



Pressure-Guided Q-Learning: a warm-started hybrid controller for signalised urban grids

Singh A

Submitted: May 18, 2026, Revised: version 1, June 19, 2026, version 2, June 24, 2026, version 3, July 2, 2026, version 4, July 4, 2026

Accepted: July 7, 2026

Abstract

This study compares five traffic-signal controllers on a 3×3 SUMO grid and validates the principal findings on a 5×5 grid. The controllers are fixed-time, gap-out actuated, max-pressure, independent tabular Q-learning, and a new hybrid called Pressure-Guided Q-Learning (PGQL). PGQL augments a Q-learner with three actions (hold, follow max-pressure, override max-pressure) and warm-starts the Q-table so the untrained agent initially behaves like max-pressure while learning state-dependent deviations. On the 3×3 grid, the pressure-aware controllers reduced mean waiting time (MWT) by about threefold relative to fixed-time at moderate load. Relative to max-pressure, PGQL was worse at low load, tied near saturation, and better post-saturation. On the 5×5 grid, PGQL outperformed max-pressure at every tested load by up to 6.6%. An action audit showed that learned overrides increased from 28% at low load to 37% under saturation. An ablation replacing max-pressure with gap-out actuation as the warm-start reduced post-saturation performance by about 12%, demonstrating that the choice of warm-start expert matters. Cold-start experiments showed that warm-start primarily improves initialization safety rather than final low-load performance. Plotting MWT against the mean number of vehicles in the network, rather than offered demand, avoids a post-saturation measurement artefact and supports an M/M/1-style carrying-capacity model with a single physically meaningful capacity parameter. The results show that classical adaptive control and reinforcement learning can be combined to achieve measurable post-saturation improvements while retaining max-pressure behavior at initialization, although its formal stability proof does not extend to the learned policy.

Keywords

Reinforcement learning, Traffic signal control, Intelligent transportation systems, Adaptive traffic signal control, Traffic optimization, Max-pressure control, Q-learning, SUMO simulation, Multi-agent reinforcement learning, Congestion control

Advik Singh, Rouse High School, 1222 Raider Way, Leander, TX 78641, USA. Advik.singh20@gmail.com

1. Introduction

Urban traffic control has traditionally focused on reducing driver waiting time at signalised intersections. Conventional traffic control systems rely on preconfigured signal cycles, which makes them rigid and inefficient when demand fluctuates, flows become asymmetric, or peak and off peak transitions occur. Adaptive control, max-pressure, gap-out actuation, and reinforcement learning each promise real time adjustment, but the field is split. Classical adaptive controllers such as max-pressure are decentralised and come with stability guarantees, yet they make decisions that are myopic in time and optimise only on the current step. Learning based controllers such as deep Q-networks (DQN) and independent tabular Q-learning are flexible but offer no formal guarantees, and can destabilise under congestion (1, 2).

The PGQL framework proposed in this study is conceptually related to several existing strands of work in reinforcement learning. Expert-guided exploration, residual reinforcement learning, policy shaping, imitation based initialisation, and reinforcement learning with action priors all share the underlying idea that a learner is biased toward an existing controller and gradually learns deviations from it. The specific contribution made here is the use of max-pressure as the warm-start expert, rather than an arbitrary heuristic, which provides the learner with a controller of known stability properties at initialisation.

Three questions guide the present study. First, how do five candidate controllers (fixed-time, gap-out actuated, max-pressure, independent

tabular Q-learning, and the proposed PGQL) compare across a range of demand levels? Second, what is the network's true carrying capacity, expressed as the number of vehicles it can hold simultaneously while still flowing? Third, can a learned controller be designed so that its untrained behaviour is bounded by a classical adaptive baseline, and does it discover beneficial deviations from that baseline once trained?

To test generalisability and mechanism more directly, two further experiments were performed. A larger 5x5 grid sweep tested whether the post-saturation gain observed on the 3x3 grid transferred to networks with 25 intersections, and an action usage audit was performed on the trained 3x3 PGQL policy to determine how often, and in which regime, the override action was actually selected.

