European Transport Research Review

ORIGINAL PAPER Open Access

A framework for benchmarking traffic signal control robustness under incidents: comparative study of reinforcement learning-based methods

Dang Viet Anh Nguyen[1*], Carlos Lima Azevedo[1], Tomer Toledo[2] and Filipe Rodrigues[1]

Abstract

Reinforcement learning-based traffic signal control (RL-TSC) has emerged as a promising approach for improving urban mobility. However, its robustness under real-world disruptions such as traffic incidents remains largely underexplored. In this study, we introduce T-REX, an open-source, SUMO-based simulation framework for training and evaluating RL-TSC methods under dynamic, incident scenarios. T-REX models realistic network-level performance considering drivers' probabilistic rerouting, speed adaptation, and contextual lane-changing, enabling the simulation of congestion propagation under incidents. To assess robustness, we propose a suite of metrics that extend beyond conventional traffic efficiency measures. Through extensive experiments across synthetic and real-world networks, we showcase T-REX for the evaluation of several state-of-the-art RL-TSC methods under multiple real-world deployment paradigms. Our findings show that while independent value-based and decentralized pressure-based methods offer fast convergence and generalization in stable traffic conditions and homogeneous networks, their performance degrades sharply under incident-driven distribution shifts. In contrast, hierarchical coordination methods tend to offer more stable and adaptable performance in large-scale, irregular networks, benefiting from their structured decision-making architecture. However, this comes with the trade-off of slower convergence and higher training complexity. These findings highlight the need for robustness-aware design and evaluation in RL-TSC research. T-REX contributes to this effort by providing an open, standardized and reproducible platform for benchmarking RL methods under dynamic and disruptive traffic scenarios.

Keywords Reinforcement learning, Traffic signal control, Incident, Robustness

*Correspondence:
Dang Viet Anh Nguyen
andng@dtu.dk
[1]Department of Technology, Management, and Economics, Technical University of Denmark (DTU), Kongens Lyngby 2800, Denmark
[2]Faculty of Civil and Environmental Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel

1 Introduction

Traffic congestion continues to be a global challenge for urban mobility, causing economic losses and reduced quality of life. In 2024, Istanbul drivers faced the highest global delays, losing an average of 105 hours in traffic, a 15% increase from the previous year. In New York City and Chicago, motorists lost 102 hours, amounting to over $1,800 in lost productivity per driver. In total, U.S. drivers lost more than four billion hours and $ 74 billion annually to congestion [1]. Effective traffic signal control is crucial for improving traffic flow, reducing delays, and enhancing overall mobility.

Traditional traffic signal control methods, including fixed-time, actuated, and adaptive systems such as SCOOT [2], SCATS [3], and TUC [4], have long been deployed in practice. While fixed-time control is simple, it lacks responsiveness to real-time fluctuations. Actuated control offers flexibility through local sensor input, but its limited coordination across intersections hinders overall network efficiency. Adaptive systems attempt real-time coordination but often struggle with large-scale disruptions and rapidly evolving traffic conditions [5].

Recent advancements in artificial intelligence have positioned reinforcement learning (RL) as a powerful alternative for traffic signal control (RL-TSC). RL agents optimize signal policies by interacting with traffic environments and learning long-term strategies that enhance network-wide performance [6]. Unlike traditional methods, RL-TSC is adaptive to real-time dynamics and capable of discovering complex temporal patterns, making it well-suited for managing urban congestion and emissions. The growing availability of traffic data and computing infrastructure supports RL-TSC as a promising, scalable solution.

However, despite its theoretical strengths, RL-TSC remains largely undeployed in real-world urban networks [7, 8]. Most studies are constrained to simulated environments or historical datasets that do not capture real-time uncertainties. These simulation setups frequently assume stable traffic conditions, failing to reflect abnormal scenarios such as incidents, sensor errors, or dynamic rerouting behaviors [9]. This disconnect between controlled experiments and real-world complexities highlights the need for rigorous robustness evaluation in RL-based traffic control.

Simulation environments play a central role in evaluating traffic signal control methods prior to real-world deployment. Classical simulators such as SUMO [10] provide high-fidelity modeling of traffic dynamics, while specialized platforms such as CityFlow [11] enable scalable multi-agent reinforcement learning experiments. Recent extensions, such as CityFlowER [12], integrate machine learning models directly into simulation to enhance behavioral realism, and frameworks like LibSignal [13] and LibSignal++ [14] support standardized benchmarking and sim-to-real evaluation. However, these platforms primarily focus on simulation efficiency, learning integration, or benchmarking, and do not explicitly address robustness under incident-driven, network-level disruptions, which remains largely underexplored.

In computer science, robustness refers to a system's ability to perform reliably under atypical or adverse conditions [15]. In control theory, it implies insensitivity to parameter variations in a closed-loop system [16]. In reinforcement learning, robustness entails coping with environmental uncertainty while using all available information [17]. In urban traffic systems, uncertainties arise from fluctuating demand, sensor errors, accidents, weather, and other unpredictable disruptions. These changes can significantly alter traffic dynamics, posing major challenges for RL-TSC systems, which often fail to generalize outside their training distribution.

We define:

Definition 1 Robustness in Reinforcement Learning-Based Traffic Signal Control (RL-TSC) is the system's ability to maintain stable and efficient performance under dynamic, uncertain, and adversarial conditions in the traffic network and infrastructure. A robust RL-TSC system should adapt to and generalize across diverse and unpredictable scenarios, including traffic demand fluctuations, incidents, sensor failures, and environmental disruptions, while ensuring safety, efficiency, and stability in signal control decisions.

A growing body of research has investigated ways to enhance robustness in RL-TSC. Since RL methods rely on accurate traffic state information, sensor failures caused by hardware faults, data loss, or detection noise can significantly disrupt policy performance. The more complex the state representation, the more sensitive the RL controller becomes to such failures. Several studies tackle this issue directly. Rodrigues and Azevedo [18] simulate sensor failures and use state aggregation, dropout-based training, and historical patterns to improve the resilience of traffic signal control at a single intersection. Shi et al. [19] introduce RGLight, which leverages graph neural networks (GNNs), distributional RL, and policy ensembles to maintain stability under incomplete or noisy inputs. Jiang et al. [20] propose a dual-policy system that switches between a standard RL controller and a cyclic fallback policy depending on observation quality. Other works adopt transfer learning and generalization strategies to improve adaptation to degraded or uncertain sensor inputs [21–23, 24].

Beyond sensing issues, demand fluctuations introduce additional stochasticity. Daily patterns, events, and unexpected congestion can cause non-stationary traffic distributions. Rodrigues and Azevedo [18] demonstrate

RL-TSC resilience to sudden demand surges at a single intersection, provided that the model has been exposed to such scenarios during training. Shi et al. [19] extend this to larger networks (e.g., Luxembourg and Manhattan), confirming improved performance under traffic volatility compared to baseline RL-TSC approaches.

More disruptive are traffic incidents, which can block lanes or intersections, triggering sharp shifts in traffic flow. Rodrigues and Azevedo [18] model incidents by halting vehicles or reducing lane availability. Their findings show that RL policies trained with incident exposure are more resilient than conventional baselines. Zeinaly et al. [25] simulate vehicle stoppages at varying distances and confirm that RL agents can maintain learning stability. Similarly, Aslani et al. [26] use the AIMSUN simulator to model lane blockages in Tehran and find that actor–critic methods, an advanced form of RL, outperform traditional approaches such as Q-learning and SARSA in managing such disruptions. However, most of these incident models remain localized, affecting only a single intersection or link, and ignore the network-wide spillover effects typical of real-world incidents. Disruptions often propagate due to rerouting, queue spillback, and congestion shockwaves, creating a systemic performance collapse across multiple intersections. Current RL-TSC research rarely addresses this full-network challenge. Moreover, while sensor failures and demand variations can be easily introduced as noise or data masking, realistic incident modeling causes distributional shifts in both traffic patterns and driver behavior, presenting a unique and underexplored challenge.

To address this gap, we introduce **T-REX** (**T**raffic signal control framework for **R**obustness **E**valuation under e**X**treme scenarios), a benchmarking framework for evaluating RL-TSC methods under disruptive traffic events. T-REX is designed to simulate network-wide disruptions by incorporating state-of-the-art traffic models that capture dynamic rerouting, speed variations, and lane-changing behaviors in response to incidents. By providing a reproducible and open-source platform, T-REX facilitates the comprehensive evaluation of the learning stability, robustness, and transferability of RL methods under various traffic uncertainties. While previous RL studies often rely on diverse evaluation protocols and metrics, this lack of standardization has led to inconsistent results and limited reproducibility [27]. T-REX addresses this critical limitation by offering a principled and standardized framework for the reliable assessment of RL-TSC methods.

Our key contributions are summarized as follows:

- We present T-REX, an open-source framework that simulates traffic networks under incident scenarios, enabling systematic, network-level evaluation of control robustness.
- We define a unified set of robustness metrics tailored to RL-TSC, extending beyond conventional traffic efficiency indicators to capture multiple dimensions of control robustness.
- We benchmark several state-of-the-art RL-TSC methods and show that robustness is highly context-dependent, with explicit incident awareness being essential for generalization across different network settings.

The remainder of this paper is organized as follows. Section 2 describes the proposed T-REX framework, detailing its architecture and key modules designed to simulate vehicle behavior under incident conditions. Section 3 outlines the traffic networks, RL-TSC methods evaluated, parameter configurations, and the robustness metrics used. Finally, Sect. 4 presents the experimental setup and results of our comparative analysis, focusing on learning performance, testing performance and generalization, transferability and adaptation of RL-TSC methods.

2 Methodology

2.1 Problem definition

Traffic signal control in urban networks involves dynamically selecting signal phases at intersections to minimize vehicle delays and congestion. The control decisions rely on data collected from traffic sensors, which provide partial and local observations of the current traffic conditions. The problem becomes more complex in the presence of incidents (e.g., lane blockages), which disrupt normal traffic patterns and introduce uncertainty in traffic flow.

We model traffic signal control under incident scenarios as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), defined by the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \Omega, O, \gamma, \mathcal{N} \rangle$, where $\mathcal{N}$ is the set of signalized intersections, each controlled by a decentralized agent. At each time step t, the global traffic state is $s_t \in \mathcal{S}$, and each agent receives a local observation $o_t^i = O(s_t)$ based on its intersection's traffic conditions and current phase. Each agent selects an action $a_t^i \in \mathcal{A}^i$, a predefined signal phase, applied for a fixed duration.

Traffic incidents are introduced at randomly selected edges $e^i \in E$, blocking lanes $l \subseteq L_i$ approaching intersection . The incident duration d and start time t_{start} are drawn from predefined distributions $\mathcal{D}$ and $\mathcal{U}$, respectively. The transition function $P(s_{t+1}|s_t, a_t)$ models the evolution of the traffic network, capturing both regular flow and incident-related disruptions, such as rerouting, speed reduction, and lane-changing behavior. These dynamics are simulated using the T-REX framework,

providing a realistic and data-driven representation of incident impacts.

The reward function $R(s_t, a_t)$ defines the immediate feedback received after taking joint action a_t in state s_t, reflecting traffic performance objectives such as minimizing congestion, delay, or queue length. Ω denotes the observation space from which each agent's local observation o_t^i is drawn, and $\gamma \in [0, 1)$ is the discount factor that balances immediate and future rewards.

Each agent aims to minimize congestion and delay through the reward function R. The goal is to learn an optimal joint policy:

$$\pi(a_t|s_t) = \prod_{i \in \mathcal{N}} \pi_\theta^i(a_t^i|o_t^i), \tag{1}$$

that maximizes the expected discounted return:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi}\left[\sum_{t=0}^{\infty} \sum_{i \in \mathcal{N}} \gamma^t r_t^i\right], \tag{2}$$

where θ_i denotes the parameters of agent 's policy, and τ is the trajectory over states and actions. Since learning a centralized policy over the joint state-action space suffers from the curse of dimensionality, we instead learn a decentralized policy $\pi = \{\pi_i\}_{i \in \mathcal{N}}$, where each agent independently maximizes its own long-term cumulative reward.

In this work, the above Dec-POMDP formulation serves as the underlying decision-making model implemented within the T-REX framework. T-REX provides the simulation environment that instantiates the transition dynamics P, observation function O, and reward function R through realistic traffic modeling. This integration allows reinforcement learning agents to be trained and evaluated under diverse incident scenarios, facilitating systematic assessment of robustness and generalization across network conditions.

2.2 Overall framework

T-REX is designed as a complementary framework that explicitly models incident-driven, non-stationary disruptions and enables systematic robustness evaluation at the network level. It operates as an offline simulation platform for analyzing traffic signal control under incident scenarios, supporting the training and evaluation of control policies, particularly reinforcement learning-based methods, by incorporating realistic disruption mechanisms such as lane blockages, probabilistic rerouting, and speed adaptation. Beyond RL, T-REX also enables the evaluation of alternative traffic management strategies, sensor configurations, and control logics in a controlled and reproducible environment. In this study, we use T-REX to systematically train and evaluate RL-TSC methods under incident conditions. A schematic overview of the framework is provided in Fig. 1.

Built on the open-source SUMO simulator [10], T-REX comprises four main modules: *Network Environment*, *Initializer*, *Deployment*, and *RL Interaction*. The *Network Environment* module forms the SUMO-based simulation core, initializing the scenario with network layout, vehicle routes, signal plans, and vehicle attributes. The *Initializer* defines incident parameters, either user-defined or randomly generated, including location, duration, blocked lanes, and start time. The *Deployment* module injects incidents into the simulation and adjusts vehicle behavior accordingly, such as rerouting, lane changes, and speed adaptation. The *RL Interaction* module enables real-time communication between agents and the simulation, transmitting traffic states and rewards while receiving control actions (signal phases).

