# CONDITIONAL EFFECTS OF CONNECTED-VEHICLE REQUESTS IN REINFORCEMENT-LEARNING TRAFFIC SIGNAL CONTROL

A PREPRINT

**Xu An**
Chair of Transportation Systems Engineering
Technical University of Munich
Munich, Germany
AnX.RTL2324@tum-asia.edu.sg

August 26, 2026

## ABSTRACT

Connected vehicles can provide upstream observations and priority cues, but a request mechanism is useful only if it improves decisions beyond queue-responsive adaptation. We test this distinction at one site-derived SUMO intersection with a request-aware deep Q-network (DQN) and a matched queue-only ablation. The variants share the architecture, base congestion objective, training schedule, and 10–20 s timing shield; the request-aware variant additionally incorporates request-state inputs and an auxiliary request-shaped reward term. Five training realizations per variant were evaluated against fixed-time, presence-actuated, and max-pressure control at three demand levels, seven connected-vehicle penetrations, and ten paired traffic realizations, yielding 2,730 one-hour runs. To represent both stochastic sources, we independently resample the training and traffic axes of each $5 \times 10$ crossed result matrix. Queue-only DQN reduced mean queue relative to each of the three conventional baselines in all 21 demand–penetration conditions, with all 21 crossed-bootstrap intervals per baseline (63 in total) excluding zero. The request-aware DQN had lower point estimates than the same baselines in all conditions, but one interval at nominal demand and 50% penetration included zero because training-realization dispersion was large. Relative to queue-only control, the 95% crossed intervals excluded zero in only eight of 21 conditions: three favored the complete request mechanism (Full DQN) and five favored Queue-only DQN. At 0% penetration, no request occurs online, so that contrast is a training-history diagnostic rather than an online request effect. Request counts show that policies approved 75–81% of hold requests and 27–37% of switch requests; these diagnostics describe policy behavior but do not identify the cause of performance differences. Across this sweep, queue-responsive adaptation is the reproducible gain; the incremental effect of request conditioning is demand- and training-sensitive and is not monotonic in penetration.

***Keywords*** adaptive traffic signal control · deep Q-network · connected vehicles · demand shift · multi-seed evaluation · SUMO

## 1 Introduction

Traffic signal control allocates a limited green-time budget among competing movements. Fixed-time plans provide predictable operation but cannot respond to demand outside their design profile. Traffic-responsive control instead adjusts timing from observed conditions, through coordinated systems such as SCOOT, multivariable regulation, simulation-based optimization, and queue-pressure policies [1–4]. Reinforcement learning (RL) extends this idea by learning a sequential phase policy from traffic outcomes [5–7]. Modern traffic-signal RL can incorporate richer states, rewards, and coordination structures [8–11], making the controller's information set a central design choice.

Connected-vehicle position and speed offer an upstream view that conventional stop-line queues do not contain. Such observations support signal control, eco-driving, and cooperative mobility applications [12–14]. Their availability, however, does not establish that a particular request-conditioned controller will improve traffic outcomes. A vehicle-level cue may favor holding the current phase for an approaching vehicle or switching to serve another movement, while the aggregate queue objective may favor the opposite decision. When request-state inputs and request-shaped reward are added together, improvement over fixed time cannot distinguish queue-responsive adaptation from the effect of that complete request mechanism.

This distinction becomes harder to evaluate when learned policies are represented by a single training seed. Deep RL outcomes can vary across independent optimization runs, and point estimates from a few policies can produce unstable comparisons [15, 16]. A penetration-specific request benefit observed for one checkpoint may therefore reflect a persistent mechanism, a traffic condition, or one optimization realization. Separating these explanations requires matched ablation, independent training replication, and traffic-seed pairing under the same executable signal constraints.

We perform that controlled evaluation in one isolated, site-derived SUMO intersection. A request-aware DQN receives queue and phase variables together with a heuristic hold-or-switch request derived from connected-vehicle position and speed. A queue-only DQN retains the architecture, optimization schedule, queue-responsive objective, and 10–20 s timing shield while neutralizing the request state and reward. Five policies per variant are trained at 20% connected-vehicle penetration and the nominal demand, then tested across three demand levels and seven penetrations. Fixed-time, presence-actuated, and max-pressure controllers provide non-learning references under the same simulated traffic conditions.

The experiments establish two different findings. Queue-responsive learning transfers consistently: the queue-only family beats fixed, actuated, and max-pressure control in every demand–penetration cell under crossed uncertainty. The complete request mechanism is conditional: only eight of 21 crossed-bootstrap intervals exclude zero, the direction changes with demand, and the largest nominal-demand point loss carries wide training-realization uncertainty. Request counts show how policies respond to hold and switch cues without turning those counts into a causal explanation. The contribution is an empirical separation of adaptive-control gain from the conditional effect of request conditioning, supported by adaptive baselines, independent training replication, cross-demand evaluation, crossed uncertainty, and direct request-use diagnostics.

## 2 Related Work

### 2.1 Adaptive and reinforcement-learning signal control

Traffic-responsive control spans incremental timing adjustment, regulator-based optimization, simulation-driven search, and queue-pressure policies [1–3, 17]. Max-pressure control connects phase choice to queue imbalance and derives its throughput rationale from constrained queue scheduling [4, 18, 19]. Presence-actuated control instead extends or terminates green from detected approach demand. These methods provide more informative references than fixed timing because they already react to traffic state.

RL replaces a prescribed control law with a policy learned through interaction. Early isolated-intersection studies established the formulation [5], and deep value approximation enabled larger state spaces [6, 7]. Later methods learned phase competition, pressure-inspired objectives, and network-level cooperation [8–11]. Recent work has also tested connected-vehicle states in deep signal policies and examined robustness to demand and network shifts [20, 21]. Surveys and comparative studies show that reported performance depends on state, reward, phase logic, demand, and evaluation protocol as much as on the neural architecture [22–25]. Queue, delay, stopped time, switching cost, and priority satisfaction encode different operating preferences, so adding a reward term can alter the policy rather than simply enrich it [26].

Optimization variability is particularly relevant when the object of study is an augmentation to a learned controller. Independent seeds can change the learned policy and even the apparent ordering between methods [15]. Interval estimates and multi-run summaries are therefore needed when a small number of training realizations represent an RL method [16]. We apply this principle to traffic-signal control by reporting between-policy dispersion and the direction of each request-aware versus queue-only contrast across five independently trained policies.

