and then a phase choice enclosed in a `<signal>` tag. If the completion contains no parseable tag, or the tagged phase is not one of the four valid signals, the controller retains the current phase. Decoding is greedy, with a maximum of 1024 new tokens. Both arms share an identical system prompt (quoted below), which supplies generic commonsense about lane blockages; the arms differ only in the event-text sections defined below. The event text is a structured incident report rendered in the controller's observation vocabulary: (signal names, approaches, and segments) rather than free-form operator language. This grounds the report in the state description: it names the exact signal and the counts it corrupts, so the model can connect text to state without cross-referencing. A single incident modifies the prompts of at most two intersections' agents, each receiving the report from its own perspective. The approach-side report is inserted only at the intersection whose approach is blocked; the exit-side report only at the intersection whose exit road is blocked; every other intersection receives a prompt byte-identical to the no-blockage case. For the intersection whose approach is blocked, the report reads:

> LANE BLOCKAGE REPORT (approaches to this intersection)
> The following blockage is currently active on an approach to this intersection: North approach, through lane (served by signal NTST), segment 1: collision — the lane is fully blocked. Vehicles behind the blockage in this lane cannot reach the intersection until it clears. Within signal NTST, the North queued and approaching counts reported above include these vehicles; the South counts are unaffected by this blockage.

TABLE I
EXPERIMENTAL CONDITIONS OF THE TWO LLM ARMS. "DURING" AND "AFTER" REFER TO THE BLOCKAGE INTERVAL $500 \le t < 1700$ S.

	LLM+text	LLM−text
Model and decoding	identical	
System prompt	identical	
Network, demand, seeds	identical	
Blockage event	identical	
2_3 prompt, during	+ approach report	—
2_4 prompt, during	+ downstream report	—
Other 14 intersections	identical to normal	
All prompts, after	identical to normal	

For the intersection whose exit road is blocked downstream:

> DOWNSTREAM BLOCKAGE REPORT (roads leaving this intersection)
> The following road leaving this intersection is blocked further downstream: Road exiting toward the South: collision — 1 of 3 lanes fully blocked, located 563 m past this intersection on a 573 m link. Movements releasing vehicles onto this road: North→South through (part of signal NTST) and East→South left turn (part of signal ELWL).

REPORT TIMING A report is inserted, before the prompt's final instruction, into every prompt built while the blockage is active ($500 \le t < 1700$ s), and into no other prompt. Since each intersection decides every 30–35 s, the first informed decision fell 0–20 s after onset and the last at 1670–1695 s, giving 35–37 informed decisions per intersection per run. The report is an oracle: it appears without detection delay and gives the exact location and severity, but no clearance time, consistent with real incident conditions [27]. Clearance is not announced; the report is simply absent from the next prompt. The two arms are otherwise identical in model, decoding, system prompt, network, demand, blockage, and seeds; the other 14 intersections receive identical prompts in both arms throughout.

Both arms share a common system prompt containing generic blockage commonsense:

> COMMONSENSE SYSTEM PROMPT You are an expert in traffic management. You can use your knowledge of traffic commonsense to solve this traffic signal control tasks. Incidents may block lanes anywhere in the network. A queue trapped behind a full lane blockage cannot discharge even when its signal is green. A road that is blocked further downstream has reduced storage and discharge capacity, and vehicles released onto it may queue back into the intersection. When such information is available, consider whether queued vehicles are actually able to move before allocating green time to them.

The ablation therefore isolates the event facts, not the concept of blockage reasoning: the LLM−text arm runs in an identical world with the identical blockage, and its prompts differ only in the omission of the report sections. It is event-blind but not blockage-naïve. This omission is the

only between-arm difference. By construction, the +text arm receives a longer prompt, so event content and prompt length are not separated in this design.

Each arm is run on three matched seeds under the identical scenario, alongside a no-blockage LLM condition.

E. Metrics and Analysis

We report four traffic metrics per run:

Average travel time (ATT): each vehicle's in-network time from insertion to arrival. Vehicles still en route when the 3600 s horizon closes are charged their time in the network so far. Measured on the full in-network clock.