The modular architecture of T-REX facilitates extensibility and integration with external data sources. Users can define new incident types by specifying attributes such as affected edges, blocked lanes, severity levels, start times, and durations through configuration files or Python interfaces. The incident-generation component supports both deterministic and stochastic parameters, enabling diverse scenario definitions without modifying the core framework. This modular design also enables T-REX to be adapted to other traffic simulators with compatible interfaces, as the incident modeling and evaluation pipeline are decoupled from the underlying simulation engine.

In addition, T-REX can be coupled with external data feeds via the TraCI interface or custom APIs, allowing real-time event injection or replay of historical incident records. For example, traffic management systems or data services can dynamically trigger incidents during simulation based on real or synthetic data streams. While specific numerical results may vary depending on simulator characteristics and calibration, the qualitative findings regarding robustness under incident-driven, non-stationary disruptions are expected to generalize across simulation environments. This design ensures that T-REX serves not only as an offline benchmarking platform but also as a foundation for online, data-driven traffic management and digital-twin applications.

2.3 Incident definition and deployment

Traffic incidents, such as collisions, roadwork, or severe weather, are unexpected events that disrupt normal traffic flow by blocking part of a road for a certain duration. Although SUMO is widely used for traffic simulation, it lacks built-in support for modeling such incidents. Existing workarounds typically involve stopping vehicles, altering traffic light phases, or reducing road capacity

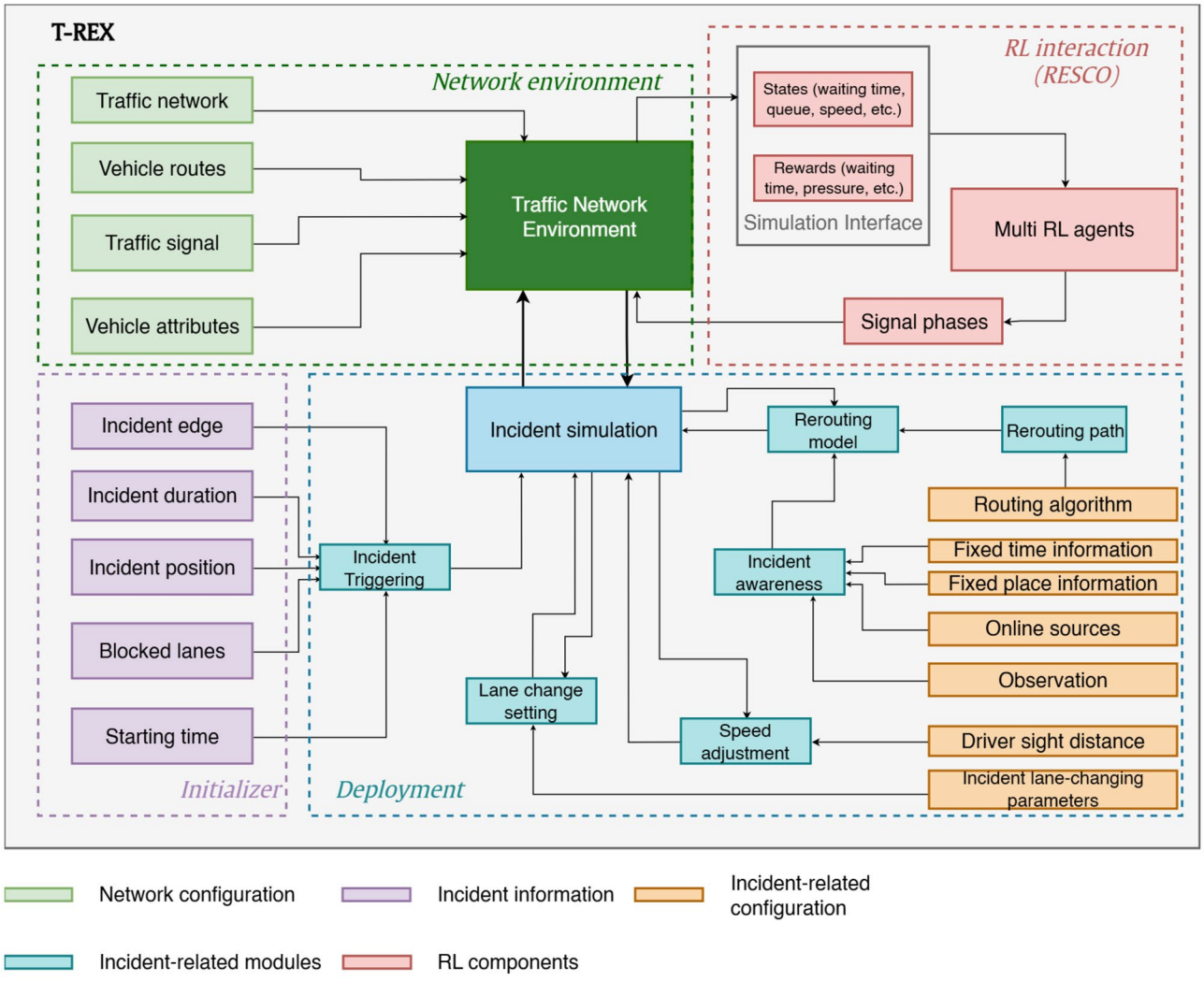


Fig. 1 Schematic overview of the T-REX framework

or speed limits [28, 29]. However, these approaches are limited: capacity or speed reductions lack realism, traffic light manipulation cannot simulate mid-edge disruptions, and manually defining affected areas becomes impractical for large or randomized networks. Furthermore, stopping vehicles using predefined routes fails to support stochastic or on-the-fly incident generation, making it unsuitable for RL training. Fixed-duration incidents also fail to capture the variability of real-world clearance times.

To address these limitations, we propose an online incident deployment method using TraCI (Traffic Control Interface). Incident parameters are defined in the *Initializer* module, supporting both user-defined and random generation. Each incident is specified by its edge $e^d \in E$, position p, blocked lanes $l \subseteq L_i$, start time t_{start}, and duration d. For randomized deployment, the number of blocked lanes is sampled from a uniform distribution $l \sim \mathcal{U}(1, l_n)$, where l_n is the number of lanes on edge e^d, and the position p is sampled uniformly within the range $(10,\ x_e)$, where x_e is the edge length, and the 10-meter buffer at each end prevents bugs in SUMO. The duration d is sampled from a distribution $\mathcal{D}$, and t_{start} is sampled uniformly from $\mathcal{U}(t_{\text{warmup}}, t_{\text{end}} - 1200)$ ensuring incidents start after warm-up and have sufficient time to affect traffic before the simulation ends. Multiple incidents can be configured to occur simultaneously.

Incident activation is managed by the *Deployment* module, which introduces a flexible and precise mechanism for simulating lane-blocking events. At the specified start time t_{start}, special "IC" vehicles are inserted at position p on edge e^d to block designated lanes. If a vehicle is already present, it is teleported to ensure consistency. This approach provides exact control over incident timing and location, guarantees immediate blockage, and supports randomized yet reproducible incident scenarios during RL training.

As mentioned in Sect. 2.2, this study focuses on applying T-REX for training and evaluating RL-TSC methods, where incidents are generated randomly. However, T-REX can also be used to simulate and evaluate a wide range of traffic management strategies at the network level under incident conditions. Four major categories of urban road incidents can be simulated by customizing the configuration of T-REX, as listed below:

- Vehicle accidents/collisions: simulated as full or partial lane blockages caused by disabled vehicles;
- Stalled or broken-down vehicles: represented by stationary obstacle vehicles on specified lanes;
- Roadworks or planned closures: represented as extended multi-lane blockages with configurable duration;
- Environmental or infrastructure failures: including speed-reduction zones (e.g., fog, heavy rain) and traffic signal malfunctions simulated by fixed-phase or blackout control at selected intersections.

Details of possible configurations and the involved modules for each incident type are provided in Appendix B.

2.4 Simulating vehicle behavior under incidents

2.4.1 Rerouting model

Existing studies on RL-TSC under incident scenarios generally do not incorporate vehicle rerouting into their simulations. In earlier work on developing a SUMO-based incident simulation,Smith et al. [28] proposed a rerouting mechanism. However, their rerouting logic is simplistic, as it reroutes any vehicle upstream of the incident edge if that edge appears in its planned route. This approach does not accurately reflect real-world rerouting behavior. In reality, rerouting is a far more complex phenomenon. Drivers' awareness of an incident is influenced by various information sources, such as radio, navigation systems, road signage, and direct observation. The timing and decision to reroute also vary greatly among individuals. Neglecting this complexity in simulation limits the realism of traffic dynamics under incident scenarios, which can lead to misleading conclusions about the robustness and effectiveness of RL-TSC methods.

To more realistically simulate rerouting behavior in our framework, we adopt the Information Comply Model (ICM) [30] and implement it in SUMO using TraCI. The ICM is built on three key assumptions. First, drivers only consider rerouting once they become aware of the incident, which depends on both direct observation and external information sources. Second, drivers choose to reroute to avoid the negative consequences of the event and to benefit from expected lower travel costs. Third, when rerouting, drivers aim for an optimal path based on their estimates of actual and future travel costs, which may be influenced by other rerouting drivers, past experiences, or predictive traffic information.

The ICM models driver behavior using two components: an awareness model, which determines when and how drivers become aware of an incident, and a rerouting model, which governs the decision-making process for selecting an alternative route.

The awareness model describes the mechanisms and information sources that inform drivers about unexpected events. These sources include fixed-time information (FTI), such as radio broadcasts and periodic TMC updates; fixed-place information (FPI), such as roadside Variable Message Signs (VMS); online sources (OS), such as social media, websites, or mobile applications; and direct observation (OB) of abnormal traffic conditions.

Once drivers become aware of an unexpected event, the rerouting model is used to estimate the probability that they will choose to deviate from their original route. This probability is derived using a binomial logit model, where the utility of rerouting is determined by the expected gain and avoided loss associated with the decision.

Further details on the ICM components are provided in Appendix A.

2.4.2 Speed adaptation model

When a vehicle enters an edge affected by an incident, drivers typically reduce their speed due to safety concerns, weather conditions, and behavioral responses such as rubbernecking [31]. Visibility, road geometry, weather, and traffic density influence the extent of this slowdown. In our simulation, we model this behavior through speed adaptation based on the driver's visibility of the incident ahead.

We propose a novel method based on the concept of Stopping Sight Distance (SSD) from road design guidelines [32] to determine the point at which a driver initiates deceleration in our simulation. In this context, SSD refers to the total distance a vehicle needs to come to a complete stop, accounting for both the perception-reaction distance and the braking distance. The perception-reaction distance (m) is computed as:

$$d_{\mathrm{PRT}} = 0.278 \times V \times t,$$

where V is the vehicle speed in km/h and $t = 2.5$ seconds, the standard reaction time recommended by AASHTO. The braking distance (m) is given by:

$$d_b = 0.039 \times \frac{V^2}{a},$$

with $a = 3.4\,\mathrm{m/s^2}$, the recommended deceleration rate. These values are standard AASHTO defaults,

representing conservative assumptions suitable for safety-oriented modeling. The total stopping sight distance is then:

$$\mathrm{SSD} = d_{\mathrm{PRT}} + d_b.$$

Unlike conventional road design, where SSD is determined from the design speed, our framework computes SSD dynamically based on each vehicle's actual speed at the moment it enters a route segment affected by an incident. This reflects realistic traffic conditions in which vehicle speeds vary due to congestion, signals, or geometry. Speed adaptation is activated when the vehicle's distance to the incident location becomes less than or equal to its computed SSD, representing the point at which the driver can visually perceive the obstruction and begins to decelerate. In other words, SSD serves as both the visibility distance and the trigger threshold for initiating speed reduction.

To model vehicle behavior near the incident, we incorporate principles from Move Over laws, which require drivers to reduce speed or change lanes near stationary roadside vehicles [33]. We implement a conservative reduced speed of 5 mph (approximately 8 km/h) when a vehicle is within SSD of the incident, representing cautious traversal of the affected edge.

2.4.3 Contextual lane-changing

In T-REX, we enhance the realism of lane-changing behavior for vehicles affected by incidents by modifying SUMO's LC2013 model [34] via TraCI. Specifically, vehicles are modeled to make early, strategic lane changes in anticipation of downstream obstructions, reflecting efforts to maintain route adherence and avoid blocked lanes. The model also accounts for speed-seeking behavior, where drivers shift to adjacent lanes perceived to offer better flow conditions. Additionally, cooperative interactions are included, allowing vehicles in neighboring lanes to yield space for merging vehicles when appropriate. The preference for keeping to specific lanes (e.g., the rightmost lane) is relaxed to prioritize incident avoidance. These adaptations are dynamically triggered based on a vehicle's proximity to an incident: lane-changing parameters are modified when a vehicle enters the SSD range of a disabled vehicle and revert to default once the vehicle has passed. This enables more anticipatory and context-aware lane-changing behavior during disruptions. Detailed settings for the lane-changing model are presented in Appendix B.

2.5 Interacting with RL-TSC agents

As discussed in Sect. 2.2, this study presents a specific application of T-REX as an environment for training and evaluating RL-TSC methods in the presence of traffic incidents. To enable such learning, it is essential to have a framework that allows RL agents to interact with the traffic simulation, observing states, taking actions, and receiving rewards. In this work, we adopt RESCO (REinforced Signal COntrol) [35] to facilitate communication between the simulation environment and RL-TSC algorithms. RESCO is a benchmarking toolkit developed on top of SUMO and integrated with the OpenAI Gym interface, enabling standardized training and evaluation of RL-based control strategies. Specifically, at predefined simulation intervals, RESCO allows the RL agent to receive traffic-related observations via TraCI commands. Based on these observations, the agent selects an action, which is then applied through TraCI, after which the simulation advances by a fixed number of steps. The agent subsequently receives a reward based on traffic performance metrics, observes the new state, and updates its policy accordingly. This interactive loop is repeated over multiple episodes, allowing the agent to progressively learn and refine effective traffic signal control strategies under incident scenarios.