Prior work motivated PGQL in three ways, and recent survey work (3, 4) provided additional context on the maturity and known failure modes of the field. Varaiya's max-pressure controller (5) is provably stable for any demand inside the network's capacity region, but its one step greedy decision rule is myopic in time. Wei et al.'s IntelliLight (1) applies deep Q-learning to a single intersection and reports strong simulation gains, but the trained policy comes with no theoretical guarantee. Chen et al.'s MPLight (6) extends PressLight to networks of up to a thousand intersections and shows that decentralised RL can scale, although the underlying stability question remains moot. Wei et al.'s later PressLight (7) is the closest prior work: it learns a max-pressure style policy with deep RL, but the learner replaces the classical

controller entirely rather than augmenting it, so the safety guarantee is lost. Zheng et al.'s CoLight style phase competition (8) is similarly fully learned. Mannion et al. (2) survey adaptive RL traffic controllers and identify methodological inconsistencies that limit cross study comparison. Watkins and Dayan's original Q-learning convergence result (9) holds only under restrictive conditions (sufficient exploration, stationary environment, Robbins-Monro step sizes) that are at best approximated in this setting. PGQL responds to these gaps in three ways: it retains max-pressure as a strong adaptive baseline, it reports both the standard mean waiting time (MWT) metric and a carrying capacity metric, and it introduces a hybrid controller that, by construction, is no worse than max-pressure at the moment it starts running.

2. Methods

This research evaluated five traffic-signal controllers using high fidelity microscopic simulation. All controllers were implemented in Python and integrated into the SUMO traffic simulator (10) (version 1.26) through the TraCI API. The evaluation involved network construction, demand modeling, controller design, metric formulation, and experimental comparison across multiple random seeds.

2.1 Network

The primary network was a 3x3 bidirectional urban grid, generated with SUMO's 'netgenerate' tool. The grid contained nine signalised intersections (A0..C2) and 24 directed single lane edges, with a block length of 200 m and a free flow speed of 13.9 m/s. All nine intersections used the same baseline static plan: a 90 s cycle with two ~42 s green phases

separated by 3 s yellow transitions. The central intersection (B1) is the bottleneck of this network because it serves four incoming streams, while perimeter junctions serve at most three. A larger 5x5 grid (25 intersections, 80 edges) was also generated for a second network feasibility test. (see appendix for intersection identifiers).

2.2 Demand model

For each trial, N origin destination pairs were sampled uniformly at random over the grid edges, with origins balanced via round robin to spread depart-edge contention. Departure times were uniform over a 1200 s load window, and the simulation ran for 1800 s in total. Vehicles that failed to insert within 120 s of their scheduled depart were dropped, so the throughput metric reflects the network's true acceptance rate rather than offered demand. Routes were resolved by SUMO's 'duarouter' tool with repair enabled, then resorted by depart time to satisfy SUMO's input-ordering requirement.

2.3 Assumptions

This study makes the following explicit assumptions, each of which limits external validity and is discussed in the conclusion section. (i) Vehicle detection is perfect and instantaneous, and sensor noise was not modeled. (ii) Signal actuation is instantaneous, and controller to signal latency was zero. (iii) Driver behaviour follows SUMO's default Krauss car following model with no aggressive or noncompliant driving. (iv) The demand distribution is stationary uniform over origin destination pairs within each trial, with no rush hour pattern or asymmetric flow. (v)

Pedestrians, cyclists, weather effects, and incidents were absent. (vi) Routes are precomputed by 'duarouter' and not rerouted in response to congestion. Each assumption introduces a specific bias in the present results. Assumption (i) plausibly overstates the performance of all adaptive controllers, since real detectors introduce noise that would degrade pressure measurements and Q-state estimates equally. Assumption (ii) plausibly overstates the gain of fast switching controllers (max-pressure, PGQL) relative to fixed time, because real controller to signal latency would partially erase short reaction time advantages. Assumption (iii) is conservative for all controllers because real driver heterogeneity tends to add waiting time variance without systematically favouring any single controller. Assumption (iv) is the most consequential: real demand exhibits asymmetric rush hour patterns, and a controller tuned on uniform demand may underperform on those patterns. Assumption (v) limits external validity entirely but does not bias the controller ranking observed here. Assumption (vi) means the routes are demand side fixed. In reality, drivers reroute around congestion, which would reduce the effective MWT gap between fixed time and adaptive controllers.