### 2.2 Connected-vehicle information and request-aware control

Connected-vehicle information extends the controller's view upstream of conventional stop-line queues. Signal and eco-driving studies use position, speed, and signal state to anticipate arrivals or coordinate vehicle motion [12, 13]. Recent examples range from DQN signal policies using connected-vehicle traffic representations [20] to cooperative-perception control that reconstructs traffic state at low penetration [27]. Multi-day trajectory aggregation can also support signal retiming at low penetration without using per-vehicle online requests [28], while connected transit-priority work shows a distinct setting in which vehicle-level information is tied to a person-delay objective [29]. These designs use vehicle data in different ways and therefore do not imply a universal penetration–benefit curve. Cooperative mobility systems also add communication semantics, localization, latency, and reliability requirements beyond the control algorithm [14]. We isolate an idealized algorithmic request mechanism rather than model those layers.

Penetration changes how often requests occur, but it can also change the simulated vehicle mix when connected and human-driven classes use different car-following parameters [30]. Cross-penetration trends therefore combine information availability with traffic-behavior changes. Our inference is based on paired controller comparisons within each common demand, penetration, and traffic seed. The matched queue-only controller then isolates the complete request mechanism, including both the request state and request-shaped reward.

### 2.3 Evaluation gap addressed in this study

Prior comparisons often ask whether a request-aware learned controller outperforms a fixed program. That contrast combines at least two effects: adaptive queue-responsive timing and the marginal influence of connected-vehicle requests. We separate them through four controls. All adaptive controllers obey the same executable green bounds, all outcomes use a controller-independent measurement clock, learned variants differ only in the complete request mechanism, and every condition is repeated across training and traffic seeds. This design asks not only whether learned adaptation works, but whether requests improve an adaptive policy consistently enough to support a general control claim.

## 3 Method

### 3.1 Simulation scenario and intersection layout

The experimental environment is modeled after a real-world, site-derived four-approach intersection from the Munich Mobility Innovation Center (MIC) test facility in Germany (Fig. 1a). Microscopic vehicle kinematics, lane assignments, detector trigger activations, and traffic signal state actuations are simulated in Eclipse SUMO (v1.20+) coupled with PyTorch via TraCI at a step resolution of $\Delta t = 0.1$ s [31, 32]. The geometric layout comprises four approaches intersecting at a central junction with dedicated right-turn bypasses and stop-line induction loop detectors ($q_1 \dots q_4$) placed on each incoming approach (Fig. 1b). A simulated Roadside Unit (RSU) defines an idealized 150 m vehicle-to-infrastructure (V2I) request eligibility zone upstream of the stop line.

The nominal traffic demand corresponds to 1,000 vehicles per hour, where the dominant approach contributes 400 veh/h and the remaining three approaches contribute 200 veh/h each. In all approaches, 80% of vehicles travel straight through the intersection and 20% make right turns; no left turns are permitted by the geometric layout. To test generalization under varying congestion, we scale route volumes uniformly by factors of 0.7, 1.0, and 1.3, yielding three stationary demand levels: 700 veh/h (low demand), 1,000 veh/h (nominal design condition), and 1,300 veh/h (elevated high-demand condition). Across each demand level, the connected-vehicle market penetration rate $p \in \{0\%, 5\%, 10\%, 20\%, 50\%, 80\%, 100\%\}$ is systematically varied. Human-driven vehicles follow the Krauss car-following model with maximum acceleration 2.6 m/s$^2$, driver imperfection $\sigma = 0.5$, and minimum standstill gap 2.5 m, whereas connected vehicles operate with 2.8 m/s$^2$, $\sigma = 0.2$, and a 2.0 m gap. Both vehicle classes share a maximum speed of 13.9 m/s (50 km/h).

### 3.2 Signal program and baseline fixed timing

The physical signal program reproduces the native Munich testbed fixed-time plan (SP 1–VA) defined in `final.net.xml` with a total cycle length of 59 s (Table 1). The sequence contains nine distinct phases serving two conflicting pairs of opposing approaches: Signal Groups $SG_1$ and $SG_3$ govern Main Green A (Phase 2, 20 s fixed duration), while Signal Groups $SG_2$ and $SG_4$ govern Main Green B (Phase 6, 20 s fixed duration). Intervening clearance intervals include 1 s red-yellow preparation, 3 s yellow transitions, 7 s all-red pedestrian clearance, and 3 s secondary all-red clearance, summing exactly to $1 + 1 + 20 + 3 + 7 + 1 + 20 + 3 + 3 = 59$ s.

Under the fixed-time baseline, the controller cyclically executes the 59 s schedule without adaptation. For all adaptive and reinforcement-learning controllers (Full DQN, Queue-only DQN, Presence-Actuated, and Max-Pressure), the clearance intervals (Phases 0, 1, 3, 4, 5, 7, 8) remain fixed and mandatory to ensure clearance safety, while decisions occur at 1.0 s intervals exclusively during active main greens (Phases 2 and 6). An action shield enforces $G_{\min} = 10$ s and $G_{\max} = 20$ s, bounding the cycle length between 39 s and 59 s.