3 Computational Experiment

3.1 Traffic networks

To evaluate the robustness of RL-TSC methods at the network level using T-REX, we focus exclusively on multi-intersection traffic networks. Specifically, our experiments are conducted on one synthetic network and four real-world networks. The synthetic grid network, adopted from Chen et al. [36], is a 4×4 intersection layout. The real-world network scenarios are based on those used in the experiments of Ault and Sharon [35], which draw from two well-established SUMO benchmark scenarios: TAPAS Cologne [37] and InTAS [38]. From TAPAS Cologne, we use two configurations: the Cologne corridor, consisting of three intersections, and the Cologne region, which comprises eight intersections. From InTAS, we use the Ingolstadt corridor, with seven intersections, and the Ingolstadt region, which includes 21 intersections. These scenarios represent realistic urban traffic networks in two German cities, Cologne and Ingolstadt, featuring actual road layouts and calibrated traffic demand. Their inclusion enables us to benchmark the performance of RL-TSC methods under realistic conditions and compare them with prior studies conducted in non-incident scenarios. Details of the traffic intersections and corresponding network layouts are provided in Appendix B.

3.2 RL-based traffic signal control methods and baselines

We evaluate a diverse set of RL-TSC methods, covering both independent and coordinated learning paradigms for network-level control. Each intersection is modeled as an agent, with decentralized training preferred due

to scalability constraints. The selected RL-TSC methods include:

- Independent Deep Q-Network (IDQN) is a multi-agent RL method in which each agent independently controls a single intersection without sharing information with other agents. Convolutional layers are used to aggregate lane-level traffic state features, as described in Ault et al. [39].
- Independent Proximal Policy Optimization (IPPO) follows the same decentralized architecture as IDQN but utilizes a policy gradient approach (PPO) instead of value-based Q-learning. Including IPPO enables a comparative analysis between value-based and policy gradient methods under incident scenarios, two foundational paradigms in reinforcement learning.
- MPLight [36] is a decentralized RL method that employs pressure-based coordination. It incorporates queue pressure into the agent's state representation, allowing partial observability of neighboring intersections. MPLight calculates Q-values for each signal phase based on the phase competition concept derived from the FRAP architecture [40]. Parameter sharing across agents also enhances its scalability in large-scale traffic networks.
- Feudal Multi-agent Actor-Critic (FMA2C)Ma and Wu [41] extends the state-of-the-art MA2C framework [42],) by introducing hierarchical coordination via feudal reinforcement learning. In this architecture, a set of intersection-level agents (workers) operate under the guidance of regional agents (managers), enabling coordination through both local learning and top-down goal setting.

These methods are chosen for their diversity in learning strategies and coordination levels. All have been validated under non-incident settings [35,36,39,41], and we adopt their open-source implementations for consistency. Further details on the MDP (Markov Decision Process) of each RL-TSC method are presented in Appendix C.

We also include four traditional rule-based baselines:

- Fixed-time control, where phase ordering and durations follow a predefined cycle, based either on real-world traffic signal plans or existing benchmark settings (e.g., for the 4×4 grid from Chen et al. [36]).
- Random control, which randomly selects a signal phase at each decision step, providing a naive baseline for comparison.
- Max-pressure control, similar to the implementation in Chen et al. [36], selects the phase with the maximum pressure, defined as the difference between upstream and downstream queue lengths.
- Greedy control, as used in Ma and Wu [41], selects the phase with the highest wave, where the wave is measured by the number of approaching vehicles at the intersection.

These rule-based baselines provide interpretable and practical benchmarks for assessing RL-based methods. They help determine whether the added complexity of RL is justified and highlight where RL offers advantages, such as adaptability to dynamic traffic patterns and resilience to disruptions.

3.3 Simulation scalability and running time analysis

To evaluate the computational efficiency and scalability of the T-REX framework, we conducted a runtime analysis across multiple benchmark networks under both normal and incident conditions (two incidents per episode). The evaluation was performed using the MaxPressure control policy to ensure a consistent and lightweight signal control strategy. Each simulation was executed for one hour of simulated time and repeated ten times to reduce stochastic variability arising from vehicle generation and random incident initialization. Table 1 reports the average runtime, real-time factor (RTF), simulation speed (steps/s), and Average Vehicle Time in Simulation

Table 1 Average simulation runtime and scalability across different network scenarios using the MaxPressure controller (1-hour simulation, averaged over 10 runs)

Scenario	Condition	N	Vehicles	Runtime (s)	RTF	Steps/s	AVTS (s)
Grid4×4	Base	16	1516	1.580	2278.000	227.816	161.574
Grid4×4	Incident	16	1516	3.009	1196.317	119.630	214.877
Cologne Corr.	Base	3	2855	1.648	2183.436	218.344	75.38
Cologne Corr.	Incident	3	2855	5.316	677.232	67.723	226.13
Cologne Reg.	Base	8	2045	0.862	4175.635	417.564	108.775
Cologne Reg.	Incident	8	2045	2.993	1202.614	120.263	175.622
Ingolstadt Corr.	Base	7	2220	1.895	1900.154	190.015	79.992
Ingolstadt Corr.	Incident	7	2220	3.448	1044.040	104.404	110.681
Ingolstadt Reg.	Base	21	4269	12.319	292.222	29.222	572.173
Ingolstadt Reg.	Incident	21	4269	21.563	166.951	16.695	638.067

(AVTS), which reflects both traffic demand and congestion impacts in each scenario.

Results show that T-REX scales efficiently with both network complexity and vehicle density. In smaller networks such as the *Cologne Region* (8 intersections, 2,045 vehicles), the framework achieved an RTF above 4,000 in the base condition, requiring less than one second of wall-clock time per simulated hour. For medium-scale configurations such as the *Grid4×4* (16 intersections, 1,516 vehicles) and *Ingolstadt Corridor* (7 intersections, 2,220 vehicles), runtimes remained below 2 s in the base condition, maintaining near real-time execution performance.

As expected, introducing incidents increases runtime due to the additional computational overhead from rerouting, lane-changing, and speed adaptation behavior. Across all networks, incident scenarios roughly doubled or tripled the runtime compared to the base cases, with RTF values decreasing from above 2,000 to around 1,000 on average. For example, in the *Grid4×4* scenario, runtime increased from 1.58 s (RTF 2,278) to 3.01 s (RTF 1,196) when two lane-blocking incidents were introduced. A similar pattern was observed in the *Ingolstadt Region* (21 intersections, 4,269 vehicles), where runtime increased from 12.32 s (RTF 292) to 21.56 s (RTF 167).

Since SUMO updates each active vehicle at every simulation step, computational load depends not only on the number of intersections and generated vehicles, but also on how long vehicles remain in the network. The AVTS therefore reflects the number of vehicle updates required per episode. We observe that scenarios with higher AVTS values correspond to longer wall-clock runtimes and lower RTF values, particularly under incident conditions where congestion propagation increases dwell time inside the network. This relationship confirms that runtime grows with the dynamic vehicle count over time, not merely static network or demand size.

Overall, the scalability trends demonstrate that T-REX's runtime grows approximately linearly with the number of controlled intersections and sub-linearly with total vehicle demand. Even in the largest tested configuration, the simulation remained computationally tractable, requiring less than 25 seconds of wall-clock time to process one hour of traffic conditions. These findings confirm that T-REX can efficiently support both small- and medium-scale network studies, making it suitable for large-scale reinforcement learning training and benchmarking. Future work will explore distributed simulation and GPU-based rollout computation to further enhance runtime scalability for extensive urban networks and real-time control applications.

3.4 Evaluation metrics

Previous studies on RL-TSC have primarily focused on conventional traffic metrics such as queue length, travel time, and delay. While RL-TSC methods often outperform traditional approaches on these metrics, the comparisons are not always aligned with real-world deployment challenges [8]. Notably, many studies report best-case performance during training, which may not reflect reliability or robustness. Strong simulation results do not guarantee real-world effectiveness, especially for RL algorithms that rely heavily on exploration and are often sample inefficient.

In practice, real-world deployment of RL-TSC can follow three main approaches: pre-training (training fully in simulation), online learning (learning directly and exclusively in the real world), and pre-training with continuous learning (fine-tuning a pre-trained policy using real-time data). For online and hybrid approaches, the learning phase after deployment is critical, as real-time adaptation affects both traffic efficiency and safety. Therefore, the stability, convergence, and exploration cost during training must also be considered in addition to final traffic performance.

To capture these aspects, we evaluate learning behavior using a set of empirical metrics that quantify stability, convergence efficiency, and exploration cost. Each metric is computed using the *performance indicator*, a general traffic-level measure (e.g., average delay, queue length, waiting time, and travel time) that is consistently defined across all benchmarked algorithms. This choice is necessary because each method optimizes a different reward signal under distinct MDP formulations (see Appendix C); thus, using internal reward values would not allow for fair cross-method comparison. All reported results are averaged over five random seeds, using a moving average window (5 episodes for value-based methods and 30 for policy-gradient methods) to smooth stochastic fluctuations. We report the average over the top-performing episodes, specifically, the best 10 consecutive episodes for IDQN and MPLight, and the best 100 consecutive episodes for IPPO and FMA2C.

3.4.1 Learning Stability Index (LSI)

RL training curves are inherently stochastic due to random exploration, varying traffic demand, and incident occurrences. Such fluctuations represent the cost of exploration, which has direct implications for real-world safety and reliability during online deployment. The LSI quantifies the extent of these fluctuations, reflecting how consistently a method improves over time:

$$LSI = \frac{1}{T}\sum_{t=1}^{T}(I_t - I_{t-1})^2,$$

where I_t is the smoothed performance indicator at episode t and T is the total number of episodes.

A higher LSI indicates frequent or large variations in learning performance, which may correspond to instability and higher exploration cost. However, we note that LSI can also capture rapid performance changes associated with fast learning, and thus may, in some cases, conflate instability with steep improvement dynamics. Therefore, LSI should be interpreted in conjunction with the complementary metrics listed below, which together provide a more comprehensive and reliable assessment of learning behavior. By contrast, a lower LSI generally reflects smoother and more stable policy adaptation, which is desirable for online or hybrid learning scenarios.

3.4.2 Final Performance Deviation (FPD)

Deep RL methods often show non-monotonic learning behavior and unstable convergence [43]. The FPD measures how closely the final training performance aligns with the best performance achieved during training:

$$FPD = \frac{|I_{\text{final}} - I_{\text{best}}|}{I_{\text{best}}},$$

where I_{final} and I_{best} are the final and peak values of the smoothed learning curve. A low FPD implies that the algorithm retains its best-learned performance toward the end of training, while a high FPD indicates poor policy retention or regression. To reduce sensitivity to single-episode noise, FPD is computed from smoothed and averaged learning curves across multiple seeds.

3.4.3 Convergence Rate (CR)

The CR provides an empirical measure of how quickly the training process stabilizes. Since TSC environments are non-stationary, perfect convergence is rarely attainable; instead, CR reflects the time required for the learning curve to remain within a tolerance band of its final stable performance:

$$CR = \begin{cases} \frac{1}{T_c}, & \text{if } I_{\text{final}} < I_0, \\ -\frac{1}{T_c}, & \text{if } I_{\text{final}} > I_0, \\ 0, & \text{otherwise,} \end{cases}$$

where I_0 is the initial performance, I_{final} is the final performance, and T_c is the episode when performance remains within a threshold ϵ of the final value. The threshold ϵ is tuned empirically for each scenario to ensure robust detection despite stochastic noise. As with FPD, CR is evaluated using smoothed and averaged learning curves to avoid false threshold crossings due to random fluctuations.

3.4.4 Area Under the Learning Curve (AUC)

While final performance reflects asymptotic quality, the AUC measures overall learning efficiency:

$$AUC = \int_0^T I(t)\, dt.$$

A smaller AUC (for metrics where lower is better) indicates faster improvement and better sample efficiency. Unlike traditional RL evaluations where exploration cost is often disregarded, here AUC intentionally incorporates the trade-off between early-stage exploration and operational performance. This aligns with a practical perspective: in real-world TSC deployments, exploratory or random control actions can cause additional delays or even unsafe traffic situations, conditions that are unacceptable to traffic authorities Han et al. [8]. Hence, AUC penalizes algorithms that require extensive exploration to achieve good performance. To account for differences in convergence timescales between value-based and policy-gradient methods, we further report the Relative AUC (RAUC). RAUC quantifies the relative change in AUC of an RL-based TSC method under normal and perturbed conditions, thereby serving as a comparative metric for evaluating the robustness of different learning approaches.

3.4.5 Relative AUC difference (RAUC)

To assess robustness under network disruptions or incident scenarios, we compare AUC values in normal and perturbed conditions using the RAUC:

$$RAUC = \frac{AUC_{G'} - AUC_G}{AUC_G},$$

where AUC_G and $AUC_{G'}$ are computed in the nominal and perturbed environments, respectively. Lower RAUC values indicate smaller performance degradation and stronger robustness to external disturbances.

3.4.6 Performance Degradation Index (PDI)

Finally, to evaluate generalization across environments (e.g., simulation-to-reality transfer), we define the Performance Degradation Index (PDI) as:

$$PDI = \frac{I_{G'} - I_G}{I_G} = \frac{\Delta I_G}{I_G},$$

where I_G and $I_{G'}$ represent performance in training and testing environments. A higher PDI indicates greater performance loss when transferring to new or perturbed settings, capturing the generalization capability of the trained policy.

Robustness in traffic signal control is closely related to safety, particularly under incident scenarios where unstable or inefficient control may lead to unsafe traffic conditions. While the proposed metrics do not explicitly measure collision-level safety, they capture system-level effects that are closely associated with safety risks, such as congestion buildup, unstable policy adaptation, and prolonged exploration. Safety-aware RL approaches [44] explicitly enforce safety constraints and can be evaluated within T-REX to assess their robustness under disruption.