Average waiting time (AWT): each vehicle's total accumulated stopped time (speed below 0.1 m/s), under the same population and horizon treatment as ATT.

Average queue length (AQL): network-wide count of stopped vehicles, sampled once per simulation step (1 s) and averaged over the horizon.

Completion rate: fraction of the 2,983 route-file vehicles that finish their trip within the horizon.

ATT and AWT are computed over all inserted vehicles, whether or not they complete their trip. The synthetic obstacle vehicle that realizes the blockage is excluded from all per-vehicle accounting, queue counts, and the completion denominator.

In the LLM runs, vehicles still in the network at 3600 s are mostly late entrants (about 93% of them entered after 3000 s), so including them lowers ATT. Restricting ATT to completed trips changes the ablation from -8.61 ± 0.72 s to -9.78 ± 0.70 s ($13.9\times$); the verdict is unchanged.

For the LLM arms we additionally report an integrity panel over LLM-queried decisions, i.e., every decision point at which the model was actually called (empty-intersection holds are excluded from the denominator):

Valid-response rate: decisions where the model named a valid phase or explicitly answered "no change".

Parse-error rate: responses with no phase tag or a tag naming an invalid phase.

Hallucination rate: responses whose tag names an invalid phase only, a subset of the parse-error rate.

Inference-error rate: decisions where the model call itself failed and produced no response.

Valid-response, parse-error, and inference-error rates sum to one. On any parse or inference error the controller holds the current phase (see Section III-D).

All comparisons are paired across matched seeds. Each experimental cell is run under three seeds and reported as mean $\pm$ standard deviation over the three runs. For any two conditions compared, we compute the difference between same-seed runs and report the mean and standard deviation of the three per-seed differences. Because the seeds are shared, seed-driven demand variation is common to both runs and cancels in the subtraction; the residual spread of the per-seed differences is therefore the noise term of the paired comparison. With n = 3 pairs, exact nonparametric paired tests cannot reach $p < 0.25$, and the normality a t-test assumes cannot be checked on three differences, so we instead adopt a fixed decision rule, committed before the CoLight and LLM results were obtained and applied uniformly to every paired comparison: a difference is said to pass the noise criterion if and only if the absolute mean of the per-seed differences is at least five times their standard deviation. This is a fixed effect-size threshold of our own choosing, not a test of statistical significance, and implies no p-value. The rule is two-sided. It adjudicates ATT only; AQL and AWT differences are computed identically and reported as directional support, with no thresholds of their own. Effects below the bound are reported as below the noise criterion. Comparisons of blockage-induced gaps across controllers are descriptive and receive no noise verdict: the controllers complete different fractions of demand, so their travel-time averages are taken over different vehicle populations.

Each of the three reference controllers (FixedTime, MaxPressure, and CoLight) is evaluated under two conditions, normal traffic and the blockage scenario (Section III-B), yielding six runs per controller across seeds 1–3. The LLM controller is evaluated under three conditions on the same seeds: normal traffic, blockage with the incident report (LLM+text), and blockage without it (LLM$-$text). Under normal traffic the two LLM arms are identical by construction — report sections appear in the prompt only while a blockage is active — so a single normal condition serves as the shared baseline for both. Each run is one simulation episode (Section III-A). This yields 27 runs in total. Seeds are matched throughout: every run with the blockage is paired with the same-seed normal run of its controller, and the two LLM blockage arms are paired with each other, enabling the per-seed differences on which the analysis in Section III-E operates. Group means and paired differences for all controllers are reported in Table II; the ablation deltas in Table III.

All 27 runs share the same network, the same route file, and therefore the same scheduled departure time for every trip: the demand specification is identical in every run. The random seed varies only the driving dynamics (the car-following imperfection parameter ($\sigma = 0.5$) and per-vehicle speed factors drawn from a distribution with 10% deviation) and, downstream of these, the realized state of the simulation: under congestion, a vehicle whose entry edge is occupied at its scheduled departure is inserted only when space becomes available, so realized insertion times and trajectories differ across seeds while the demand itself does not. CoLight is trained once, on normal traffic only, under a single training seed, and its fixed weights are evaluated on seeds 1–3 in both conditions; the blockage scenario is thus by design outside its training distribution, and training and evaluation use the same route configuration (Section III-C). The LLM runs execute on the GPU server described in Section III-D; the remaining controllers run on CPU. Since control is synchronous over TraCI (the simulation advances only after the controller returns its decision) the compute platform affects wall-clock time but not simulated dynamics.