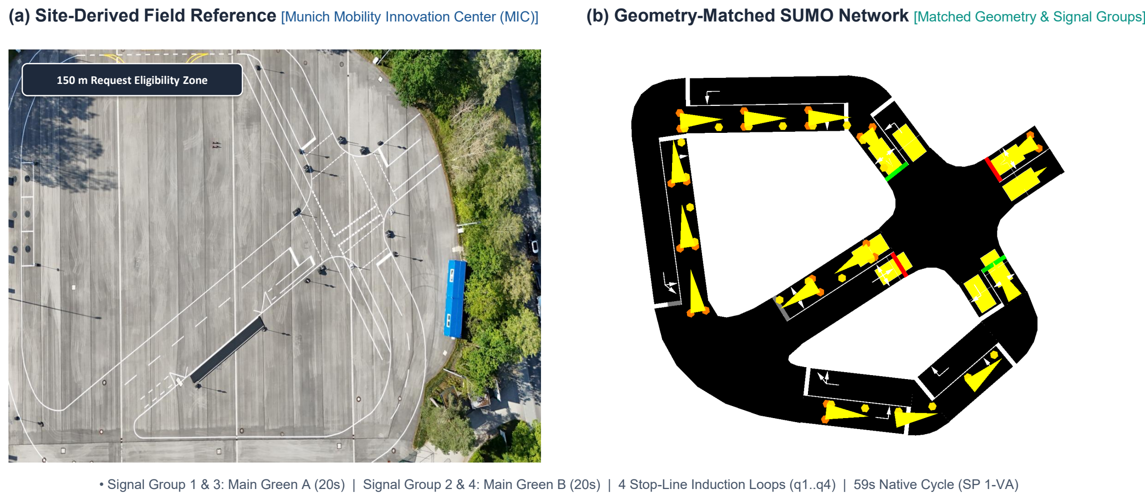


Figure 1: Intersection layout and simulation environment. (a) Aerial view of the site-derived Munich Mobility Innovation Center (MIC) field reference (image courtesy of MIC/TUM) showing approach geometry, stop-line markers, and the 150 m request eligibility zone. (b) Site-derived microscopic simulation network in Eclipse SUMO with matched geometric alignment, four stop-line loop detectors ($q_1 \dots q_4$), and signal groups ($SG_1/SG_3$ for Main Green A, $SG_2/SG_4$ for Main Green B) operating under the native 59 s cycle program.

Table 1: Signal program phase sequence and baseline fixed timing (SP 1–VA, 59 s total cycle length) implemented in `final.net.xml`. Clearance phases are strictly locked across all controllers, while adaptive controllers adjust Main Green A and B durations within $[G_{\min}, G_{\max}] = [10, 20]$ s.

| Phase | Signal State | Description / Active Movements | Fixed Duration | Adaptive Bounds |
|---|---|---|---|---|
| 0 | `rrrrrrrrrrrr` | All-red initial clearance / prep | 1 s | Fixed (1 s) |
| 1 | `rrryyyrrryyy` | Red-yellow transition (Phase A) | 1 s | Fixed (1 s) |
| 2 | `rrrGGgrrrGGg` | **Main Green A** ($SG_1, SG_3$: Opposing) | 20 s | $10 \le G_A \le 20$ s |
| 3 | `rrryyyrrryyy` | Yellow clearance A | 3 s | Fixed (3 s) |
| 4 | `rrrrrrrrrrrr` | All-red / Pedestrian clearance | 7 s | Fixed (7 s) |
| 5 | `yyyrrryyyrrr` | Red-yellow transition (Phase B) | 1 s | Fixed (1 s) |
| 6 | `GGgrrrGGgrrr` | **Main Green B** ($SG_2, SG_4$: Opposing) | 20 s | $10 \le G_B \le 20$ s |
| 7 | `yyyrrryyyrrr` | Yellow clearance B | 3 s | Fixed (3 s) |
| 8 | `rrrrrrrrrrrr` | All-red secondary clearance | 3 s | Fixed (3 s) |
| **Total Cycle Duration** | | | **59 s** | $39 \le C \le 59$ s |

### 3.3 State and action

At decision time $t$, the controller observes a ten-dimensional state

$$\mathbf{s}_t = [I_t^A, I_t^B, \tilde{g}_t, \tilde{q}_{1,t}, \dots, \tilde{q}_{4,t}, \tilde{c}_t, \tilde{\tau}_t, \tilde{n}_t]. \tag{1}$$

$I_t^A$ and $I_t^B$ indicate the active main green. If the signal is in a clearance phase, neither indicator is active and the controller does not issue a decision. The elapsed green $g_t$ is divided by 20 s and clipped to one. Each lane queue is the SUMO last-step halting count, divided by 20 vehicles and clipped to one.

The final three elements describe the request heuristic. The request code $c_t \in \{-1, 0, +1\}$ represents switch, no request, and hold, then maps to $[0, 1]$. The minimum estimated time to arrival $\tau_t$ is divided by 20 s and clipped to one. The candidate count $n_t$ is divided by four because at most one candidate per incoming lane is retained. In the queue-only ablation, these three inputs are fixed at their neutral values. The network dimension remains ten so that the architecture and optimization schedule are unchanged.

The action $a_t \in \{0, 1\}$ requests a hold or switch from the execution layer. A switch before 10 s of green is ignored, whereas a transition is forced at 20 s even when the policy requests a hold. Intermediate yellow and all-red phases

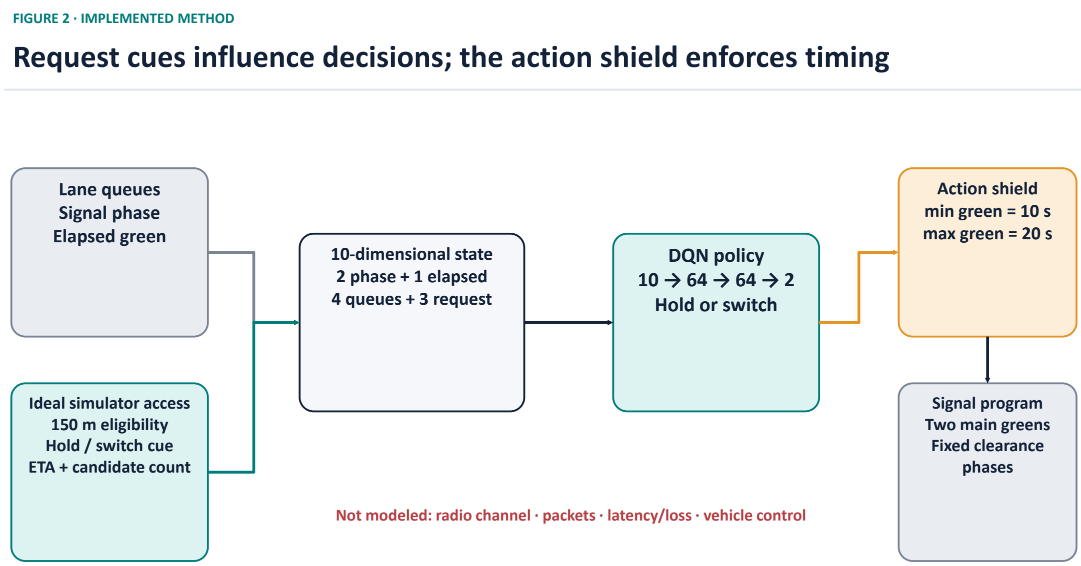


Figure 2: Implemented closed-loop control framework. Lane queues and signal timing are available to both learned controllers. The request-aware variant additionally receives a heuristic request code, minimum estimated arrival time, and candidate count, and its reward includes request satisfaction. The action shield, not the DQN, enforces minimum and maximum green. No physical communication channel or vehicle-control output is modeled.

follow the programmed sequence and cannot be skipped. The shield therefore makes the 10–20 s green-time range executable. Elapsed-time normalization alone would not prevent an indefinite hold.