3.5 Experimental setup

We conduct a series of experiments to assess the performance of RL-TSC methods across multiple dimensions of robustness. These dimensions, learning performance, generalization, and transferability/adaptation, correspond to key evaluation aspects that are critical under different real-world deployment strategies of RL-TSC. Across all experiments, we consider multiple traffic networks and include non-learning baselines for benchmarking. Each traffic episode simulates 3,600 seconds, including a 100-second warm-up phase.

Experiment 1: learning performance. In this experiment, we evaluate the learning performance of four RL-TSC methods under both normal and incident scenarios across various traffic networks. While real-world validation remains the ultimate goal, simulation provides a controlled and reproducible environment to analyze the learning behavior and convergence of different RL-TSC methods, particularly under online training or continuous adaptation scenarios.

Value-based methods (IDQN, MPLight) are trained for 100 episodes, while policy-gradient methods (IPPO, FMA2C), which generally require more extensive training, are trained for 1,400 episodes. Performance is measured using average queue length, travel time, waiting time, and delay. Incident scenarios include two randomly generated disruptions per episode to induce congestion propagation, rerouting, and conflicting demand patterns.

Experiment 2: testing performance and generalization. This experiment investigates the ability of RL-TSC methods to generalize across environments without additional training, a scenario representative of pre-training deployment where real-world data is limited. We evaluate three methods - IDQN, MPLight, and FMA2C, by training them in one scenario and testing them in another without further adaptation (i.e., zero-shot generalization). Due to overfitting observed during training, IPPO is excluded from this experiment.

Specifically, we examine four training–testing combinations: (i) training and testing in the base (no-incident) scenario, (ii) training in the base scenario and testing in the incident scenario, (iii) training and testing in the incident scenario, and (iv) training in the incident scenario and testing in the base scenario. For each method, we select the best policy based on the lowest average travel time over ten consecutive training episodes and evaluate it directly in the target test scenario.

Experiment 3: transferability and online adaptation. This experiment evaluates how well RL-TSC methods adapt to dynamic conditions under a deployment strategy that combines pre-training with continued online learning. Each method is first trained in a base scenario without incidents and then transferred to an incident scenario, where training continues without model reinitialization.

IDQN and MPLight are trained for 100 episodes per phase (pre-transfer and post-transfer), while FMA2C is trained for 1,400 episodes per phase to account for its greater training complexity. The experiments are conducted on both the synthetic Grid 4×4 network and the real-world Ingolstadt Region network. We assess adaptation and resilience by observing changes in average travel time across the training and transfer phases.

4 Results and discussion

4.1 Experiment 1: learning performance

Table 2 presents the best performance achieved during training for both the base (normal) and incident scenarios. Bolded values indicate the best-performing RL-TSC method for each metric. Using travel time as the primary performance indicator, we compute the robustness metrics defined in Sect. 3.4, with the results summarized in Table 3. These metrics include LSI, FPD, CR, and RAUC. For all metrics, lower values indicate better performance.

Under normal conditions, RL-TSC methods generally outperform traditional baselines. For instance, in the Ingolstadt Region, IDQN, representing independent value-based learning, achieves 242.88 seconds in travel time, outperforming Fixed-Time (295.22s). This suggests that even without inter-agent coordination, well-structured local state representations and value-based updates can yield effective control. Meanwhile, FMA2C, which leverages hierarchical coordination through feudal learning, performs best in the Cologne Corridor (88.35s), showing that coordinated decision-making across layers enhances optimization in more complex urban environments.

However, this advantage does not consistently carry over to incident scenarios. In the Cologne Corridor, MPLight, representing decentralized coordination using pressure-based reward design, experiences significant performance degradation, with travel time increasing to 178.74 seconds, falling behind Greedy (157.74s) and Fixed-Time (116.94s). This suggests that while pressure-based mechanisms can facilitate implicit coordination under stable conditions, they may be less robust

Table 2 Performance of the investigated methods on training

Grid	Avg. Queue		Avg. Travel time		Avg. Waiting		Avg. Delay	
	Base	Incident	Base	Incident	Base	Incident	Base	Incident
Fixed-time	-	-	204.51	214.07	65.09	73.01	93.26	103.23
Random	2.56	2.85	242.67	255.75	101.09	112.23	132.18	144.89
Max-pressure	0.61	1.09	160.14	181.75	23.11	42.50	49.06	70.27
Greedy	0.30	0.39	145.41	160.78	10.96	24.28	34.51	49.42
IDQN	**0.35**	1.25	**145.79**	191.92	**12.49**	51.27	**34.25**	80.72
MPLight	0.60	**1.14**	160.38	**187.01**	21.97	**45.77**	49.06	**75.27**
IPPO	2.11	3.25	223.95	271.16	84.49	128.70	113.34	161.01
FMA2C	1.94	2.32	215.75	234.10	76.38	92.69	105.07	122.98
Cologne Cor.								
Fixed-time	-	-	76.23	116.94	24.89	62.88	38.85	79.64
Random	6.96	9.00	140.65	203.31	82.85	142.43	124.22	203.68
Max-pressure	4.38	7.95	128.94	190.10	78.98	138.62	95.32	165.51
Greedy	2.53	3.16	135.07	157.74	87.04	106.71	101.16	139.03
IDQN	**2.48**	**4.81**	91.60	150.16	38.13	95.99	61.69	132.75
MPLight	5.11	6.46	111.00	178.74	54.87	124.70	89.84	169.90
IPPO	4.20	7.30	114.53	189.66	62.04	134.47	82.41	172.51
FMA2C	2.89	4.85	**88.35**	**126.96**	**37.30**	**73.61**	**53.92**	**101.58**
Cologne Reg.								
Fixed-time	-	-	114.51	128.91	29.37	41.20	49.37	62.34
Random	4.69	4.81	166.97	177.91	62.32	70.77	102.30	115.03
Max-pressure	0.61	0.98	91.77	107.97	8.63	22.81	27.63	44.08
Greedy	0.30	0.34	84.89	93.29	4.69	11.65	20.80	29.72
IDQN	**0.39**	**0.81**	**86.37**	**101.60**	**5.96**	**17.50**	**22.34**	**37.58**
MPLight	1.01	1.10	99.27	107.34	13.98	20.50	35.12	42.88
IPPO	1.25	4.59	110.78	178.36	24.98	74.33	46.92	115.07
FMA2C	1.02	1.29	98.12	109.07	14.76	22.69	33.86	44.86
Ingolstadt Cor.								
Fixed-time	-	-	116.91	175.84	48.07	101.25	73.01	131.62
Random	4.42	5.28	126.44	156.56	52.15	79.54	94.50	135.97
Max-pressure	1.35	2.23	79.79	111.11	15.28	44.39	42.23	88.75
Greedy	1.81	2.19	100.58	120.19	34.60	53.10	77.38	105.83
IDQN	**0.74**	**1.97**	**72.43**	**103.34**	**10.22**	**37.02**	**35.29**	**75.94**
MPLight	1.39	2.95	78.78	125.89	16.53	59.54	46.43	108.18
IPPO	1.29	3.71	104.19	132.52	38.48	61.06	69.12	110.21
FMA2C	1.88	2.73	87.13	113.51	22.48	45.42	53.27	89.42
Ingolstadt Reg.								
Fixed-time	-	-	295.22	309.06	102.41	114.40	150.12	163.15
Random	4.46	4.61	326.63	349.36	124.66	146.50	192.79	218.46
Max-pressure	5.40	5.53	551.37	550.81	358.23	357.09	420.98	424.40
Greedy	2.26	2.50	301.70	334.42	114.37	142.79	159.59	192.27
IDQN	**1.67**	**1.91**	**242.88**	**263.05**	**60.72**	**76.27**	**106.78**	**127.88**
MPLight	3.14	3.36	284.65	306.57	92.14	110.18	147.40	169.86
IPPO	3.54	3.99	310.40	337.09	115.26	142.31	176.84	204.88
FMA2C	2.15	2.60	252.65	281.96	65.12	90.40	117.15	147.93

to sudden disruptions. Similarly, IPPO, as a representative of decentralized policy-gradient methods, underperforms Greedy in the Ingolstadt Corridor (132.52s vs. 120.19s in travel time), indicating that its slower convergence and higher variance may hinder responsiveness to incident-induced dynamics.

In the synthetic Grid4×4 network, all RL-TSC methods underperform relative to the Greedy baseline during incidents. For example, IDQN's travel time rises from 145.79 to 191.92 seconds, while Greedy increases only modestly from 145.41 to 160.78 seconds. his behavior may reflect the simplicity and regularity of the grid network, where

Table 3 Learning performance metrics of RL-TSC methods

Grid4×4	LSI		FPD		CR		AUC		RAUC
	Base	Incident	Base	Incident	Base	Incident	Base	Incident	
IDQN	7.330	814.661	0.034	0.342	0.014	0.067	17050.706	21152.862	24.059
MPLight	7.486	612.472	0.007	0.346	0.013	0.080	17979.397	21173.720	17.767
IPPO	875.889	3001.635	2.174	4.770	−0.0008	−0.001	665797.802	1241386.056	86.451
FMA2C	19.838	330.332	0.034	0.092	0.002	0.042	318017.798	351935.840	10.665
Cologne Cor.									
IDQN	7775.667	114.436	1.543	1.179	0.059	−0.013	13626.695	22531.990	65.352
MPLight	5695.477	140.097	1.373	1.257	−0.014	−0.012	17114.030	24729.504	44.498
IPPO	10657.108	12242.250	9.966	3.621	−0.007	−0.004	365325.553	219815.749	−39.830
FMA2C	8212.069	9967.432	0.200	1.621	0.002	0.023	199662.804	279639.156	40.056
Cologne Reg.									
IDQN	51.853	656.083	0.009	0.169	0.013	0.029	10278.780	12223.445	18.919
MPLight	511.220	676.036	1.435	1.173	−0.010	−0.010	13185.565	13442.965	1.952
IPPO	112.498	1711.564	0.192	2.045	0.031	−0.002	183600.283	528496.517	187.852
FMA2C	90.924	516.186	0.020	0.024	0.001	0.001	164846.319	190315.011	15.450
Ingolstadt Cor.									
IDQN	26.754	1840.784	0.017	0.579	0.013	0.333	8676.864	13636.753	57.162
MPLight	104.527	5903.952	0.040	2.457	0.015	−0.010	9182.465	19714.950	114.702
IPPO	62.338	4578.024	0.060	5.111	0.033	−0.002	151446.714	783816.170	417.552
FMA2C	99.902	2152.070	0.030	0.065	0.001	0.002	140259.031	205025.106	46.176
Ingolstadt Reg.									
IDQN	2742.280	3805.556	0.330	0.038	0.110	0.615	26980.021	29725.092	10.174
MPLight	6286.990	4468.380	2.051	0.615	−0.010	−0.013	41709.383	37294.750	−10.584
IPPO	4725.375	4386.292	4.140	1.716	−0.001	−0.002	1348850.585	1055946.293	−21.715
FMA2C	6918.328	3211.393	0.407	0.156	0.250	0.003	427084.614	467161.467	9.384

independent control and uniform topology limit the advantages of adaptive RL policies.

Robustness metrics reveal significant degradation in learning performance under incidents. IDQN, as an independent learner, suffers from the most volatile behavior: its LSI in Grid4×4 spikes from 7.33 to 814.66, and its RAUC reaches 24.06%, indicating instability in adapting to dynamic disruptions. IPPO exhibits a similar pattern, with a RAUC of 417.55% in the Ingolstadt Corridor, underscoring the challenges of sample inefficiency and sensitivity in policy gradient methods. Negative CR values, such as IDQN's drop from 0.059 to −0.013 in the Cologne Corridor, further indicate convergence failure in turbulent environments. These issues are also reflected in MPLight's performance, highlighting limitations of pressure-based coordination when facing non-stationary disruptions.

Among all evaluated methods, FMA2C consistently demonstrates the highest robustness. It maintains low LSI and FPD values and achieves the lowest RAUC in several networks - for example, only 9.38% in the Ingolstadt Region. Its hierarchical architecture allows for structured cooperation: high-level agents adapt strategy while low-level agents manage local actions, providing resilience to large-scale disturbances. However, this robustness is offset by slower convergence and sample inefficiency. For instance, FMA2C's CR values remain low across networks (e.g., 0.001 in the Cologne Region), reflecting the cost of learning coordinated behavior. This trade-off suggests that while hierarchical RL is well-suited for complex, disruption-prone networks, its use in real-time systems may be limited by the need for random exploration and sample inefficiency.

4.2 Experiment 2: testing performance and generalization

Generalization is quantified using the Performance Degradation Index (PDI), reported in Table 4, where columns denote training–testing scenario combinations. Detailed performance results are presented in Appendix D. While PDI enables method-level comparison across networks, it is less suited for comparing across training-testing pairs, since scenarios with incidents naturally incur longer travel times. To complement this, we also report each method's percentage improvement over the Max-pressure baseline under various conditions (Fig. 2). We use Max Pressure as the baseline in Experiment 2 because it is a well-established rule-based control policy known for its analytical stability guarantees Varaiya [45], and strong empirical performance in non-learning traffic signal control. It serves as a standard benchmark in RL-TSC studies Wei et al. [46], Zheng et al. [40], Chen et al. [36], Ma and Wu [41], offering a reliable reference for evaluating the robustness and generalizability of RL-based methods under traffic incidents.