The grid supports three levels of comparison. The main

comparison is between the +text and −text arms of the LLM-based controller: the two arms share seed, world state, model, and system prompt, and differ only in the incident-report sections, so any difference that passes the noise criterion is attributable to the event text. The second is the degradation of each controller in the presence of the blockage, measured as the paired difference between its blockage injected runs and normal runs. The last is the comparison of these degradations across controllers, which is descriptive only and receives no noise verdict (Section III-E).

IV. Results

Under normal conditions, the controllers reproduce the expected performance ordering (CoLight 378.33 ± 0.96 s $<$ MaxPressure 389.61 ± 1.60 s $<$ LLM 399.51 ± 17.74 s $<$ Fixed-Time 538.81 ± 2.20 s; Table II), which validates the pipeline before the blockage comparisons. The LLM-based controller lands in the adaptive band; $\sim$10 s behind MaxPressure and $\sim$139 s ahead of fixed time. The zero-shot LLM has a valid response rate of 1.0, zero parse errors, zero hallucinated phases across all seeds. The completion rates are similar for CoLight, LLM and MaxPressure ($\sim$0.9) versus 0.844 for FixedTime. Table II also reports completion and the vehicles remaining in the network at 3600 s. The blockage shifts neither by more than 0.3 percentage points or 10 vehicles for any controller, and the two LLM arms are indistinguishable on both, so the ATT comparisons are taken over nearly identical vehicle populations; FixedTime, which leaves about 460–470 vehicles in the network, is the exception discussed below.

Injecting the blockage into otherwise identical simulations yields per-controller paired deltas that differ by an order of magnitude (Table II). MaxPressure degrades most, by $+27.67 \pm 0.83$ s ATT ($33.2\times$ the paired-delta SD, passes the noise criterion under the pre-specified criterion), followed by CoLight at $+15.61 \pm 0.93$ s ($16.7\times$, passes the noise criterion). FixedTime shifts by only $+4.17 \pm 2.59$ s ($1.6\times$), below the noise criterion. The same physical event is therefore consequential for adaptive control while its cost is strongly controller-dependent. Since FixedTime completes only 0.84 of demand versus $\sim$0.90 for the other controllers, its ATT is computed over a different vehicle population; cross-controller comparisons of delta magnitudes are accordingly descriptive. This controller dependence motivates the central question of the ablation: whether describing the event changes what the LLM controller does under it.

For the LLM-based controllers, the +text arm attains lower ATT than the −text arm (404.57 ± 14.02 vs 413.18 ± 14.55 s; paired $\Delta = -8.61 \pm 0.72$ s, Table III). Paired across matched seeds, this is $12\times$ the paired-delta SD and passes the noise criterion. AWT corroborates this in sign and magnitude (-8.48 ± 0.83 s). The LLM group's SDs are the largest ($\pm$14–18 s vs ≤ 2.2 s for the other controllers), driven by a consistent seed-1 offset across all three LLM configurations — yet the paired deltas are tight ($\pm$0.72). The +text vs normal comparison ($+5.06 \pm 5.10$ s, $1.0\times$) and −text vs normal ($+13.67 \pm 5.20$ s, $2.6\times$) are both below the noise criterion at n = 3, while the between-arm difference is not.