### 3.4 Request heuristic

The request mechanism uses connected vehicles located on the four incoming lanes within 150 m of the stop line. It retains the nearest eligible vehicle on each lane, calculates

$$\tau_i = \frac{d_i}{\max(v_i, 2\ \mathrm{m/s})}, \tag{2}$$

and selects the candidate with minimum $\tau_i$. A request is generated only during a main green. If the candidate's movement is currently served, a hold request is created when $\tau_i \leq 6$ s but exceeds the remaining programmed time. If the movement is not served, a switch request is created when $\tau_i \leq 6$ s and the minimum green has elapsed. Once a selected vehicle produces a nonzero request, it is excluded from further requests until it leaves the simulation.

Distance and speed are read directly from SUMO, and the 150 m range acts as a deterministic eligibility threshold. The request therefore represents ideal information availability rather than a propagation, sensing, or packet-delivery model. Its only output is a cue to the signal policy. Vehicles receive no speed advisory or control command.

### 3.5 Reward and DQN training

The transition reward is

$$r_t = -\left(Q_t + \lambda_s S_t + \lambda_w \Delta W_t\right) + \lambda_r H_t, \tag{3}$$

where $Q_t$ is total queue over four incoming lanes, $S_t$ indicates an executed phase switch, and $\Delta W_t$ is the stopped time accumulated by all vehicles since the previous decision. A vehicle is stopped when its speed is at most 0.1 m/s. The weights are $\lambda_s = 2$, $\lambda_w = 0.5$, and $\lambda_r = 1$ for the request-aware controller. For a nonzero request, $H_t = +1$ when the executed action satisfies the request and $H_t = -1$ otherwise. When there is no request, $H_t = 0$. The queue-only ablation sets $\lambda_r = 0$ in addition to neutralizing the request inputs.

The online Q-network contains two fully connected hidden layers with 64 rectified linear units each and two outputs. Its target for a sampled transition is

$$y_t = r_t + (1 - d_t)\gamma \max_{a'} Q_{\text{target}}(\mathbf{s}_{t+1}, a'), \tag{4}$$

where $d_t$ indicates termination and $\gamma = 0.99$. Training minimizes the smooth $L_1$ loss between $Q_{\text{online}}(\mathbf{s}_t, a_t)$ and $y_t$ using Adam with learning rate $10^{-3}$. The replay buffer holds 50,000 transitions. Updates start after 2,000 transitions, use batches of 64 every four decision steps, clip the gradient norm at 10, and copy online weights to the target network every 1,000 steps. Epsilon decreases linearly from 1.0 to 0.05 over 30,000 decision steps.

We trained five request-aware and five queue-only policies at the nominal demand and 20% penetration. Training seeds were 42, 142, 242, 342, and 442; each policy used the following 20 integers as its SUMO episode seeds. Every run comprised 20 one-hour episodes. The episode-20 checkpoint was selected before evaluation without reference to test performance, and all evaluation actions were greedy.

Table 2: Simulation and learning settings. The DQN variants differ only in the request inputs and request-reward term.

| Setting | Value |
|---|---|
| Scenario | One isolated, four-approach intersection |
| Demand levels and horizon | 700 / 1,000 / 1,300 veh/h; 3,600 s |
| SUMO step / decision interval | 0.1 s / 1.0 s |
| Minimum / maximum green | 10 s / 20 s |
| Training condition | 1,000 veh/h; 20% penetration |
| Training policies | 5 per DQN variant |
| Training seeds | 42, 142, 242, 342, 442 |
| Training episodes | 20 per policy |
| Evaluation penetrations | 0, 5, 10, 20, 50, 80, 100% |
| Evaluation seeds | 1000–1009, paired across controllers |
| Network | 10–64–64–2 |
| Replay / batch size | 50,000 / 64 |
| Discount / learning rate | 0.99 / $10^{-3}$ |
| Checkpoint | Pre-specified final checkpoint, episode 20 |

## 4 Experimental Design

### 4.1 Controllers and factorial evaluation

The request-aware and queue-only DQNs follow Section 3. Three non-learning controllers provide operational references. Fixed time executes the native program. The presence-actuated controller checks a 40 m stop-line zone on the served approaches each second, extends green while a vehicle is present, and terminates at the first empty check after 10 s. The isolated-intersection max-pressure controller compares the halting queue on the served approach pair with the opposing pair and switches after 10 s when the opposing pressure is larger [4, 18]. Because the isolated test intersection has no modeled downstream spillback constraint, pressure is calculated directly from competing incoming halting queues; this should be interpreted as an isolated-intersection queue-pressure implementation rather than a network-level formulation. Both adaptive baselines are forced to switch at 20 s and use the same clearance sequence as the DQNs.

The evaluation therefore contains 13 controller policies: five request-aware DQNs, five queue-only DQNs, fixed time, presence-actuated control, and max-pressure control. Every policy was run at three demand levels, seven penetrations, and ten traffic seeds from 1000 to 1009. Pairing aligns stochastic vehicle generation within each demand–penetration cell. The formal package contains $13 \times 3 \times 7 \times 10 = 2{,}730$ one-hour evaluation runs, in addition to 200 one-hour training episodes. Every planned run and checkpoint was retained.

### 4.2 Outcomes and mechanism diagnostics

The primary outcome is average total queue, denoted $\overline{Q}$. At every whole second, including green, yellow, and all-red intervals, the four incoming-lane halting counts are summed. The episode outcome is the mean of those one-second samples. The measurement clock is independent of the DQN decision clock and is identical for all controllers.

The secondary congestion outcomes are stopped time per observed vehicle and travel time per completed vehicle. At every 0.1 s step, stopped time increases for vehicles travelling at no more than 0.1 m/s. Throughput is the number of completed vehicles, and phase switches measure how often each controller changes the served movement. These outcomes test whether a queue result is accompanied by consistent traffic consequences.