Table 4 Performance degradation index of RL-TSC methods

Grid4×4	Base-base	Base-incident	Incident-incident	Incident-base	Average
IDQN	0.04	0.61	0.07	0.13	0.21
MPLight	−0.01	0.20	−0.06	−0.16	−0.01
FMA2C	0.14	0.24	0.05	0.04	0.12
Cologne Cor.					
IDQN	−0.18	0.36	0.93	−0.09	0.25
MPLight	0.66	1.92	0.02	−0.30	0.57
FMA2C	0.32	0.92	0.08	−0.21	0.28
Cologne Reg.					
IDQN	0.00	0.78	−0.03	−0.40	0.09
MPLight	−0.06	0.31	0.44	0.15	0.21
FMA2C	0.60	1.95	0.61	0.31	0.87
Ingolstadt Cor.					
IDQN	0.01	1.54	0.06	−0.50	0.28
MPLight	−0.18	0.52	0.17	−0.43	0.02
FMA2C	0.55	1.09	0.38	0.01	0.51
Ingolstadt Reg.					
IDQN	−0.04	0.11	0.50	−0.10	0.12
MPLight	0.63	0.65	0.62	0.64	0.63
FMA2C	0.41	0.40	0.12	0.20	0.28

The observed generalization patterns in our results can be interpreted through the lens of epistemic and aleatoric uncertainty in RL. Epistemic uncertainty, which arises from limited knowledge or insufficient data, reflects the model's uncertainty about the environment and can often lead to overfitting. Aleatoric uncertainty, in contrast, stems from inherent randomness or noise in the environment and cannot be reduced by collecting more data. Notably, Figure 3 reveals that policies trained in base scenarios in many cases outperform those trained directly under incident conditions when evaluated in the presence of disruptions. This counterintuitive finding can be partially attributed to the epistemic uncertainty inherent in incident-trained models. Since traffic incidents during training vary in location, severity, and timing, the agent may overfit to these specific conditions, leading to brittle policies that fail to generalize to unseen disruptions during testing. This form of model uncertainty, caused by limited and biased exposure to disruptions, limits the agent's ability to build a robust policy.

In contrast, policies trained in base scenarios may capture more stable and generalizable traffic patterns, allowing them to better handle moderate disruptions during testing - even without direct exposure to such conditions during training. This suggests that reducing epistemic uncertainty through broader or more diverse training experiences may be more beneficial than narrowly focusing on incident-specific conditions.

Additionally, the environment's aleatoric uncertainty, stemming from the inherent stochasticity of traffic behavior, such as variable vehicle arrival times or driver responses, remains irreducible and further challenges generalization. The poor transferability of incident-trained policies to base scenarios also underscores that without explicitly accounting for both types of uncertainty—e.g., through robust algorithmic design, regularization, or uncertainty-aware exploration strategies—RL-TSC methods struggle to maintain performance under train-test distribution shifts.

Method-specific analysis reveals important differences in how RL-TSC approaches handle uncertainty and generalize across scenarios. IDQN, representing the independent value-based learning paradigm, performs reliably when training and testing conditions are aligned (e.g., base–base or incident–incident), but its performance degrades sharply under distribution shifts. In Grid4×4, for example, IDQN improves travel time by 5.62% over Max-Pressure in the base–base setting but suffers performance drops of 57.41% and 23.2% in the incident–base and incident–incident cases, respectively. This sharp decline reflects high epistemic uncertainty, IDQN's reliance on detailed local state features enables it to model regular traffic dynamics effectively, but it also leads to overfitting to simulation-specific patterns. Although the 4×4 grid is relatively large, its uniform and symmetric structure limits environmental diversity, making it easier for the model to overfit. In contrast, irregular real-world networks such as Ingolstadt expose the agent to more heterogeneous topologies and rerouting conditions, improving generalization and reducing epistemic uncertainty, as reflected by the lower PDI values (0.11 for base–incident and 0.50 for incident–incident). This improvement is likely due to the greater structural diversity and rerouting flexibility in such networks, which expose the agent to a broader range of conditions during training and reduce epistemic uncertainty.

MPLight, which follows a decentralized coordination paradigm using pressure-based state and reward representations, exhibits more stable performance across scenarios. Its abstraction over raw traffic features, by focusing on upstream–downstream queue differences, effectively mitigates aleatoric uncertainty by filtering out low-level stochastic variations in traffic flow. In Grid4×4, MPLight maintains consistent performance across all training–testing combinations, achieving the lowest average PDI (−0.01). However, its use of parameter sharing and uniform logic across intersections becomes a limitation in heterogeneous networks like Cologne and Ingolstadt. In these settings, diverse traffic patterns and intersection geometries introduce epistemic uncertainty

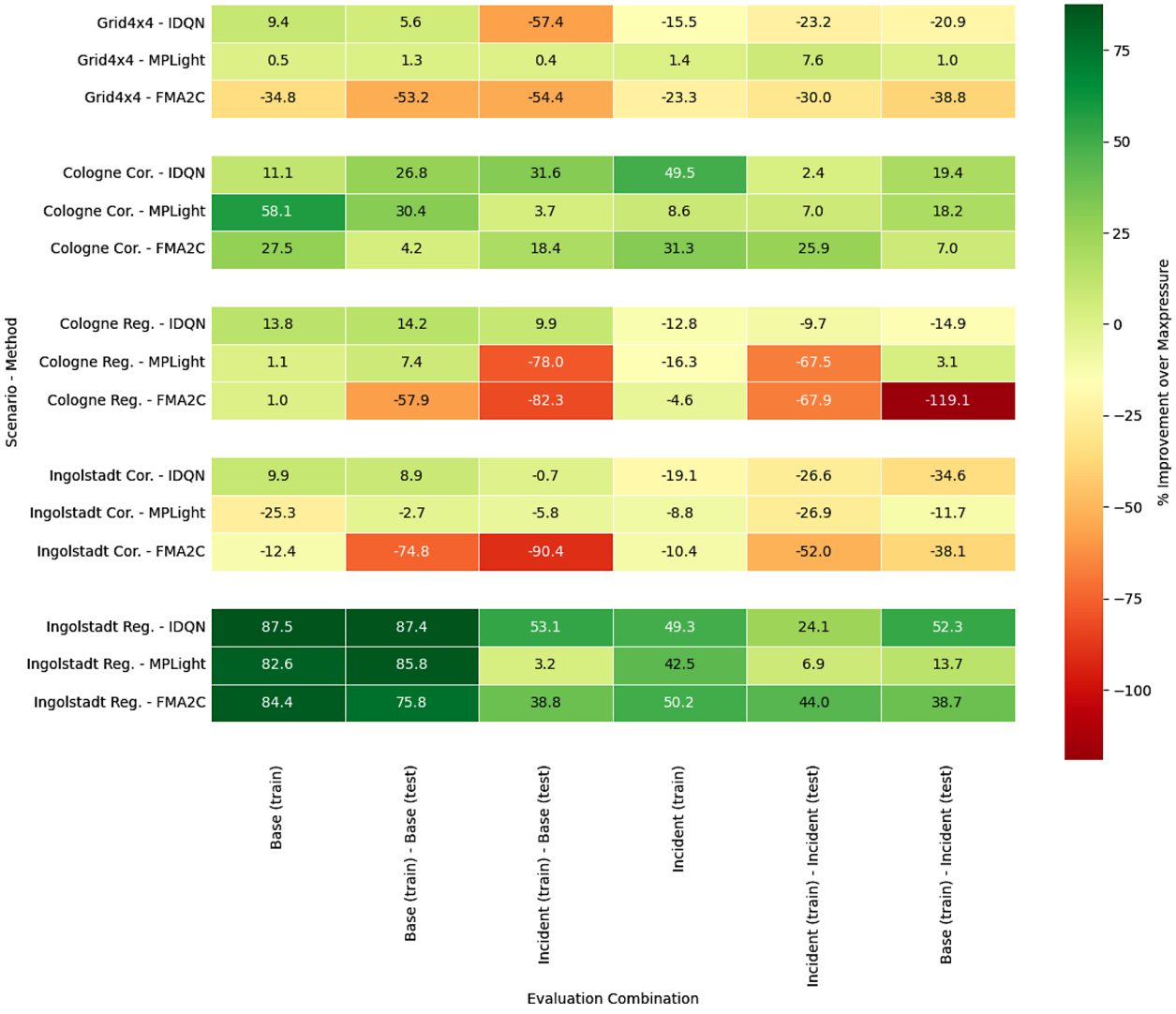


Fig. 2 RL-TSC performance improvement over Max-pressure

that MPLight's generalized design cannot fully capture, leading to higher degradation indices (e.g., 1.92 and 0.65, respectively).

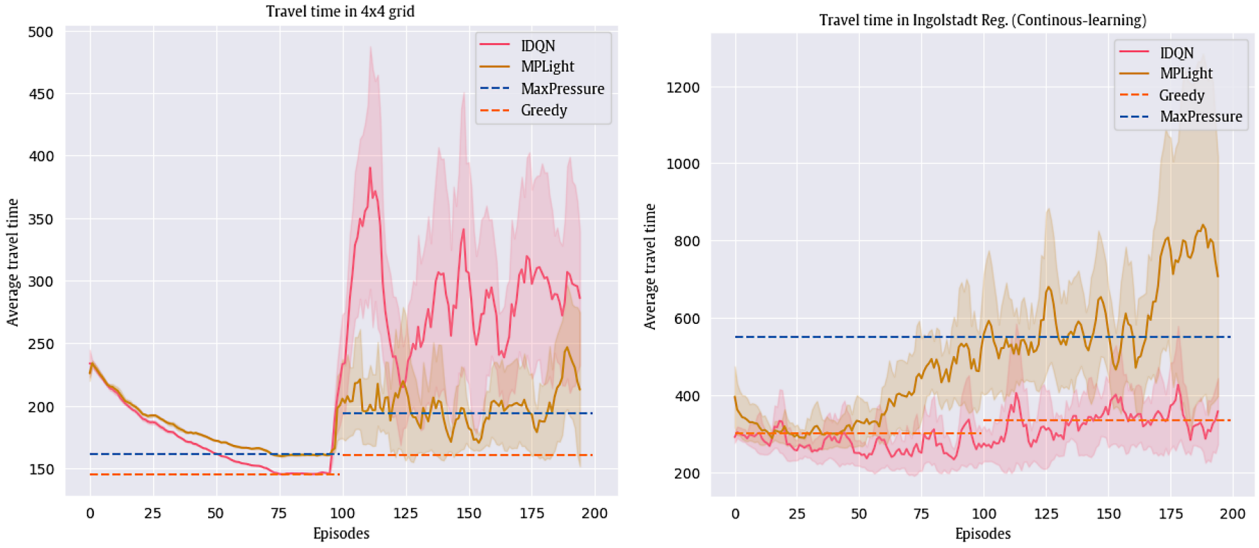
FMA2C, a hierarchical RL-TSC method employing manager–worker coordination, exhibits the most pronounced generalization challenges across multiple settings. In structured, mid-sized networks like Grid4×4 and Cologne Region, FMA2C struggles with epistemic uncertainty arising from the sensitivity of its learned coordination strategies to small shifts in regional traffic flows. This is reflected in consistently high PDI values (e.g., 1.95 in Cologne Region base–incident), indicating disrupted alignment between manager and worker agents. Moreover, the hierarchical structure is also vulnerable to aleatoric uncertainty, as stochastic variations can misalign the intended top-down coordination. Nonetheless, in large-scale, irregular environments such as the Ingolstadt Region, the hierarchical architecture aligns more naturally with real-world traffic patterns, allowing FMA2C to achieve more stable performance (e.g., PDI of 0.12 in incident–incident, 0.20 in incident–base). These results suggest that while hierarchical coordination offers scalability and structure, its effectiveness depends heavily on network characteristics and requires careful adaptation to manage both types of uncertainty.

Our findings highlight the need to better handle both epistemic and aleatoric uncertainties in RL-based TSC, especially when aiming for robust real-world deployment across varying traffic conditions.

4.3 Experiment 3: transferability and adaptation

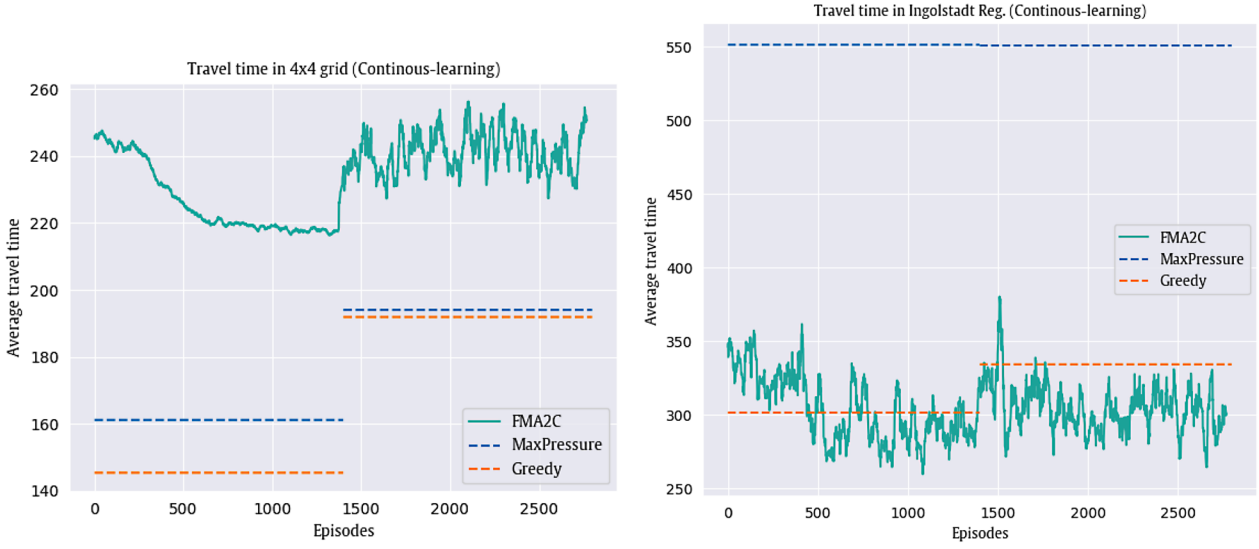
Figures 3 and 4 illustrate the average travel time per episode across both the training and transfer phases in Experiment 3.