To examine how the report changed control decisions, Table IV reports the green-time allocation during the blockage at intersection 2_3, whose North through lane is blocked, and at 2_4, whose exit road is blocked. Without the report, the controller at 2_3 gave 85.3% of green time to NTST, the phase serving the blocked lane, and changed phase only 10.7 times in 1200 s. With the report, NTST received 25.2%, below its 43.6% share under normal traffic, and the remaining green went mainly to ETWT, with the rest to the two left-turn phases (27.7 phase changes; paired $\Delta = +17.0 \pm 3.6$). Since every selection grants a fixed 30 s green, the share of decisions selecting each phase matches its green share to within one percentage point. NTST also serves the unblocked South through movement, so the event-blind arm's green was not entirely lost, but one of the two movements it released could not discharge. After clearance, the NTST share at 2_3 returns to 49–52% in both arms. At 2_4, the shift is small. As specified in Section III-E, these differences are reported as directional support; the noise criterion is applied to ATT only.

Table V reports traffic conditions around the blockage. Queues are grouped by distance from 2_3: "other lanes" are the lanes entering 2_3 except the blocked one, and "adjacent" covers the four intersections one step away. With the report, queues on the other lanes into 2_3 stayed at their no-blockage level (16.6 m) instead of growing several-fold. Part of the released traffic queued at the adjacent intersections, and intersections farther away showed no consistent change. After clearance, shorter residual queues indicate faster recovery, and the travel-time gain is concentrated in vehicles departing during the blockage. The network-average gain therefore originates in control at the blocked intersection.

Both arms of the LLM-based controller remained well-formed under the blockage across all seeds: valid response rate 1.0, zero parse errors, zero hallucinated phases.

Inference latency. Control in simulation is synchronous over TraCI, so latency does not affect the results. It does, however, rule out real-time operation in our setup: a decision took a median of 42.1 s (95th percentile 78.5 s), and 99.9% of calls exceeded the 30 s green interval. Most of this time is spent generating the step-by-step analysis ($\sim$700 tokens per response).

Descriptively, the +text arm under blockage (404.57 s) still trails CoLight under the identical event (393.94 s). The event text measurably helps the LLM-based controller but it still lags behind CoLight (Table II).

V. Discussion

Under the blockage, the mean paired Δ for LLM+text − LLM−text is -8.61 ± 0.72 s ATT (lower ATT in the +text arm) and both arms trailed behind CoLight. This shows the event text measurably improves the LLM controller's performance.

By construction, the insertion also lengthens the +text prompt, so text content and prompt length are not separable

TABLE II

AVERAGE TRAVEL TIME (S; MEAN ± SD OVER SEEDS 1–3) UNDER NORMAL CONDITIONS AND WITH BLOCKAGE, WITH THE PAIRED PER-SEED DIFFERENCE. THE MULTIPLIER IS THE ABSOLUTE MEAN OF THE PER-SEED PAIRED DELTAS DIVIDED BY THEIR SD. COMPL.: SHARE OF THE 2,983 ROUTE-FILE VEHICLES THAT FINISH WITHIN 3600 S; REMAIN.: VEHICLES STILL IN THE NETWORK AT 3600 S; BOTH NORMAL / BLOCKAGE, SEED MEANS.

Controller	Normal	Blockage	Paired Δ	Mult.	Compl. (%)	Remain.
FixedTime	538.81 ± 2.20	542.99 ± 0.52	$+4.17 \pm 2.59$	$1.61\times$	84.4 / 84.2	459 / 469
MaxPressure	389.61 ± 1.60	417.28 ± 1.04	$+27.67 \pm 0.83$	$33.19\times$	89.8 / 89.9	304 / 302
CoLight	378.33 ± 0.96	393.94 ± 0.29	$+15.61 \pm 0.93$	$16.74\times$	90.3 / 90.3	290 / 288
LLM−text	399.51 ± 17.74	413.18 ± 14.55	$+13.67 \pm 5.20$	$2.63\times$	90.0 / 89.9	299 / 301
LLM+text	399.51 ± 17.74	404.57 ± 14.02	$+5.06 \pm 5.10$	$0.99\times$	90.0 / 90.0	299 / 298

TABLE III

ABLATION UNDER BLOCKAGE: PAIRED PER-SEED DIFFERENCE, LLM+TEXT − LLM−TEXT (MEAN ± SD, SEEDS 1–3). NEGATIVE VALUES MEAN THE STRUCTURED INCIDENT REPORT IMPROVES THE METRIC.