Request diagnostics are recorded for the learned policies at every decision. They include request frequency, hold and switch shares, approval by request type, hold requests overridden by maximum green, forced switches, and total phase switches. The diagnostics describe how the learned policy responds to requests; they do not by themselves identify a causal explanation for performance.

The primary contrasts compare request-aware DQN with fixed, actuated, max-pressure, and queue-only control. The queue-only contrast identifies the marginal effect of the complete request mechanism. The other contrasts combine queue-responsive learning with that mechanism. Queue-only versus fixed time provides an additional check that adaptation survives when the request mechanism is removed.

### 4.3 Statistical analysis

Each DQN condition forms a crossed matrix with five training realizations as rows and ten traffic realizations as columns. The 50 evaluations are therefore not treated as 50 independent replicates. For every DQN comparison on the primary outcome, we compute the mean over the complete matrix and obtain a two-sided 95% percentile interval by independently resampling five rows and ten columns with replacement. We use 50,000 bootstrap replicates and a fixed analysis seed. For Full-versus-Queue-only contrasts, the identical resampled training-seed indices and traffic-seed indices were applied jointly to both variants before differencing, preserving the paired structure along both axes. For DQN–baseline contrasts, the baseline result for each traffic realization is shared across the five DQN rows.

The crossed bootstrap is the main uncertainty analysis for average queue because it represents both training-realization and traffic-realization variation. We describe an interval as excluding zero rather than treating its 50 underlying cells as independent evidence. The earlier traffic-seed analysis—which first averages over training policies, then applies paired $t$ and Wilcoxon tests with Holm correction across seven penetrations—is retained in the evidence package as a traffic-conditional diagnostic. A traffic-clustered interval over all 50 cells is also retained for traceability, but it is not interpreted as a complete two-source interval. We separately report the standard deviation and range of the five policy means. The checkpoint rule, seed pairing, outcomes, aggregation, and multiplicity procedure were fixed before statistical processing. No run or training realization was removed as an outlier.

## 5 Results

### 5.1 Queue-responsive adaptation across demand and penetration

Both learned controller families had lower mean queue than fixed time in every demand–penetration cell. The request-aware reductions were 13.25–24.01%, and the queue-only reductions were 13.43–25.57%. More importantly for the method-level claim, every queue-only versus fixed, actuated, and max-pressure crossed-bootstrap interval excluded zero in the favorable direction ($21/21$ per baseline, 63 in total). Queue-only control also showed 5/5 training-realization direction agreement in all 21 fixed-time comparisons. Queue-responsive adaptation therefore remained beneficial without the request mechanism.

The request-aware DQN also had lower point estimates than presence-actuated and max-pressure control in all 21 cells (Table 3). Twenty of 21 crossed intervals excluded zero for each conventional-baseline comparison. The exception was nominal demand at 50% penetration: request-aware mean queue remained below fixed, actuated, and max-pressure control, but all three intervals crossed zero because between-training variation was large. We therefore report the favorable mean without treating this cell as established across training realizations.

Table 3: Range of average-queue percentage reductions across the seven penetrations at each demand level. Crossed-bootstrap intervals exclude zero for every Queue-only comparison ($21/21$ per baseline, $63/63$ total) and for $20/21$ Request-aware comparisons per baseline; the common exception is nominal demand at 50% penetration.

| Demand factor | Full–Fixed (%) | Full–Act. (%) | Full–MP (%) | Queue–Fixed (%) | Queue–Act. (%) | Queue–MP (%) |
|---|---|---|---|---|---|---|
| 0.7 | 15.23–22.59 | 17.58–25.61 | 16.66–25.95 | 13.43–22.39 | 16.64–23.58 | 16.68–22.54 |
| 1.0 | 13.62–19.08 | 11.66–33.08 | 17.60–20.50 | 18.47–25.57 | 19.45–36.90 | 20.26–29.00 |
| 1.3 | 13.25–24.01 | 12.93–23.48 | 13.78–24.61 | 13.76–18.65 | 13.73–18.99 | 14.15–19.29 |

### 5.2 Request conditioning at the training demand

The matched ablation changes the interpretation of the baseline result. At nominal demand, Full-minus-Queue-only point estimates were positive at every penetration, but crossed uncertainty supports a clear increase only at 20 and 80% penetration (Fig. 3). The respective differences were $+0.141$ vehicles (95% CI $[+0.070, +0.217]$) and $+0.169$ ($[+0.079, +0.265]$). The largest point difference occurred at 50% penetration ($+0.351$ vehicles), yet its interval was $[-0.103, +1.028]$ because training-realization dispersion dominated traffic-realization dispersion. It is therefore inappropriate to present the 50% cell as a stable degradation of the learning method.

No request occurs online at 0% penetration. The positive full-minus-queue difference in that cell therefore reflects policies learned under different training inputs and objectives rather than the use of requests during evaluation. More generally, the ablation measures the complete request mechanism, not the isolated value of one observation coordinate or reward term.

Extending the 2D crossed bootstrap to secondary delay metrics confirms identical behavior: for both stopped time and completed-vehicle travel time, the 95% crossed-bootstrap intervals exclude zero in the exact same eight of 21 conditions (and in matching directions: three favoring Full DQN, five favoring Queue-only DQN), while completed-vehicle counts differ by only 0.05–0.15%.