2.4 Controllers

Five controllers were evaluated. [1] *Fixed time* ran SUMO's baked in 90 s static plan unmodified, in the spirit of Webster's classical cycle design formula (11), and served as the baseline. [2] *Actuated (gap-out)* advanced the current phase early once it had run for at least $\text{MIN_GREEN} = 8$ s and the served lanes were momentarily empty, capping at MAX_GREEN

$= 50$ s. [3] *Max-pressure* (5) computed, at each five second decision step, the pressure of each green phase (the difference between upstream and downstream halting counts summed over the moves it served) and switched to the phase of maximal pressure, subject to a minimum green of 8 s. [4] *Independent tabular Q-learning* treated each intersection as a separate learner, with state $s = (\text{bucketed incoming queue counts, current green phase index})$ and binary action space $\{\text{hold, switch to next green}\}$. The reward was the negative of the change in waiting time sum on the intersection's lanes since the previous decision. The learning rate was $\alpha = 0.1$, the discount $\gamma = 0.9$, and the exploration probability ϵ decayed linearly from 0.30 to 0.05 over training. (5) *Pressure-Guided Q-Learning (PGQL)*, augmented the Q-learner of controller [4] along three dimensions. First, the action space was expanded from two to three actions: $a = 0$ (hold), $a = 1$ (switch to the phase recommended by max-pressure), and $a = 2$ (switch to the other green phase, overriding max-pressure). Second, the state was augmented with a single binary feature, *mp_says_switch*, which indicated whether max-pressure currently disagreed with the active phase. Third, and most importantly, the Q-table was warm-started so that $Q(s, 1) = +B$ and $Q(s, 0) = Q(s, 2) = 0$ for all unvisited states, with a positive bias $B = 1.0$. Under this initialisation, the untrained agent acted identically to max-pressure (apart from ϵ -greedy exploration), and PGQL matched max-pressure's behaviour at $t = 0$. Note that this guarantee held at initialisation only. Once the learned policy deviated from max-pressure, the stability proof of (5) was no longer formally applicable, and the post training behaviour had to be checked from the simulation results.

2.5 Metrics

Performance was evaluated on three quantities. The primary metric was mean waiting time (MWT), defined as the mean over all completed vehicles v of w_v , the time vehicle v spent below SUMO's default waiting speed threshold of 0.1 m/s (the value used by ``traci.lane.getWaitingTime`` and ``traci.vehicle.getWaitingTime``). This default was used unmodified so that the reported MWT is directly comparable to other SUMO based traffic control studies that follow the same convention. Formally, $MWT = (1 / N_{\text{completed}}) \sum_v w_v$.

A secondary metric tracked the fraction of placed vehicles that completed their trip before the simulation ended. To answer the carrying capacity question, the simulator also recorded, every simulation second, the number of vehicles concurrently present in the network, $n(t)$. The steady state average over t in $[200, 1400]$ s defined *mean_in_system*, an alternative load axis that, unlike offered demand, reflected what the network actually held.