Metric	Paired Δ	Mult.
Avg. travel time (s)	-8.61 ± 0.72	$11.95\times$
Avg. waiting time (s)	-8.48 ± 0.83	—
Avg. queue length	-7.03 ± 0.69	—

TABLE IV

GREEN-TIME SHARE BY PHASE DURING THE BLOCKAGE (500–1700 S), LLM CONTROLLER, SEED MEANS OVER SEEDS 1–3. Δ: PAIRED PER-SEED DIFFERENCE, +TEXT − −TEXT (MEAN ± SD, PERCENTAGE POINTS), DIRECTIONAL SUPPORT ONLY (SECTION III-E). YELLOW AND ALL-RED TIME EXCLUDED. EACH Δ HAS THE SAME SIGN IN ALL THREE SEEDS.

	+text	−text	Δ
2_3: NTST	25.2	85.3	-60.1 ± 8.0
2_3: ETWT	47.1	10.5	$+36.6 \pm 7.5$
2_4: NTST	11.9	18.7	-6.8 ± 4.2

in this design. The structural difference is, however, local and temporary: prompt construction is identical across arms except for the two insertion sections, which appear only at the affected intersections and only while the blockage is active. State values in the prompts diverge between arms over the course of a run, but this divergence is a consequence of the differing control decisions (the treatment effect itself) not an additional manipulation.

TABLE V

TRAFFIC AROUND THE BLOCKAGE, LLM CONTROLLER, SEED MEANS OVER SEEDS 1–3. Δ: PAIRED PER-SEED DIFFERENCE, +TEXT − −TEXT (MEAN ± SD), DIRECTIONAL SUPPORT ONLY (SECTION III-E). DURING: 500–1700 S; AFTER: 1700–3600 S. QUEUES ARE LANE QUEUE LENGTHS FROM SUMO'S QUEUE OUTPUT (M). EACH Δ HAS THE SAME SIGN IN ALL THREE SEEDS, EXCEPT DEPARTURES AFTER CLEARANCE.

	+text	−text	Δ
Queue (m)			
2_3 other lanes, during	17.5	104.5	-87.0 ± 33.5
2_3 other lanes, after	18.2	30.0	-11.8 ± 2.4
Adjacent int., during	128.0	102.1	$+25.9 \pm 9.6$
Network, during	831.4	917.6	-86.2 ± 29.1
Network, after	407.9	453.7	-45.8 ± 9.6
Travel time by departure window (s)			
Departing before	453.3	459.9	-6.6 ± 3.3
Departing during	457.7	475.8	-18.1 ± 2.7
Departing after	346.8	348.3	-1.5 ± 1.7

Although the ablation shows the text channel is functioning, both LLM arms trail CoLight, which is blind to the event, so access to event information is not what limits the LLM here. Most of the gap is present without the blockage: the LLM trails CoLight by 21.18 s under normal traffic, more than the 10.63 s gap under the blockage. One explanation is the base control capability of this zero-shot configuration; others are differences in observation (CoLight sees neighbouring intersections, each LLM agent only its own approaches), in coordination, and in training on the evaluation routes. Our design does not separate these, since only one LLM, one prompt, and no fine-tuned model were evaluated. Fine-tuning has brought LLM controllers close to RL performance [9], though in a different simulator, with a different base model, and under normal traffic only; whether it closes the gap here while preserving use of the report is left to future work. Cross-controller comparisons here are descriptive; completion rates differ across controllers, so the underlying vehicle populations are not identical (Section III-E).

Taken together, these results suggest that event-information use and control capability can be assessed separately: the report reduced ATT by 8.61 s even though both LLM arms trailed CoLight. This suggests a second route alongside fine-tuning, in which the LLM supplies event information to an established controller or selects among vetted control policies rather than acting as a per-step controller. This role is not tested here; Section VI describes the information-matched classical control it would require.