In the Grid 4×4 network, all three RL-TSC methods exhibit clear convergence during the initial training



(a) *Grid 4x4 network* (b) *Ingolstadt Region network*

Fig. 3 Performance of IDQN and MPLight in the transferability experiment (rolling average over a window of 5 episodes) The plots show the evolution of average travel time (y-axis) during training episodes (x-axis) in (**a**) The grid 4×4 and (**b**) The Ingolstadt Region networks. For both methods, the first 100 episodes correspond to training in the non-incident environment, after which the model is transferred and continuously trained for another 100 episodes in the incident environment. Solid lines represent RL-based controllers (IDQN, MPLight), and dashed lines indicate rule-based baselines (MaxPressure, Greedy). Shaded areas denote standard deviations across random seeds



(a) *Grid 4x4 network* (b) *Ingolstadt Region network*

Fig. 4 Performance of FMA2C in transferability experiment (rolling average over a window of 30 episodes) the figure shows the change in average travel time (y-axis) across training episodes (x-axis) for (**a**) The grid 4×4 and (**b**) The Ingolstadt Region networks. FMA2C is trained for 1400 episodes in the non-incident environment and then continuously trained for another 1400 episodes under incident conditions. Solid lines show the FMA2C learning curve, while dashed lines represent the MaxPressure and Greedy baselines. (**a**) Grid 4×4 network. (**b**) Ingolstadt Region network

phase in the base scenario. The convergence levels align with their respective learning paradigms. IDQN, as an independent value-based method, quickly approaches the performance of Greedy through simple exploitation of local patterns, while MPLight, using decentralized coordination via pressure, converges near Max-pressure. FMA2C, which relies on hierarchical coordination and requires more extensive sampling to align manager–worker behaviors, converges the slowest and underperforms, with an average travel time of 217.45 seconds

compared to 161.17 for Max-pressure and 145.12 for Greedy.

Upon transfer to the incident scenario, all methods experience a sharp performance drop, but the severity varies by paradigm. IDQN suffers the most substantial increase in travel time, from about 150 seconds to nearly 400 seconds, highlighting the fragility of methods that rely heavily on detailed traffic state representations when exposed to dynamic and previously unseen conditions. In contrast, MPLight and FMA2C show more moderate increases of approximately 60 and 25 seconds, respectively. This smaller spike for MPLight can be attributed to its pressure-based abstraction, which simplifies input features but also reveals overfitting tendencies during continued training. Meanwhile, FMA2C maintains a more stable (although underperforming) trajectory post-transfer, consistent with the higher complexity and sensitivity of hierarchical coordination structures. Notably, IDQN shows signs of adaptation through continued learning, gradually recovering from the initial shock. However, its average performance remains highly volatile. This high variance is partly due to the stochastic nature of incident generation, which introduces variability in traffic dynamics across episodes. However, it is primarily driven by the instability of independent learning in IDQN. Without coordination or shared policy mechanisms, each agent adapts locally to rapidly changing conditions, leading to inconsistent system-level behavior under non-stationary disruptions. In contrast, MPLight exhibits lower variance due to its pressure-based coordination and shared policy structure, while rule-based approaches such as Max-pressure remain stable given their deterministic control logic. Both MPLight and FMA2C diverge over time in this transfer setting, suggesting limited resilience in handling abrupt environmental changes without additional architectural or mechanisms to capture non-stationary dynamics.

In the Ingolstadt Region network, learning behaviors shift significantly due to the network's larger scale and structural diversity. Here, MPLight, despite its prior stability in synthetic settings, fails to converge even in the base scenario, and its performance worsens further post-transfer, highlighting the challenge of applying a shared-policy decentralized method in highly heterogeneous networks. IDQN performs more stably during base scenario training, achieving slightly better travel times than Greedy (293.21s vs. 301.70s), but fails to adapt effectively after transfer. Its performance fluctuates heavily under incident conditions, suggesting that while independent value-based learning may generalize somewhat in regular patterns, it struggles to re-optimize when encountering new, disruptive dynamics.

In contrast, FMA2C, representing hierarchical coordination, demonstrates strong adaptability in this larger, more complex setting. It converges gradually in the base scenario and continues improving post-transfer, despite an early performance spike. Its post-transfer performance stabilizes around 303.07 seconds, outperforming Greedy (334.42s) and showing both robustness and continued learning capacity. The architecture's ability to maintain regional strategy alignment and local responsiveness allows it to outperform other paradigms in this dynamic environment. These results affirm the potential of hierarchical methods to support ongoing adaptation in real-world networks - despite the cost of slower initial convergence and sampling inefficiency.

5 Conclusion

In this study, we introduced T-REX, an open-source and reproducible simulation framework designed to train and evaluate RL-TSC methods under incident conditions. Built upon the SUMO platform, T-REX incorporates key behavioral modules, including probabilistic rerouting, speed adaptation, and contextual lane-changing, to realistically model vehicle behavior in disrupted networks. These features enable T-REX to simulate complex, network-level incident dynamics with a high degree of fidelity, supporting the development and benchmarking of robust RL-TSC strategies.

To complement the simulation framework, we proposed a suite of robustness-oriented evaluation metrics that extend beyond conventional traffic efficiency indicators. These metrics provide a more comprehensive understanding of RL-TSC behavior under different real-world deployment paradigms, including pre-training, online learning, and pre-training with continuous learning. Using extensive experiments across both synthetic and real-world networks, we evaluated several state-of-the-art RL-TSC methods in terms of learning efficiency, generalization ability, and adaptability to dynamic traffic conditions, with a particular focus on incident scenarios.

The experimental results reveal that no single RL-TSC paradigm consistently dominates across all networks, scenarios, metrics, and robustness dimensions. Independent value-based methods like IDQN demonstrate fast convergence and strong performance under stable conditions but suffer from overfitting and instability when exposed to dynamic or unseen incident scenarios. This reflects high epistemic uncertainty, as IDQN's reliance on detailed local states makes it vulnerable to distributional shifts not seen during training. Decentralized pressure-based coordination methods such as MPLight offer better generalization in structured environments due to their abstracted state representation, which helps mitigate aleatoric uncertainty by filtering out stochastic fluctuations. However, they struggle to adapt in heterogeneous networks where local traffic patterns vary widely, revealing limitations in handling epistemic uncertainty.

Hierarchical coordination methods like FMA2C exhibit robustness and adaptability, particularly in large-scale or irregular networks, by leveraging structured top-down control. Yet, their performance in mid-sized structured networks is hindered by sensitivity to small regional flow changes, leading to coordination failures and elevated epistemic uncertainty. Moreover, this approach comes at the cost of slower convergence and requires substantially more training interactions with the environment to learn an effective policy. Overall, base-trained policies generalize better than incident-trained ones, suggesting that exposure to disruptions alone is insufficient without addressing the underlying uncertainty. Transferability remains challenging for all methods, highlighting the need for RL-TSC architectures that are explicitly designed to manage both epistemic and aleatoric uncertainties in dynamic real-world traffic environments.

These findings collectively indicate that current state-of-the-art RL-TSC methods are not yet ready for reliable real-world deployment, as they lack robustness to distribution shifts, struggle with out-of-distribution generalization, and fail to consistently handle incident conditions. Bridging this gap will require both architectural innovations and training strategies that explicitly promote generalization and robustness under uncertainty.

Furthermore, we analyzed the simulation scalability and running time of T-REX across multiple synthetic and real-world networks. The results demonstrate that runtime scales approximately linearly with network size and sub-linearly with vehicle density, with incident scenarios incurring higher computation time due to additional rerouting and behavioral adaptation. These findings confirm that T-REX remains computationally efficient and suitable for medium- to large-scale studies, providing practical insights for users in planning experiments.

Despite its contributions, the T-REX framework has certain limitations. Parameters governing driver awareness and response to incident information are currently inferred or assumed, in the absence of empirical datasets. Future work should aim to integrate more realistic behavioral data to improve the accuracy of rerouting and adaptation modeling. Moreover, while T-REX currently models driver-level rerouting through the Information Comply Model, it does not yet represent system-level dynamic rerouting and recovery processes, such as coordinated detour guidance, adaptive network rebalancing, or emergency response mechanisms. Extending T-REX to include these coordinated recovery dynamics would enable a more complete assessment of network resilience and post-incident recovery strategies. In addition, future extensions should enable seamless coupling with external traffic data feeds or live information systems to allow real-time event injection and data-driven incident triggering. Such integration would enhance T-REX's interoperability with digital-twin platforms and online adaptive control frameworks, broadening its applicability beyond offline benchmarking. T-REX can be extended to support broader robustness assessments, such as sensor failures, fluctuating demand, and adversarial conditions, thereby increasing its utility as a benchmarking platform for real-world RL-TSC deployment.

6 Reproducibility statement

To support reproducibility, we have made the full source code for the T-REX framework publicly available, including the simulation scripts, traffic network configurations, and the modified SUMO interface used to model incident dynamics. All RL-TSC baselines (IDQN, IPPO, MPLight, and FMA2C) are integrated using openly accessible implementations, with any modifications clearly documented.

Appendix B provides full details of our experimental setup, including the parameter configurations for the *Initializer* and *Deployment* modules, as well as all hyperparameter settings used during training and evaluation. This ensures that our results can be replicated under consistent experimental conditions.

We encourage the research community to use T-REX as a standardized platform for benchmarking RL-TSC methods under real-world disruptions, and to extend its capabilities for future robustness-oriented studies.

The full codebase and associated resources are publicly available at: https://github.com/MLSM-at-DTU/T-REX

Appendix A Rerouting model

A.1 Incident awareness model

Let $P_{\text{aware}}(e,t)$ denote the probability that a driver becomes aware of the event while traversing edge $e \in E$ and exiting at time t. This probability is modeled as a joint function of the complementary probabilities associated with each available source of information $i \in \{\text{FTI}, \text{FPI}, \text{OS}, \text{OB}\}$:

$$P_{\text{aware}}(e,t) = 1 - \prod_{i \in \{\text{FTI},\text{FPI},\text{OS},\text{OB}\}} \left(1 - P^{i}_{\text{aware}}(e,t)\right).$$

The probability of awareness through fixed-time information, $P^{\text{FTI}}_{\text{aware}}(e,t)$, is based on the number of times a driver is exposed to relevant broadcasts while traveling along edge e:

$$P^{\text{FTI}}_{\text{aware}}(e,t) = \rho^{\text{FTI}}(t + t_e) - \rho^{\text{FTI}}(t),$$

where $\rho^{\text{FTI}}(t)$ is the cumulative probability that a driver has become aware of the event via broadcasts up to time t:

$$\rho^{\mathrm{FTI}}(t) = \mu_{\mathrm{FTI}} \cdot \mu_{\mathrm{ON}} \cdot \sum_{i=1}^{\max(n:\, t \le t_n^{\mathrm{FTI}})} (1-\mu_{\mathrm{ON}})^{i-1}$$

with μ_{FTI} representing the market penetration of radio listeners, μ_{ON} the probability that a listener notices the event during a broadcast, and t_n^{FTI} the time of the n-th broadcast.

The probability of becoming aware through fixed-place information, $P_{\mathrm{aware}}^{\mathrm{FPI}}(e,t)$, depends on both spatial and temporal proximity to the information source:

$$P_{\mathrm{aware}}^{\mathrm{FPI}}(e,t) = \delta_e^{\mathrm{FPI}} \cdot B(t \ge t_{\mathrm{FPI}}) \cdot \frac{1}{1+\left(\frac{\ell(e,e^d)}{\ell_2}\right)^{\beta^{\mathrm{FPI}}}},$$

with

$$\ell(e,e^d) = ||e,e^d|| + \bar{\nu} \cdot ||\mathrm{Mid}(0, t_{\mathrm{start}} - t, t_{\mathrm{end}} - t)||.$$

In this formulation, δ_e^{FPI} is a binary indicator that equals 1 if the edge e is equipped with a VMS, and 0 otherwise. The function $B(\cdot)$ evaluates to 1 if the VMS broadcast has occurred before the driver passes the sign. The function $\ell(e,e^d)$ denotes the effective distance from the edge to the event location e^d, combining both spatial distance $||e,e^d||$ and a temporal component scaled by the average speed $\bar{\nu}$. The parameter ℓ_2 defines the maximum effective range of the VMS, and β^{FPI} controls the rate at which the probability decays with distance. The event duration is represented by t_{start} and t_{end}.

Awareness through online sources is modeled as a cumulative probability function, where the probability of becoming aware while traversing edge e is:

$$P_{\mathrm{aware}}^{\mathrm{OS}}(e,t) = \rho^{\mathrm{OS}}(t+t_e) - \rho^{\mathrm{OS}}(t),$$

and the cumulative probability of awareness at time t $\rho^{\mathrm{OS}}(t)$ is defined as:

$$\rho^{\mathrm{OS}}(t) = \mu_{\mathrm{OS}} \cdot \left[1 - \exp\left(-\frac{(t-t_{\mathrm{OS}})^2}{2\sigma^2}\right)\right].$$

where μ_{OS} denotes the proportion of drivers using online platforms, t_{OS} is the time at which the event was first published online, and σ is a spread-rate parameter governing how quickly the information propagates.

Finally, drivers may become aware of an event through direct observation of abnormal traffic conditions. The probability of observation-based awareness is modeled as:

$$P_{\mathrm{aware}}^{\mathrm{OB}}(t) = \mathrm{Mid}\left(0, 1, \xi_{\mathrm{OB}} \cdot (t_e - \hat{t}_e)\right) - t_e^{\mathrm{OB}},$$

where t_e is the actual travel time on edge e, $\hat{t}_e$ is the typical (pre-event) travel time, t_e^{OB} is the minimum delay required to trigger awareness, and ξ_{OB} is a sensitivity parameter that scales the effect of delay on awareness. The function $\mathrm{Mid}(0,1,\cdot)$ ensures that the resulting probability is bounded between 0 and 1.