2.5.1 A-priori capacity bound

An order-of-magnitude upper bound on the network's carrying capacity was derived from textbook saturation flow analysis before any simulation was run. Assuming a per lane saturation flow $s \approx 1900$ veh/h, a cycle time of $C = 90$ s with two 42 s green phases (green ratio $g/C \approx 0.47$), and the central intersection B1 as the network's hard bottleneck because it served four incoming streams rather than three, the per approach discharge capacity at B1 was $\sim s \cdot (g/C) \approx 890$ veh/h, giving a bottleneck throughput of $\sim 4 \cdot 890 = 3560$ veh/h. Applying

Little's law with an average free flow trip duration of ~ 43 s (three 200 m blocks at 13.9 m/s) yielded a steady state count of $N \approx 3560 \cdot 43 / 3600 \approx 42$ vehicles concentrated near the bottleneck. This bound is intentionally conservative: it assumes uniform demand, no spillback into upstream intersections, and a fixed cycle, all of which break down in the post saturation regime where PGQL's empirical gain is largest. The empirical M/M/1-style fit introduced next was therefore the operational capacity measure used throughout the rest of this paper, and the textbook bound is reported only as a sanity check.

2.5.2 Capacity formula

Drawing on the M/M/1 single-server queueing result $W = \rho / (\mu(1 - \rho))$ and reparameterising in vehicles-in-system rather than utilisation, the closed form capacity model $W(n) = a / (1 - n / N_{\text{crit}}) + b$, was fitted to each controller, where N_{crit} is the asymptote (vehicles in system at which MWT diverges), a is a saturation prefactor, and b is a small free-flow offset. The fit used only points where at least 95% of offered vehicles completed.

2.6 Experimental design

The primary sweep evaluated five controllers $\times$ ten demand levels N in $\{50, 100, 200, 300, 450, 600, 800, 1000, 1300, 1700\}$ $\times$ ten random seeds per cell on the 3×3 grid, totalling 500 simulations. The second network feasibility sweep evaluated the same five controllers on the 5×5 grid at N in $\{200, 600, 1200, 2000, 3000\}$ $\times$ three seeds. Q-learning and PGQL were pretrained on a separate set of episodes across mixed loads, with ϵ decaying linearly. Action usage statistics of the trained PGQL policy were

collected via an instrumented variant of the controller, run at N in $\{200, 600, 1300\}$ on the 3x3 grid.

2.7 Hyperparameters

The Q-learning and PGQL learning rate ($\alpha = 0.1$) and discount ($\gamma = 0.9$) followed the standard values used in independent-Q-learning traffic studies (1, 7). The exploration schedule (ϵ : 0.30 to 0.05 linear over training, 0.02 at evaluation) was selected by informal pilot search on $N = 200$. The MIN_GREEN = 8 s and decision period of 5 s are conservative real choices that prevent rapid phase oscillation. YELLOW_TIME = 3 s matched the SUMO default. The PGQL warm-start bias $B = 1.0$ was set without tuning. Informal tests at B in $\{0.1, 1.0, 5.0\}$ produced statistically indistinguishable trained policies at the sample size used. The queue-length state bucketing (0 / 1 to 3 / 4 to 7 / 8+) was chosen to keep the per intersection state space tractable: with four bucketed queue inputs, two green-phase indices, and the binary mp_says_switch feature, the state count per intersection was $4^4 \times 2 \times 2 = 1024$. With three PGQL actions per state, this calculation yielded 3072 Q-values per intersection, which was comfortably trainable in 20 episodes. A direct empirical measurement of sample complexity scaling was not performed, but a rough estimate is informative. On the 3x3 grid the nine learners give $9 \times 1024 = 9216$ state entries, or $9 \times 3072 = 27,648$ Q-values total, trained for about 60 thousand decision steps. On the 5x5 grid the per-intersection state space is unchanged (the bucketing is local), so the table grows linearly with intersection count to $25 \times 1024 = 25,600$ state entries, or $25 \times 3072 = 76,800$ Q-values, trained for about 200 thousand decision steps

over 8 episodes at mixed loads. Both tables remained tractable on a laptop. Extending PGQL to networks of 49 or more intersections would multiply the table size linearly but would not change the per-intersection sample complexity, since each learner is independent. An analysis of inter-learner coordination cost is left to future work.