Because signal control is safety-critical, it matters which safeguards the design provides and which it does not. The LLM never sets individual signal heads: it selects one of four predefined phases, each releasing only non-conflicting movements (Fig. 1), and the yellow and all-red clearance intervals and the 30 s minimum green are enforced by the surrounding controller, not chosen by the model. An unparseable or invalid response defaults to holding the current phase (Section III-D). The 100% valid-response rate we observe is an empirical result for these runs, not a guarantee. Three hazards remain unaddressed. First, there is no maximum green, so repeated reselection could starve a movement indefinitely. Second, the measured latency (median 42 s per decision, Section IV) exceeds the 30 s control interval; a deployed system would need a faster model or serving stack and a decision deadline with a default action. Third, the controller trusts the incident report, so a false or stale report could withhold green from a

lane that is actually clear. These are hazards of liveness and information integrity rather than of conflicting movements, and they favor the role suggested above, in which the LLM advises a verified controller that retains these guarantees.

VI. Limitations

First, separating the value of the event information from the value of interpreting it requires an information-matched classical control, a conventional controller given the same event facts through a hand-coded rule; we scope this control to future work. Without it, our results show that event text helps the LLM controller, not that an LLM is required to exploit such information.

Second, the inserted event report necessarily lengthens the +text prompt, so text content and prompt length are confounded by construction; Section V bounds this difference as local and temporary. A length-matched placebo insertion is the control that would separate the two, which we leave to future work.

Third, with three matched seeds, formal paired hypothesis tests are uninformative at n = 3, so we pre-specified an effect-size rule instead: a paired difference counts as passes the noise criterion only if it exceeds five times the standard deviation of the per-seed paired deltas. This rule is a fixed heuristic, not a calibrated significance level; effects near the threshold would be undecidable under it. The headline ablation result is not near the threshold ($12\times$), but smaller reported differences should be read with this coarseness in mind.

Finally, the scope of the evidence is narrow along several axes: one network, one blockage event of fixed type, severity, duration, and position. A second model and an incident-aware rule-based baseline were not evaluated and are the first items of future work. Some parts of the method do not depend on the grid: the blockage site is selected by a static demand-to-storage analysis applicable to any network (Section III-B), and each controller observes only its own approaches, with no global topology. The prompt and the incident report, however, are written for four-approach intersections with three lanes per approach and a four-phase plan. Applying them to T-junctions, five-leg or irregular intersections, other phase plans, or left-hand-traffic cities would require redefining the phase and approach vocabulary. Grid networks also offer parallel routes around a blocked link, and the value of the report may differ where such alternatives are scarce. Other benchmark networks with different sizes and demand, such as Jinan and New York [4], are a natural next test; whether the effect transfers across networks, event types, models, and prompt formulations remains an open empirical question. Real-time operation is also not demonstrated: with our model and hardware, decision latency exceeds the control interval (Section IV).

VII. Conclusions

This study investigated whether a structured incident report, rendered in the controller's observation vocabulary, changes the behavior of a zero-shot LLM traffic signal controller during an unexpected lane blockage. We tested this with a +text/−text ablation on the Hangzhou 4×4 network under a single severe blockage. The event text measurably improved the LLM controller's performance, but both LLM arms trail the trained RL baseline (CoLight). In this configuration, most of the gap to RL is present without the blockage, so it is not attributable to the event channel; whether it stems from zero-shot use, observation scope, or training cannot be resolved here. These findings carry several limitations. The experiments cover one zero-shot LLM, one network, and one blockage scenario. Because the incident report is inserted only in the +text arm, the two arms differ in prompt length as well as content. The ablation also cannot separate the value of having the event information from the value of the LLM interpreting it; an information-matched classical control (a conventional controller given the same event facts through a hand-coded rule) is scoped to future work. Finally, with three seeds per condition, we report verdicts against a pre-specified seed-noise threshold rather than formal significance tests.

Future work should examine whether fine-tuning can bring the LLM controller closer to trained RL baselines while preserving its ability to respond to described events. A second direction is to pair the LLM with an established controller, either supplying event information to it or selecting among vetted control policies, and to compare this against the information-matched classical control described in Section VI. Larger seed counts and a broader set of networks and blockage scenarios would enable formal statistical testing and show how far the effect of event text generalizes. Adding length-matched neutral text to the −text arm would remove the prompt-length confound.