A.2 Rerouting decision model

The expected gain, denoted by $\Delta p^i(t)$, captures the change in the utility of available routes by comparing actual edge transition probabilities to typical (pre-event) ones. This is computed based on the cosine similarity between the two probability distributions:

$$\Delta p^i(t) = 1 - \frac{\sum_{e\in E_i^+} p^e(t)\, \hat{p}^e(t)}{\sqrt{\sum_{e\in E_i^+}[p^e(t)]^2} \cdot \sqrt{\sum_{e\in E_i^+}[\hat{p}^e(t)]^2}},$$

where $p^e(t)$ represents the actual transition probability from edge to edge $e \in E_i^+$ at time t, while $\hat{p}^e(t)$ denotes the corresponding typical (pre-event) transition probability. The set E_i^+ contains all outgoing edges from edge toward the vehicle's destination.

The avoided loss, denoted by $\Delta w^i(t)$, quantifies the improvement in travel cost achieved by rerouting compared to following the typical route. This is calculated as the difference in expected edge potentials (i.e., cost-to-go values) weighted by the transition probabilities:

$$\Delta w^i(t) = \frac{\sum_{e\in E_i^+} p^e(t)\, w^e(t) - \sum_{e\in E_i^+} \hat{p}^e(t)\, w^e(t)}{\sum_{e\in E_i^+} \hat{p}^e(t)\, w^e(t)}.$$

In this formulation, $w^e(t)$ denotes the minimum expected cost to reach the destination from edge e at time t.

The total utility of rerouting, V_{reroute}, is modeled as a linear combination of expected gain and avoided loss:

$$V_{\mathrm{reroute}} = \beta_{\mathrm{gain}} \cdot \Delta p^i(t) + \beta_{\mathrm{loss}} \cdot \Delta w^i(t) + \beta_0,$$

where β_{gain} and β_{loss} are sensitivity parameters that represent the driver's responsiveness to gain and loss, respectively, while β_0 controls the general willingness to reroute.

The resulting rerouting probability at edge and time t, denoted by $\kappa^i(t)$, is given by the standard logistic function:

$$\kappa^i(t) = \frac{1}{1+\exp(-V_{\mathrm{reroute}})}.$$

A.3 Routing algorithm

When a driver decides to reroute, the new path must be computed based on the current network conditions. Although the `rerouteTravelTime()` function provided by SUMO via TraCI is intended to support dynamic rerouting, it is currently known to be unreliable in certain scenarios [47]. To address this limitation, our framework integrates three alternative online routing algorithms: greedy search, A*, and Dijkstra's algorithm, which provide flexible and accurate route selection under dynamic traffic conditions.

Appendix B Parameter settings and configuration

This section outlines the configuration of the *Initializer* and *Deployment* modules in T-REX, along with the hyperparameters used for RL-TSC methods.

Initializer module. Incident locations and durations are assigned based on network type. For the synthetic 4×4 grid, a uniform distribution over all edges is used. In real-world networks, we construct an empirical discrete distribution by simulating one hour of traffic and assigning incident probabilities proportional to observed edge-level traffic volumes. Incident durations follow an exponential distribution Exp(0.029), based on real-world incident data from [48].

Deployment module. Awareness and rerouting behaviors are modeled using multiple information sources:

- *Fixed-time information:* Market penetration $\mu_{FTI} = 0.7$, detection probability $\mu_{ON} = 0.5$, with broadcasts every 5 minutes.
- *Fixed-place information (VMS):* Deployed on 40% of edges, active 10 minutes post-incident ($t_{FPI} = t_{start} + 10$), with an effective range of 200 meters and spatial decay $\beta^{FPI} = 2$.
- *Online sources:* Penetration rate $\mu_{OS} = 0.8$, viral delay of 5 minutes, and awareness propagation rate $\sigma = 10$ minutes.
- *Observation-based awareness:* Triggered when delay exceeds 2 minutes ($t_e^{OB} = 2$), scaled by sensitivity $\xi_{OB} = 0.5$.

Rerouting model. We adopt the utility-based model from Kucharski and Gentile [30], with parameters $\beta_{gain} = 2.5$, $\beta_{loss} = 2.5$, and $\beta_0 = -5$. Drivers choose their next edge based on normalized congestion factors, while default route transitions are given probability $\hat{p}^e(t) = 1$.

Driver heterogeneity. We simulate four driver types with varying travel time estimation errors: experienced (40%, 5%), novice (30%, 10%), distracted (20%, 20%), and CAVs (10%, 1%). All other parameters follow SUMO defaults.

Lane-changing parameters: When a vehicle enters its Stopping Sight Distance (SSD) range of a disabled vehicle ahead, its lane-changing behavior is dynamically adjusted in real-time by setting the parameters `lcStrategic` = 1, `lcSpeedGain` = 1, `lcCooperative` = 1, and `lcKeepRight` = 0. Once the vehicle has passed the disabled vehicle, these parameters are reset to their default values.

RL-TSC methods. We evaluate IDQN, IPPO, MPLight, and FMA2C using open-source implementations from Ault and Sharon [35]. Default hyperparameters are adopted for IPPO and FMA2C, while IDQN and MPLight undergo additional tuning.

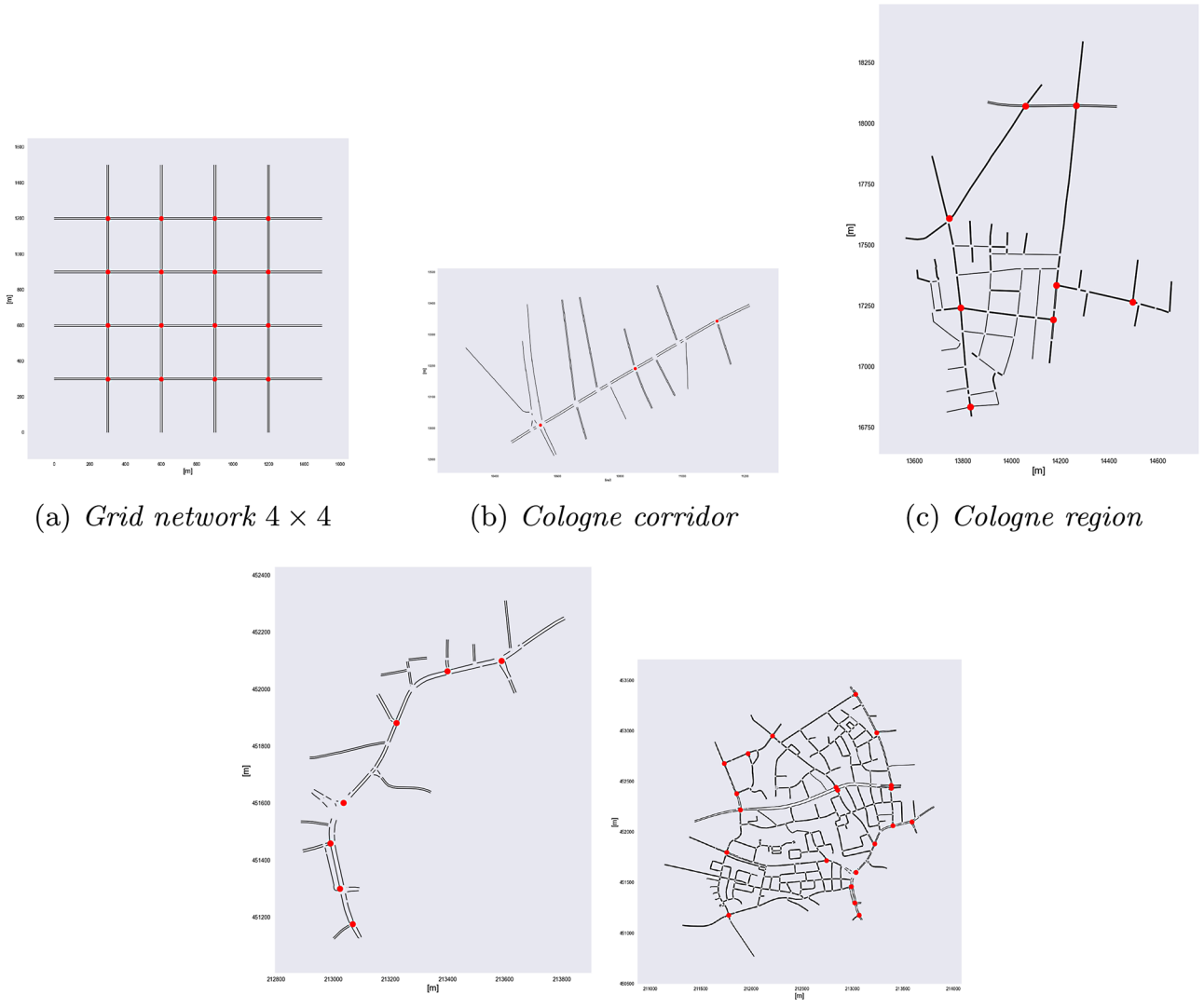
IDQN: We conduct a grid search over learning rates $\{0.01, 0.005, 0.001, 10^{-4}, 10^{-5}, 10^{-6}\}$ and discount factors $\{0.99, 0.95, 0.90\}$. A learning rate of 10^{-5} performs best for the 4×4 grid and Ingolstadt Region; 0.001 is optimal for other networks. Discount factor $\gamma = 0.99$ yields the best performance across all environments.

MPLight: Optimal learning rates are 0.005 for the grid, 0.01 for Ingolstadt Region, and 0.001 for others, with $\gamma = 0.99$.

Phase length. We test durations of 5 and 10 seconds and observe no significant performance differences. For consistency and training efficiency, we use a fixed 10-second phase length across all experiments [35,36].

Traffic network. Details of the traffic intersections for each benchmarked network are shown in Fig. B1.

Customizing Incidents. As mentioned in Sect. 2.3, the simulated incidents in T-REX can be customized to represent specific types of disruptions by adjusting the corresponding configuration parameters. Table B1 summarizes how different incident types are defined across various T-REX modules for specific simulation purposes.



(a) *Grid network* 4×4 (b) *Cologne corridor* (c) *Cologne region*

(d) *Ingolstadt corridor* (e) *Ingolstadt region*

Fig. B1 Benchmark scenarios

Table B1 Supported incident types and configurable parameters in T-REX

Incident Type	Description	Key Parameters	Configurable Range / Distribution	Module
Collision / Accident	Blocked lanes by inserted disabled "IC" vehicle(s)	Edge ID, blocked lane count, blockage position, duration, start time	Position $\sim U(10\,\text{m}, \text{edge\_length} - 10\,\text{m})$; duration $\sim D$; lanes $\sim U(1, l_n)$	Deployment
Stalled vehicle	Stationary obstacle vehicle with lower severity	Same as above but single-lane only	—	Deployment
Roadworks / Planned closure	Long-duration lane or full-edge closure	Duration (user-defined); affected lanes; start time	—	Initializer
Speed-reduction / Environmental	Reduced speed zone (e.g., rain, fog)	Affected edges, reduction ratio	0.3–0.8× normal speed	Speed Adaptation
Traffic-signal malfunction	Intersection operates in fixed-time / blackout mode	Intersection ID, duration, phase pattern	duration $\sim U(t_{start}, t_{end})$	Initializer / RL Interaction

Appendix C MDP formulation of RL-TSC methods

C.1 IDQN

Observation space (o^i): Composed of features extracted from the intersection, including for each lane:

- Current signal assignment (green/yellow/red for each phase),
- Number of approaching vehicles,
- Number of stopped vehicles,
- Accumulated waiting time of stopped vehicles,
- Average stop time,
- Average speed of approaching vehicles.

These are all non-negative and phase-dependent, with a convolutional layer grouping the inputs from lanes that belong to the same road.
Action space (a^i): The assignment of the next signal phase (e.g., selecting a non-conflicting phase set) must satisfy two constraints: (1) no conflicting traffic movements, and (2) a yellow phase must be inserted between red and green transitions.
Reward function (r^i): Reward is based on the reduction in accumulated traffic delay.
Policy and learning algorithm: Each agent employs Q-learning (DQN) to approximate its action-value function and selects the phase corresponding to the highest Q-value.

C. 2 IPPO

Similar to IDQN, but using PPO as the learning algorithm.

C. 3 MPLight

Observation space (o^i): Each agent observes:

- The current traffic signal phase
- The pressure values are defined for each movement through the intersection. Specifically, the movement pressure for (l, m) is given by $x_{(l,m)} = q_{\text{in}} - q_{\text{out}}$, where q_{in} and q_{out} are the queue lengths on the entering and exiting lanes, respectively. The phase pressure for a signal phase s is then computed as $p(s) = \sum_{(l,m)\in s} x_{(l,m)}$, representing the total pressure across all permitted movements in that phase.

Observations are padded if the intersection has fewer than 12 movements.
Action space a^i: Similar to IDQN.
Reward function r^i: The negative pressure of intersection is defined as $r^i = -P_i$, where P_i is the difference between the total entering and exiting queue lengths at intersection .
Policy and learning algorithm: DQN with FRAP architecture.
Algorithm design:

- Decentralized: Each intersection is controlled by an individual RL agent.
- Shared Parameters: All agents share the same model weights to scale efficiently across thousands of intersections.

C. 4 FMA2C

Traffic network hierarchy: Each intersection is controlled by an agent (worker). The network is partitioned into regions, each managed by a manager agent who sets high-level goals for its workers.
MDP for worker agent:
Observation o^i: Number of approaching vehicles and cumulative waiting time of the first vehicle at each lane.
Action space a^i: Similar to IDQN
Reward function r^i: Penalizes congestion and waiting time
MDP for manager agent:
Observation o^k: Aggregated flows in and out of the region (e.g., directional wave states at region boundaries)
Action a^k: A high-level goal (desired flow direction) for the region.
Reward r^k: Reflects global traffic throughput and liquidity in the region.
Learning algorithm: Uses Advantage Actor-Critic (A2C) for both manager and worker levels.
Information exchange: Each worker agent's input includes local and neighbor observations and fingerprints (policy distributions) of neighbors for stability.