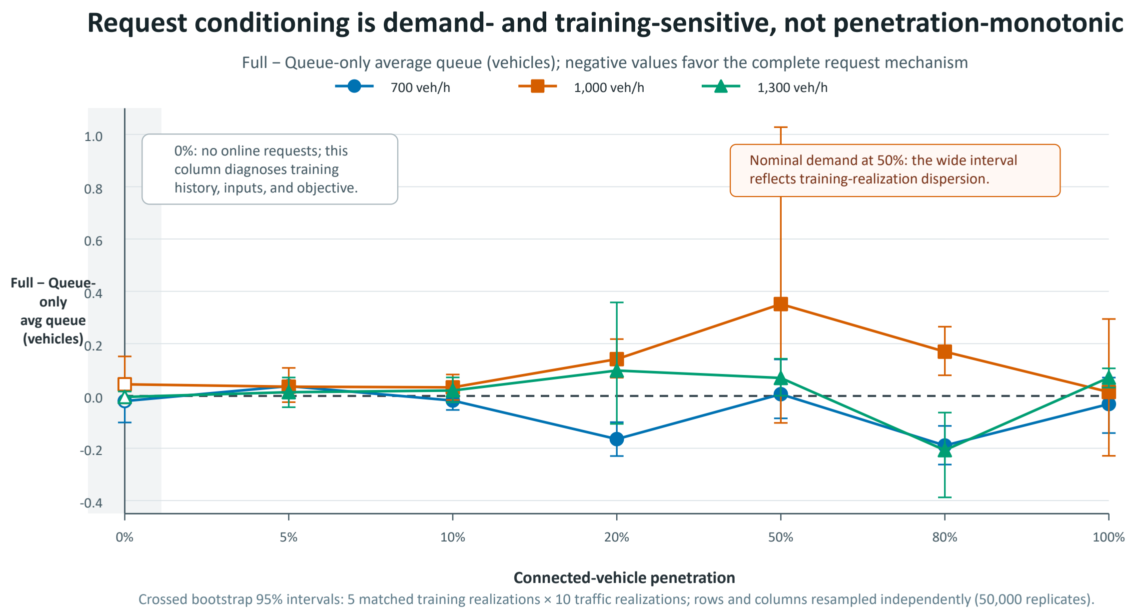


Figure 3: Effect of the complete request mechanism on average queue. Points are Full minus Queue-only means; negative values favor Full. Error bars are 95% crossed-bootstrap intervals obtained by independently resampling five matched training realizations and ten traffic realizations. Hollow markers at 0% penetration denote a training-history diagnostic because no request occurs online. The response changes with demand and is not monotonic in penetration.

Table 4: Complete request-mechanism effect on average queue, reported as Full minus Queue-only mean [95% crossed-bootstrap interval] in vehicles. Negative values favor Full.

| Pen. (%) | Demand 0.7 | Demand 1.0 | Demand 1.3 |
|---|---|---|---|
| 0 | $-0.020$ $[-0.101, +0.059]$ | $+0.044$ $[-0.028, +0.151]$ | $-0.004$ $[-0.023, +0.017]$ |
| 5 | $+0.038$ $[+0.016, +0.058]$ | $+0.035$ $[-0.024, +0.107]$ | $+0.014$ $[-0.043, +0.071]$ |
| 10 | $-0.018$ $[-0.054, +0.022]$ | $+0.033$ $[-0.015, +0.082]$ | $+0.021$ $[-0.024, +0.071]$ |
| 20 | $-0.165$ $[-0.230, -0.100]$ | $+0.141$ $[+0.070, +0.217]$ | $+0.097$ $[-0.107, +0.358]$ |
| 50 | $+0.007$ $[-0.086, +0.140]$ | $+0.351$ $[-0.103, +1.028]$ | $+0.069$ $[+0.001, +0.142]$ |
| 80 | $-0.189$ $[-0.262, -0.115]$ | $+0.169$ $[+0.079, +0.265]$ | $-0.207$ $[-0.388, -0.064]$ |
| 100 | $-0.031$ $[-0.142, +0.070]$ | $+0.015$ $[-0.229, +0.294]$ | $+0.070$ $[+0.032, +0.106]$ |

### 5.3 Demand dependence of the request effect

Changing demand altered both the direction and magnitude of the request contrast. At 700 veh/h, crossed intervals favored Full at 20 and 80% penetration, favored Queue-only at 5%, and included zero in the other four cells. At 1,000 veh/h, the intervals favored Queue-only at 20 and 80% and included zero elsewhere. At 1,300 veh/h, 80% favored Full, whereas 50 and 100% favored Queue-only; the remaining cells included zero. The complete request mechanism is therefore favored in three of 21 conditions, whereas Queue-only is favored in five; the other 13 remain unresolved under crossed uncertainty.

The sign pattern is not ordered by penetration. For example, 80% favors Queue-only at nominal demand but Full at both shifted demands. Likewise, high-demand results change from Queue-only at 50% to Full at 80% and back to Queue-only at 100%. These reversals are evidence of a demand-conditioned response, not a penetration dose–response relation.

### 5.4 Independent training exposes policy variability

Most request-aware policies formed a compact group at nominal demand, with between-policy standard deviations of 0.026–0.041 vehicles from 0 to 80% penetration except at 50% (Table 5). At 50%, policy means ranged from 1.837 to 3.918 vehicles. The 2.081-vehicle range was more than sixteen times the mean within-policy traffic-realization standard deviation of 0.127. The same training realization had the largest mean at 100%, where the five-policy range was 0.896 vehicles. Its 50 evaluation runs were complete and are retained. We describe this observation only as poor out-of-sample performance for one training realization; the available evidence does not diagnose why it occurred.

Queue-only policies also varied across training seeds, but their largest nominal-demand range was 0.288 vehicles. The contrast between the two DQN families was correspondingly less seed-consistent than their contrasts with conventional control: full versus queue-only direction agreement ranged from 2/5 to 5/5 across cells. The penetration-specific sign pattern from a single policy is therefore not a stable property of the request mechanism.

Table 5: Between-policy variation in nominal-demand average queue across five training realizations. Values are standard deviation and range of the five policy means.

| Pen. | Request-aware | | Queue-only | |
|---|---|---|---|---|
| (%) | SD | Range | SD | Range |
| 0 | 0.033 | 0.087 | 0.108 | 0.253 |
| 5 | 0.032 | 0.086 | 0.076 | 0.175 |
| 10 | 0.032 | 0.083 | 0.046 | 0.119 |
| 20 | 0.026 | 0.052 | 0.082 | 0.171 |
| 50 | 0.812 | 2.081 | 0.116 | 0.246 |
| 80 | 0.041 | 0.094 | 0.079 | 0.174 |
| 100 | 0.348 | 0.896 | 0.125 | 0.288 |

### 5.5 How the learned policy uses requests

Request density at the nominal demand increased from zero to 595 observations per 1,000 vehicles as penetration rose (Table 6). Hold requests accounted for 57–75% of the total. The policies approved 75–81% of holds but only 27–37% of switches. A request cue therefore did not directly determine the action; the learned policy rejected most requests that asked it to leave the current phase.

Hold requests overridden by the 20 s shield accounted for at most 0.22% of holds. The DQNs executed approximately 164–174 phase switches per hour, compared with 122 for fixed time. These event counts document frequent adaptation and asymmetric approval, but they do not establish whether an accepted or rejected request caused any queue outcome.