Acknowledgment

This work was supported by the Center of Excellence in Digital Earth and Emerging Technology (CoE: DEET). It was also partially conducted under the National Research Council of Thailand (NRCT) through the Hub of Talents in AI and Emerging Technology (AI-NEXT). The first author acknowledges the Excellent Foreign Students (EFS) scholarship awarded by the Sirindhorn International Institute of Technology (SIIT), Thammasat University.

References

[1] M. Hassan, T. D. Turjo, A. H. Rambe, M. Jusoh, M. K. Islam, M. M. Hafizur Rahman, M. S. Alam, and M. M. Rahman, "A comprehensive survey of adaptive traffic signal control: Methods, applications, challenges, and future research," *Archives of Computational Methods in Engineering*, Apr. 2026. [Online]. Available: http://dx.doi.org/10.1007/s11831-026-10574-y

[2] M. Korecki, D. Dailisan, and D. Helbing, "How well do reinforcement learning approaches cope with disruptions? the case of traffic signal control," *IEEE Access*, vol. 11, pp. 36 504–36 515, 2023. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2023.3266644

[3] F. Rodrigues and C. L. Azevedo, "Towards robust deep reinforcement learning for traffic signal control: Demand surges, incidents and sensor failures," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, Oct. 2019, pp. 3559–3566. [Online]. Available: http://dx.doi.org/10.1109/ITSC.2019.8917451

[4] H. Wei, G. Zheng, V. Gayah, and Z. Li, "A survey on traffic signal control methods," *arXiv preprint arXiv:1904.08117*, 2019.

[5] X. Yang, Y. Yu, Y. Feng, and W. Y. Ochieng, "Improving the urban transport system resilience through adaptive traffic signal control enabled by decentralised multiagent reinforcement learning," *Journal of Advanced Transportation*, vol. 2024, no. 1, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1155/2024/3035753

[6] Q. Wang, S. Wei, and K. Yang, "LLMs as virtual traffic police: Incident-aware traffic signal control augmented by large language models," in *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Nov. 2025, pp. 2242–2247. [Online]. Available: http://dx.doi.org/10.1109/ITSC60802.2025.11423524

[7] 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," *IEEE Transactions on Vehicular Technology*, pp. 1–14, 2026. [Online]. Available: http://dx.doi.org/10.1109/TVT.2026.3674284

[8] M. Movahedi and J. Choi, "The crossroads of LLM and traffic control: A study on large language models in adaptive traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 2, pp. 1701–1716, 2025.

[9] S. Lai, Z. Xu, W. Zhang, H. Liu, and H. Xiong, "LLMLight: Large language models as traffic signal control agents," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, ser. KDD '25. ACM, Jul. 2025, pp. 2335–2346. [Online]. Available: http://dx.doi.org/10.1145/3690624.3709379

[10] X. Fu, H. Yu, H. Jiang, M. Li, Z. Cui, and Y. Ren, "LLM-enabled universal traffic signal control across different intersections and traffic flows," *Knowledge-Based Systems*, vol. 340, p. 115663, May 2026. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2026.115663

[11] J. Yao, J. Li, X. Xu, C. Tan, K. H. Yap, and R. Su, "Incorporating vision-based artificial intelligence and large language model for smart traffic light control," *Applied Soft Computing*, vol. 179, p. 113333, Jul. 2025. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2025.113333

[12] T. Boonchuntuk, T. Buaprakhong, V. Sithisint, A. Phusaensaart, S. Wilke, T. Ganokratanaa, and M. Ketcham, "Large language model-driven traffic signal optimization for reducing energy consumption and urban pollution," *Energy Engineering*, vol. 123, no. 5, pp. 1–10, 2026. [Online]. Available: http://dx.doi.org/10.32604/ee.2026.069005

[13] X. Li, Q. Guo, J. Chen, Z. Guo, S. Xu, L. Li, and L. Zhang, "Retrieval augmented generation-enhanced distributed LLM agents for generalizable traffic signal control with emergency vehicles," in *2025 IEEE 31st International Conference on Parallel and Distributed Systems (ICPADS)*, 2025, pp. 1–8.