Appendix D Performance of RL-TSC methods on testing environment

Table D1 presents the detailed performance of RL-TSC methods across different combinations of training and testing scenarios in the generalization experiment. The values reported in the table represent the average travel time.

Table D1 Performance of RL-TSC methods on different combinations of training and testing scenarios

	Base - Base		Incident - Base	Incident - Incident		Base - Incident
Grid4 × 4	**Train (last 10)**	**Test (avg. 100)**	**Test (avg. 100)**	**Train (last 10)**	**Test (avg. 100)**	**Test (avg. 100)**
Max-pressure	161.17±1.40			194.05±19.68		
IDQN	146.08±1.52	152.12±15.73	253.69±69.29	224.11±70.89	239.04±81.84	234.57±90.36
MPLight	160.38±1.28	159.03±1.55	160.60±1.50	191.27±23.44	179.33±25.28	192.02±46.43
FMA2C	217.33±2.39	246.90±6.13	248.92±6.83	239.26±20.49	252.30±31.88	269.40±27.98
Cologne Cor.						
Max-pressure	178.29±93.12			266.49±91.25		
IDQN	158.46±220.82	130.46±157.98	121.93±155.25	134.70±41.77	260.14±222.51	214.86±185.97
MPLight	74.79±4.54	124.17±149.12	171.63±203.16	243.60±133.95	247.96±179.14	218.07±180.63
FMA2C	129.22±130.39	170.89±143.56	145.46±92.49	183.11±139.71	197.52±136.84	247.89±177.75
Cologne Reg.						
Max-pressure	99.95±14.03			133.44±33.95		
IDQN	86.12±0.58	85.75±0.48	90.08±0.50	150.52±101.43	146.39±101.89	153.26±123.71
MPLight	98.84±2.81	92.59±0.67	177.88±48.78	155.25±84.28	223.53±112.81	129.32±69.40
FMA2C	98.97±1.83	157.86±11.18	182.20±19.44	139.52±63.85	224.02±75.40	292.32±84.70
Ingolstadt Cor.						
Max-pressure	80.62±2.30			137.14±43.41		
IDQN	72.64±1.02	73.41±1.95	81.16±3.63	163.40±95.14	173.61±128.51	184.58±167.66
MPLight	101.01±62.35	82.81±16.34	85.29±14.39	149.18±48.54	174.06±124.25	153.24±126.79
FMA2C	90.65±9.76	140.90±18.09	153.47±19.70	151.38±87.34	208.52±93.25	189.37±94.11
Ingolstadt Reg.						
Max-pressure	581.72±63.83			596.90±57.83		
IDQN	255.69±89.30	245.01±85.36	273.01±84.10	302.722±81.986	453.238±184.311	284.89±97.19
MPLight	311.94±76.59	507.77±114.44	563.39±123.11	343.379±46.331	555.62±156.836	515.20±116.72
FMA2C	260.81±36.63	367.42±61.23	356.02±79.09	297.13±71.23	334.26±90.61	366.08±63.04

Acknowledgements
The authors acknowledge support from the Independent Research Fund Denmark (Danmarks Frie Forskningsfond) under grant no. 2035-00200B. Tomer Toledo was partially supported by a grant from the Ministry of Science and Technology, Israel.

Author contributions
Conceptualization: DVAN, CLA, TT, FR; Methodology: DVAN, CLA, TT, FR; Software: DVAN; Formal analysis: DVAN; Visualization: DVAN; Writing – original draft: DVAN; Writing – review #x0026; editing: CLA, TT, FR; Supervision: CLA, TT, FR; Validation: FR; Funding acquisition: FR.

Funding
This work was supported by the Independent Research Fund Denmark (Danmarks Frie Forskningsfond) under grant no. 2035-00200B.

Data availability
The source code and data for the T-REX framework are publicly available athttps://github.com/MLSM-at-DTU/T-REX.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 16 June 2025 / Accepted: 18 August 2026
Published online: 16 September 2026

References

1. INRIX. (2024). 2024 global traffic scorecard. https://inrix.com/press-releases/2024-global-traffic-scorecard-us/. Accessed 17 Feb 2025.
2. Robertson, D. I., & Bretherton, R. D. (1991). Optimizing networks of traffic signals in real time-the scoot method. *IEEE Transactions on Vehicular Technology*, *40*(1), 11–15. https://doi.org/10.1109/25.69966
3. Sims, A. G., & Dobinson, K. W. (1980). The sydney coordinated adaptive traffic (scat) system philosophy and benefits. *IEEE Transactions on Vehicular Technology*, *29*(2), 130–137. https://doi.org/10.1109/T-VT.1980.23833
4. Diakaki, C., Papageorgiou, M., & Aboudolas, K. (2002). A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, *10*(2), 183–195. https://doi.org/10.1016/S0967-0661(01)00121-6
5. Agarwal, A., Sahu, D., Mohata, R., Jeengar, K., Nautiyal, A., & Saxena, D. K. (2024). Dynamic traffic signal control for heterogeneous traffic conditions using max pressure and reinforcement learning. *Expert Systems with Applications*, *254*, 124416. https://doi.org/10.1016/j.eswa.2024.124416
6. Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z. S. H., & Bazzan, A. L. C. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, *199*, 116830. https://doi.org/10.1016/j.eswa.2022.116830
7. Chen, R., Fang, F., & Sadeh, N. (2022). The real deal: A review of challenges and opportunities in moving reinforcement learning-based traffic signal control systems towards reality. arXiv:220611996.
8. Han, Y., Wang, M., & Leclercq, L. (2023). Leveraging reinforcement learning for dynamic traffic control: A survey and challenges for field implementation. *Communications in Transportation Research*, *3*, 100104. https://doi.org/10.1016/j.commtr.2023.100104
9. Haydari, A., & Yılmaz, Y. (2022). Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(1), 11–32. https://doi.org/10.1109/TITS.2020.3008612
10. Lopez, P. A., Behrisch, M., & Bieker-Walz, L., et al (2018). Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2575–2582). IEEE.
11. Zhang, H., Feng, S., & Liu, C., et al (2019). Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In *The World Wide Web Conference. WWW '19* (pp. 3620–3624). ACM. https://doi.org/10.1145/3308558.3314139

12. Da, L., Chu, C., & Zhang, W., et al. (2024). Cityflower: An efficient and realistic traffic simulator with embedded machine learning models. URL https://arxiv.org/abs/2402.06127, arXiv:2402.06127
13. Mei, H., Lei, X., Da, L., Shi, B., & Wei, H. (2024). Libsignal: An open library for traffic signal control. *Machine Learning*, *113*(8), 5235–5271. https://doi.org/10.1007/s10994-023-06412-y
14. Zhang, Y., Vo, K., & Da, L., et al (2025). Reproducible and low-cost sim-to-real environment for traffic signal control. In *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems (with CPS-IoT* Week 2025), pp 1–2.
15. Pullum, L. L. (2022). Review of metrics to measure the stability, robustness and resilience of reinforcement learning. arXiv preprint arXiv:220312048.
16. Åström, K. J., & Murray, R. M. (2021). *Feedback systems: An introduction for scientists and engineers*. Princeton, NJ: Princeton University Press.
17. Yamagata, T., & Santos-Rodriguez, R. (2024). Safe and robust reinforcement learning: Principles and practice. arXiv preprint arXiv:240318539.
18. Rodrigues, F., & Azevedo, C. L. (2019). Towards robust deep reinforcement learning for traffic signal control: Demand surges, incidents and sensor failures. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 3559–3566). IEEE.
19. Shi, T., Devailly, F. X., Larocque, D., & Charlin, L. (2024). Improving the generalizability and robustness of large-scale traffic signal control. *IEEE Open Journal of Intelligent Transportation Systems*, *5*, 2–15. https://doi.org/10.1109/OJITS.2023.3331689
20. Jiang, Q., Qin, M., Zhang, H., Zhang, X., & Sun, W. (2024). Blindlight: High robustness reinforcement learning method to solve partially blinded traffic signal control problem. *IEEE Transactions on Intelligent Transportation Systems*, *25*(11), 16625–16641. https://doi.org/10.1109/TITS.2024.3416154
21. Tan, K. L., Sharma, A., & Sarkar, S. (2020). Robust deep reinforcement learning for traffic signal control. *Journal of Big Data Analytics in Transportation*, *2*(3), 263–274. https://doi.org/10.1007/s42421-020-00029-6
22. Wu, C., Ma, Z., & Kim, I. (2020). Multi-agent reinforcement learning for traffic signal control: Algorithms and robustness analysis. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1–7). IEEE.
23. Zhang, R., Ishikawa, A., Wang, W., Striner, B., & Tonguz, O. K. (2021). Using reinforcement learning with partial vehicle detection for intelligent traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *22*(1), 404–415. https://doi.org/10.1109/TITS.2019.2958859
24. Xu, D., Li, C., & Wang, D., et al (2022). Robustness analysis of discrete state-based reinforcement learning models in traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *24*(2), 1727–1738.
25. Zeinaly, Z., Sojoodi, M., & Bolouki, S. (2023). A resilient intelligent traffic signal control scheme for accident scenario at intersections via deep reinforcement learning. *Sustainability*, *15*(2), 1329. https://doi.org/10.3390/su15021329
26. Aslani, M., Seipel, S., Mesgari, M. S., & Wiering, M. (2018). Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown tehran. *Advanced Engineering Informatics*, *38*, 639–655. https://doi.org/10.1016/j.aei.2018.08.002
27. Jordan, S., Chandak, Y., & Cohen, D., et al (2020). Evaluating the performance of reinforcement learning algorithms. In *International Conference on Machine Learning, PMLR* (pp. 4962–4973.
28. Smith, D., Djahel, S., & Murphy, J. (2014). A sumo based evaluation of road incidents' impact on traffic congestion level in smart cities. In *39th Annual IEEE Conference on Local Computer Networks Workshops* (pp. 702–710). IEEE.
29. Li, T., Zhao, W., & Baumanis, C., et al (2022). A python extension in sumo for simulating traffic incidents and emergency service vehicles. In *Canadian Society of Civil Engineering Annual Conference* (pp. 527–539). Springer.
30. Kucharski, R., & Gentile, G. (2019). Simulation of rerouting phenomena in dynamic traffic assignment with the information comply model. *Transportation Research Part B: Methodological*, *126*, 414–441. https://doi.org/10.1016/j.trb.2018.12.001
31. Cao, D., Wu, J., Dong, X., Sun, H., Qu, X., & Yang, Z. (2021). Quantification of the impact of traffic incidents on speed reduction: A causal inference based approach. *Accident Analysis and Prevention*, *157*, 106163. https://doi.org/10.1016/j.aap.2021.106163
32. AASHTO. (2018). *A policy on geometric design of highways and streets* (7th edn ed.). Washington, D.C: AASHTO., commonly referred to as the AASHTO Green Book.
33. FHWA. (2008). Traffic incident management quick clearance laws: A national review of best practices - move over laws - fhwa office of operations. https://ops.fhwa.dot.gov/publications/fhwahop09005/move_over.htm. Accessed 1 Apr 2025.
34. Erdmann, J. (2015). Sumo's lane-changing model. In *Modeling Mobility with Open Data: 2nd SUMO Conference 2014* (pp. 105–123). Springer, Berlin, Germany. May 15–16, 2014.
35. Ault, J., & Sharon, G. (2021). Reinforcement learning benchmarks for traffic signal control. In *Proceedings of the 35th Conference on Neural Information Processing Systems. Datasets and Benchmarks Track*.
36. Chen, C., Wei, H., & Xu, N., et al (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence* (pp. 3414–3421).
37. Varschen, C., & Wagner, P. (2006). Mikroskopische modellierung der personenverkehrsnachfrage auf basis von zeitverwendungstagebüchern. Integrierte Mikro-Simulation von Raum-und Verkehrsentwicklung Theorie, Konzepte, Modelle. *Praxis*, *81*, 63–69.
38. Lobo, S. C., Neumeier, S., & Fernandez, E. M., et al (2020). Intas–the ingolstadt traffic scenario for sumo. arXiv preprint arXiv:201111995.
39. Ault, J., Hanna, J., & Sharon, G. (2019). Learning an interpretable traffic signal control policy. arXiv preprint arXiv:191211023.
40. Zheng, G., Xiong, Y., & Zang, X., et al (2019). Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1963–1972).
41. Ma, J., & Wu, F. (2020). Feudal multi-agent deep reinforcement learning for traffic signal control. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (pp. 816–824).
42. Chu, T., Wang, J., Codecà, L., & Li, Z. (2020). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 1086–1095. https://doi.org/10.1109/TITS.2019.2901791
43. Henderson, P., Islam, R., & Bachman, P., et al (2018). Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*.
44. Du, W., Ye, J., & Gu, J., et al (2023). Safelight: A reinforcement learning method toward collision-free traffic signal control. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 14801–14810).
45. Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, *36*, 177–195. https://doi.org/10.1016/j.trc.2013.08.014
46. Wei, H., Zheng, G., & Yao, H., et al (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* (pp. 2496–2505.
47. SourceForge. (2014). Re: [Sumo-user] fwd: Help with traci reroutes | simulation of urban mobility. URL 2025-04-21 https://sourceforge.net/p/sumo/mailman/message/32148991/, [Online; .
48. Pereira, F., Rodrigues, F., & Ben-Akiva, M. (2013). Text analysis in incident duration prediction. *Transportation Research Part C: Emerging Technologies*, *37*, 177–192. https://doi.org/10.1016/j.trc.2013.10.002

Publisher's Note