Table 6: Request diagnostics for the request-aware DQN at nominal demand. Request rate is per 1,000 observed vehicles; approvals and forced overrides are percentages.

| Pen. (%) | Req. rate | Hold share (%) | Hold appr. (%) | Switch appr. (%) | Forced hold override (%) | Switches (/h) |
|---|---|---|---|---|---|---|
| 0 | 0.0 | – | – | – | – | 174.36 |
| 5 | 26.9 | 62.3 | 78.3 | 30.6 | 0.00 | 173.62 |
| 10 | 56.9 | 65.7 | 81.2 | 28.3 | 0.05 | 172.54 |
| 20 | 103.5 | 61.3 | 75.0 | 26.9 | 0.22 | 170.32 |
| 50 | 324.0 | 57.4 | 76.4 | 31.9 | 0.02 | 163.72 |
| 80 | 458.5 | 69.5 | 79.8 | 30.2 | 0.05 | 164.48 |
| 100 | 595.1 | 74.7 | 76.8 | 37.3 | 0.01 | 164.84 |

# 6 Discussion

The experiments separate two claims that are easily conflated. First, queue-responsive learning produced a consistent advantage over fixed-time, actuated, and max-pressure control in the tested intersection. Second, adding request-state inputs and request-shaped reward did not produce a consistent marginal gain over the matched queue-responsive controller. The first result concerns adaptive timing. The second concerns the complete request mechanism, not a pure observation contribution. Treating the conventional-baseline comparison as evidence for that mechanism would overstate what the experiment establishes.

A request cue and its reward term can conflict with the aggregate congestion objective. A hold request concerns one approaching vehicle, whereas queue and stopped-time penalties aggregate conditions across movements. A hold may preserve service for that arrival while delaying an opposing queue; a switch may incur clearance time and truncate useful green. The learned controller approved roughly three quarters of hold requests and about one third of switch requests. These counts are compatible with selective request use, but they cannot determine which decisions caused the observed performance differences.

The crossed analysis changes the strength of the demand-dependent conclusion. Only eight of 21 intervals exclude zero, and the largest nominal-demand point loss at 50% has the widest interval. The supported pattern is therefore sparse rather than uniformly adverse: Full is favored at low-demand 20 and 80% and high-demand 80%, while Queue-only is favored in five other cells. Neither penetration nor matching the training penetration predicts the direction reliably.

Independent retraining changes the interpretation of a single-checkpoint pattern. Most request-aware policies were closely grouped at low and medium penetrations, but one retained training realization performed substantially worse at 50 and 100%. Omitting it would suppress a genuine source of method-level uncertainty. Training realizations are therefore part of the evidence for deep RL, not merely a device for narrowing error bars [15, 16]. A penetration trend inferred from one checkpoint can describe one learned policy rather than the training procedure.

The practical design implication is to learn when a request deserves influence rather than apply request augmentation unconditionally. Hold and switch requests could be represented separately and conditioned on queue imbalance, phase age, and congestion state. A factorial experiment that independently removes request observations and request reward would separate the contribution of the state inputs from reward shaping. It would also help characterize high-penetration training-realization variability without assuming its cause in advance.

The strong baseline results should also be read at the scale of the study. Max-pressure was adapted to the same isolated two-phase junction and safety timings, and the actuated controller used a common detector-based logic. The DQN advantage is therefore meaningful for this controlled site, but it is not evidence that one learned controller dominates all actuated or max-pressure implementations. Likewise, exact simulator access isolates an algorithmic mechanism; it does not measure the reliability of a deployed V2X system with delayed, missing, or erroneous messages.

# 7 Limitations

The study uses one isolated, site-derived intersection, one turning pattern, and three scaled demand levels. It does not establish transfer to coordinated corridors, alternative geometries, pedestrian operation, incident conditions, or sustained oversaturation. The site has not been calibrated against field queues or travel times.

Five training realizations expose material variability but do not fully characterize convergence or hyperparameter sensitivity. Training was limited to 20 one-hour episodes under a fixed configuration, and the poor high-penetration realization was not diagnosed through representation or gradient-level analysis. The whole-mechanism ablation also removes request observations and request reward together, so it cannot isolate their individual effects. Crossed-bootstrap intervals are pointwise exploratory intervals across 21 conditions rather than simultaneous familywise intervals, and the training axis contains only five realizations.

Penetration changes both information availability and simulated vehicle composition because connected and human-driven vehicles use different car-following parameters. Paired controller contrasts within a cell remain controlled, but comparisons across penetration levels are not a pure information dose–response experiment. In addition, the request diagnostics are observational and cannot show which accepted or rejected requests caused an outcome.

The request mechanism assumes exact simulator state inside a deterministic 150 m range. Communication delay, loss, localization error, security, controller hardware, and fail-safe behavior are outside the model. The max-pressure baseline is an isolated-intersection adaptation rather than a network implementation. Future work should therefore combine a factorial state–reward ablation with independent variation of information and vehicle behavior, multiple networks and training budgets, explicit communication impairments, and hardware-in-the-loop or field validation.

## 8 Conclusion

Across three demands and seven connected-vehicle penetrations, the queue-only DQN reduced mean queue relative to fixed-time, actuated, and max-pressure baselines in every tested condition, with crossed intervals excluding zero throughout. The matched five-realization ablation supports a narrower conclusion for request conditioning: three conditions favored the complete request mechanism, five favored its removal, and the remaining 13 were unresolved under crossed uncertainty. The direction was not monotonic in penetration and changed with demand. One retained high-penetration training realization also widened uncertainty enough to overturn an apparently strong point comparison. Request conditioning should therefore be evaluated as a context-dependent mechanism, with both training and traffic variation represented explicitly.

## Data and Code Availability

The evaluation CSV files, analysis configuration, controller source code, SUMO network and route definitions, and execution manifest are retained in a versioned local research package and are available from the corresponding author on reasonable request. The arXiv package includes the manuscript source and both vector figures, but not trained model binaries or the Python environment.

## Ethics Statement

This study used traffic simulation and did not involve human participants, animals, personal data, or intervention in public traffic. Institutional ethics approval was therefore not required.

## Author Contributions

Xu An: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, and Writing – review and editing.

## Conflict of Interest

The author declares no conflict of interest.

## Funding

This research received no external funding.