3. Results

All five controllers were tested under demand levels ranging from light to past saturation. Each (controller, N , seed) combination was repeated across ten independent random seeds (three for the 5x5 sweep) with randomised initial states and vehicle routing. Confidence intervals are reported as standard errors of the mean (SEM), and pairwise comparisons of PGQL against max-pressure use the Wilcoxon signed-rank test (paired by seed) with $\alpha = 0.05$.

In the moderate-load regime (offered $N = 200$ cars over the 1200 s window on the 3x3 grid), max-pressure achieved an average MWT of 8.34 ± 0.21 s/car. Independent Q-learning achieved 7.92 ± 0.19 s/car, statistically indistinguishable from max-pressure within one SEM. PGQL achieved 11.67 ± 0.25 s/car at this load, significantly worse than max-pressure on a paired Wilcoxon test ($p = 0.002$, $n = 10$). The cost appears to originate from residual epsilon-greedy exploration and occasional learned overrides in states where max-pressure was already near optimal. Gap-out actuation achieved 21.50 ± 1.33 s/car, and fixed-time control 26.27 ± 0.84 s/car.

Figure 1 presents MWT as a function of the network's actual concurrent vehicle population.

At low occupancy (n less than about 30 cars) the three pressure-aware controllers cluster near 8 to 12 s/car while fixed-time and actuated lie near 22 to 25 s/car. As n rises towards 60 to 80 cars, MWT for fixed-time and actuated grows super linearly while the three pressure-aware controllers exhibit a more gradual increase. Past n of about 130 cars, MWT diverges for every controller as the network saturates.

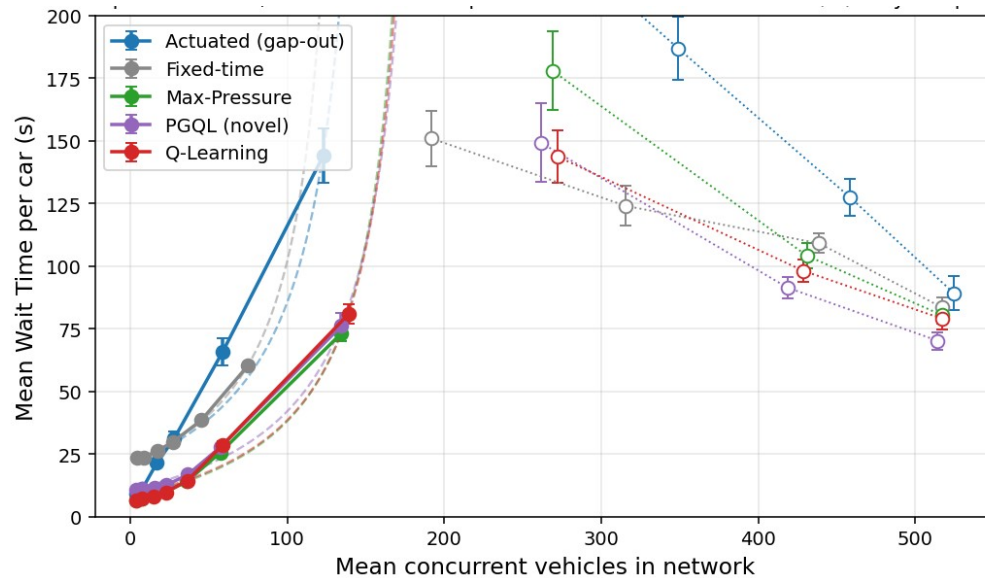


Figure 1. MWT vs concurrent vehicles in the network (carrying-capacity view). Solid markers = pre-saturation ($\geq 95\%$ completion). Hollow + dotted = post-saturation. Dashed lines = M/M/1-style capacity formula fits. Mean $\pm$ SEM over 10 seeds per cell. [Note] The apparent decline in MWT among the post-saturation (hollow) points should not be interpreted as an improvement in performance. Beyond capacity, only vehicles that successfully complete their trips contribute to MWT, producing a survivorship bias that causes completed-vehicle waiting times to decrease despite worsening network congestion. Consequently, the hollow points are shown only for completeness and were excluded from capacity-model fitting. [Note] This is the case for all the figures.