[14] J. Yan, D. Li, and Q. Yang, "LLM-TrafficBrain: An information-centric framework for dynamic signal control with large language models," in *2025 IEEE 26th China Conference on System Simulation Technology and its Applications (CCSSTA)*. IEEE, Jul. 2025, pp. 252–256. [Online]. Available: http://dx.doi.org/10.1109/IEEECONF65522.2025.11137082

[15] Z. Mao, J. Li, N. Zheng, K. Tei, and S. Honiden, "Transfer learning method in reinforcement learning-based traffic signal control," in *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*. IEEE, Oct. 2021, pp. 304–307. [Online]. Available: http://dx.doi.org/10.1109/GCCE53005.2021.9621842

[16] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "MetaLight: Value-based meta-reinforcement learning for traffic signal control," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1153–1160.

[17] B. R. Naick and K. D. Babu, "A context-aware deep swarm and federated meta-reinforcement learning signal controller for sustainable urban mobility optimization," *International Journal of Engineering Trends and Technology*, vol. 74, no. 5, pp. 524–539, May 2026. [Online]. Available: http://dx.doi.org/10.14445/22315381/IJETT-V74I5P133

[18] G. K. Rattan, Z. Mao, J. Li, M. Zhang, and K. Tei, "Incremental reinforcement learning with prioritized sweeping for traffic signal control," in *2023 IEEE 8th International Conference on Intelligent Transportation Engineering (ICITE)*, 2023, pp. 21–28.

[19] Q. Yang, Z. Xie, H. Wei, D. Zhang, and Y. Yang, "MalLight: Influence-aware coordinated traffic signal control for traffic signal malfunctions," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, ser. CIKM '24. ACM, Oct. 2024, pp. 2879–2889. [Online]. Available: http://dx.doi.org/10.1145/3627673.3679605

[20] Z. Zeinaly, M. Sojoodi, and S. Bolouki, "A resilient intelligent traffic signal control scheme for accident scenario at intersections via deep reinforcement learning," *Sustainability*, vol. 15, no. 2, p. 1329, Jan. 2023. [Online]. Available: http://dx.doi.org/10.3390/su15021329

[21] F. Rasheed, K.-L. Alvin Yau, R. Md Noor, and Y.-W. Chong, "Deep reinforcement learning for addressing disruptions in traffic light control," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 2225–2247, 2022. [Online]. Available: http://dx.doi.org/10.32604/cmc.2022.022952

[22] W.-L. Shang, Y. Chen, X. Li, and W. Y. Ochieng, "Resilience analysis of urban road networks based on adaptive signal controls: Day-to-day traffic dynamics with deep reinforcement learning," *Complexity*, vol. 2020, pp. 1–19, Nov. 2020. [Online]. Available: http://dx.doi.org/10.1155/2020/8841317

[23] D. T. Thai and Q. T. Minh, "A real-time adaptive traffic signal control framework using graph neural networks and multi-agent reinforcement learning with crowdsourced event integration," in *Intelligent Systems and Data Science*. Springer Nature Singapore, Oct. 2025, pp. 270–285. [Online]. Available: http://dx.doi.org/10.1007/978-981-95-3358-9_20

[24] H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "CoLight: Learning network-level cooperation for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, ser. CIKM '19. ACM, Nov. 2019, pp. 1913–1922. [Online]. Available: http://dx.doi.org/10.1145/3357384.3357902

[25] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, and Z. Li, "Learning phase competition for traffic signal control," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1963–1972.

[26] H. Mei, X. Lei, L. Da, B. Shi, and H. Wei, "LibSignal: An open library for traffic signal control," *Machine Learning*, pp. 1–37, 2023.

[27] C. Sun, J. Hao, X. Pei, Z. Zhang, and Y. Zhang, "A data-driven approach for duration evaluation of accident impacts on urban intersection traffic flow," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1354–1359.

[28] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X13001782

[29] Qwen Team, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115