## Declaration of AI Assistance

Generative AI tools were used under the author's supervision to assist with manuscript restructuring, language editing, code inspection, statistical cross-checking, and figure preparation. The author reviewed the experimental design, source code, reported values, claims, citations, and final text and takes responsibility for the content.

## References


[1] Peter B. Hunt, Dennis I. Robertson, R. D. Bretherton, and R. I. Winton. SCOOT: A traffic responsive method of coordinating signals. Technical Report LR1014, Transport and Road Research Laboratory, 1981. URL https://www.trl.co.uk/publications/lr1014.

[2] Christina Diakaki, Markos Papageorgiou, and Kostas Aboudolas. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice*, 10(2):183–195, 2002. doi:10.1016/S0967-0661(01)00121-6.

[3] Aleksandar Stevanovic, Peter T. Martin, and Jelka Stevanovic. VisSim-based genetic algorithm optimization of signal timings. *Transportation Research Record: Journal of the Transportation Research Board*, 2035(1):59–68, 2007. doi:10.3141/2035-07.

[4] Pravin Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195, 2013. doi:10.1016/j.trc.2013.08.014.

[5] Baher Abdulhai, Rob Pringle, and Grigoris J. Karakoulas. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering*, 129(3):278–285, 2003. doi:10.1061/(ASCE)0733-947X(2003)129:3(278).

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi:10.1038/nature14236.

[7] Li Li, Yisheng Lv, and Fei-Yue Wang. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3):247–254, 2016. doi:10.1109/JAS.2016.7508798.

[8] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. IntelliLight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2496–2505, 2018. doi:10.1145/3219819.3220096.

[9] Guanjie Zheng, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li. Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1963–1972, 2019. doi:10.1145/3357384.3357900.

[10] Hua Wei, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash V. Gayah, Kai Xu, and Zhenhui Li. PressLight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1290–1298, 2019. doi:10.1145/3292500.3330949.

[11] Hua Wei, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li. CoLight: Learning network-level cooperation for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1913–1922, 2019. doi:10.1145/3357384.3357902.

[12] Li Li, Ding Wen, and Danya Yao. A survey of traffic control with vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):425–432, 2014. doi:10.1109/TITS.2013.2277737.

[13] Hesham Rakha and Raj Kishore Kamalanathsharma. Eco-driving at signalized intersections using V2I communication. In *2011 14th International IEEE Conference on Intelligent Transportation Systems*, pages 341–346, 2011. doi:10.1109/ITSC.2011.6083084.

[14] Laurens Hobert, Andreas Festag, Ignacio Llatser, Luciano Altomare, Filippo Visintainer, and Andras Kovacs. Enhancements of V2X communication in support of cooperative autonomous driving. *IEEE Communications Magazine*, 53(12):64–70, 2015. doi:10.1109/MCOM.2015.7355568.

[15] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 3207–3214, 2018. doi:10.1609/aaai.v32i1.11694.

[16] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

[17] Markos Papageorgiou, Christina Diakaki, Vasiliki Dinopoulou, Apostolos Kotsialos, and Yibing Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, 2003. doi:10.1109/JPROC.2003.819610.

[18] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12): 1936–1948, 1992. doi:10.1109/9.182479.

[19] Tung Le, Péter Kovács, Neil Walton, Hai L. Vu, Lachlan L. H. Andrew, and Serge S. P. Hoogendoorn. Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, 58:431–450, 2015. doi:10.1016/j.trc.2014.11.009.

[20] Yang Shi, Zhenbo Wang, Tim J. LaClair, Chieh Wang, Yunli Shao, and Jinghui Yuan. A novel deep reinforcement learning approach to traffic signal control with connected vehicles. *Applied Sciences*, 13(4):2750, 2023. doi:10.3390/app13042750.

[21] Tianyu Shi, François-Xavier Devailly, Denis Larocque, and Laurent Charlin. Improving the generalizability and robustness of large-scale traffic signal control. *IEEE Open Journal of Intelligent Transportation Systems*, 5:2–15, 2024. doi:10.1109/OJITS.2023.3331689.

[22] Patrick Mannion, Jim Duggan, and Enda Howley. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic Road Transport Support Systems*, pages 47–66. Springer, 2016. doi:10.1007/978-3-319-25808-9_4.

[23] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, 68(2):1243–1253, 2019. doi:10.1109/TVT.2018.2890726.

[24] Mohammad Noaeen, Atharva Naik, Liana Goodman, Jared Crebo, Taimoor Abrar, Zahra Shakeri Hossein Abad, Ana L. C. Bazzan, and Behrouz Far. Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, 199:116830, 2022. doi:10.1016/j.eswa.2022.116830.

[25] Myungeun Eom and Byung-In Kim. The traffic signal control problem for intersections: A review. *European Transport Research Review*, 12(1):50, 2020. doi:10.1186/s12544-020-00440-8.

[26] Saad Touhbi, Mohamed Amine Babram, Tri Nguyen-Huu, Nicolas Marilleau, Moulay L. Hbid, Christophe Cambier, and Serge Stinckwich. Adaptive traffic signal control: Exploring reward definition for reinforcement learning. *Procedia Computer Science*, 109:513–520, 2017. doi:10.1016/j.procs.2017.05.327.

[27] Wangzhi Li, Tianheng Zhu, and Yiheng Feng. A cooperative perception based adaptive signal control under early deployment of connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 169: 104860, 2024. doi:10.1016/j.trc.2024.104860.

[28] Xingmin Wang, Zachary Jerome, Zihao Wang, Chenhao Zhang, Shengyin Shen, Vivek Vijaya Kumar, Fan Bai, Paul Krajewski, Danielle Deneau, Ahmad Jawad, Rachel Jones, Gary Piotrowicz, and Henry X. Liu. Traffic light optimization with low penetration rate vehicle trajectory data. *Nature Communications*, 15(1):1306, 2024. doi:10.1038/s41467-024-45427-4.

[29] Tianjia Yang and Wei Fan. Transit signal priority under connected vehicle environment: Deep reinforcement learning approach. *Journal of Intelligent Transportation Systems*, 29(5):505–517, 2025. doi:10.1080/15472450.2024.2324385. Published online in 2024.

[30] Alireza Talebpour and Hani S. Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71:143–163, 2016. doi:10.1016/j.trc.2016.07.007.

[31] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000. doi:10.1103/PhysRevE.62.1805.

[32] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, 2018. doi:10.1109/ITSC.2018.8569938.