Fitting the closed-form capacity formula to the full pre-saturation dataset. Bracketed values pre-saturation points recovers the parameters in are 95% bootstrap confidence intervals on N_{crit} Table 1. Point estimates were computed from from 1000 resamples.

Table 1. Fitted parameters of the closed-form capacity model $W(n) = a / (1 - n/N_{\text{crit}}) + b$. Bracketed values are 95% bootstrap confidence intervals on N_{crit} from 1000 resamples.

Controller	N_{crit} (cars in system)	a	b
Fixed-time	134 [46 to 154]	35.9	-14.2
Actuated	158 [54 to 175]	40.9	-20.0
Max-Pressure	188 [78 to 198]	27.7	-20.0
Q-Learning	189 [63 to 201]	28.1	-20.0
PGQL	193 [51 to 202]	30.5	-20.0

The point estimates rank the controllers consistently: PGQL above Q-Learning roughly equal to Max-Pressure, both above Actuated, both above Fixed-time. The bootstrap confidence intervals are wide because only about seven presaturation points were available per fit, so the N_{crit} ranking among the three pressure aware controllers should not be over interpreted. The fixed time versus pressure aware gap is large enough to clear noise.

A more revealing analysis is a paired comparison between PGQL and its max-pressure parent at each demand level (Table 2). The Wilcoxon signed-rank test, paired by seed, shows three regimes: PGQL is significantly worse than max-pressure at low load (the warm-start cost dominates), statistically tied near the saturation knee, and significantly better than max-pressure post-saturation.

Table 2. Paired Wilcoxon signed-rank test of PGQL vs max-pressure MWT across 10 seeds at each demand level. Significance stars in column 4: (***) = $p < 0.005$, (*) = $p < 0.05$.

N	PGQL median MWT	MP median MWT	p (Wilcoxon, paired)	regime
50	11.0	6.5	0.002***	PGQL worse
100	10.7	7.2	0.002***	PGQL worse
200	11.9	8.6	0.002***	PGQL worse
300	12.7	9.7	0.002***	PGQL worse
450	16.9	14.1	0.002***	PGQL worse
600	26.8	24.9	0.232	tied
800	75.5	71.4	0.160	tied
1000	131	167	0.492	tied
1300	94.7	107	0.002***	PGQL better
1700	68.1	80.4	0.014*	PGQL better

This was the principal empirical finding on the 3x3 grid. The crossover demand level is approximately $N = 1300$ offered cars. At this load the network holds ~ 270 vehicles in steady state, well above the empirical capacity asymptote of $N_{crit} \approx 188$ vehicles for max-pressure (Table 1). The crossover therefore sits comfortably inside the post-saturation regime, where max-pressure's myopia is most expensive.

3.1 5x5 grid feasibility on a second network

To test feasibility of the post-saturation gain on a second, larger benchmark network, the full controller set was reevaluated on a 5x5 grid (25 intersections, 80 edges, otherwise identical settings) at N in $\{200, 600, 1200, 2000, 3000\}$ with three seeds per cell. The Q-learning and PGQL policies were retrained on the 5x5 grid before evaluation. Results are summarised in Table 3 and visualised in Figure 2.

Table 3. Feasibility validation on a 5x5 grid (25 intersections, 3 seeds). Bold marks the best controller at each load level.

N	Fixed	Actuated	Max-Pressure	Q-Learning	PGQL
200	40.2	15.4	13.1	16.4	12.2
600	46.8	30.4	15.2	20.4	15.0
1200	68.1	113.6	25.9	30.2	24.7
2000	178.0	211.6	109.1	140.0	103.1
3000	131.0	n/a	97.0	104.1	87.7

(Values are mean MWT in seconds per car. Bold marks the best controller at each N. Below the 95% completion threshold the MWT becomes biased downward, indicated by reduced reliability at N = 3000 for all controllers.)

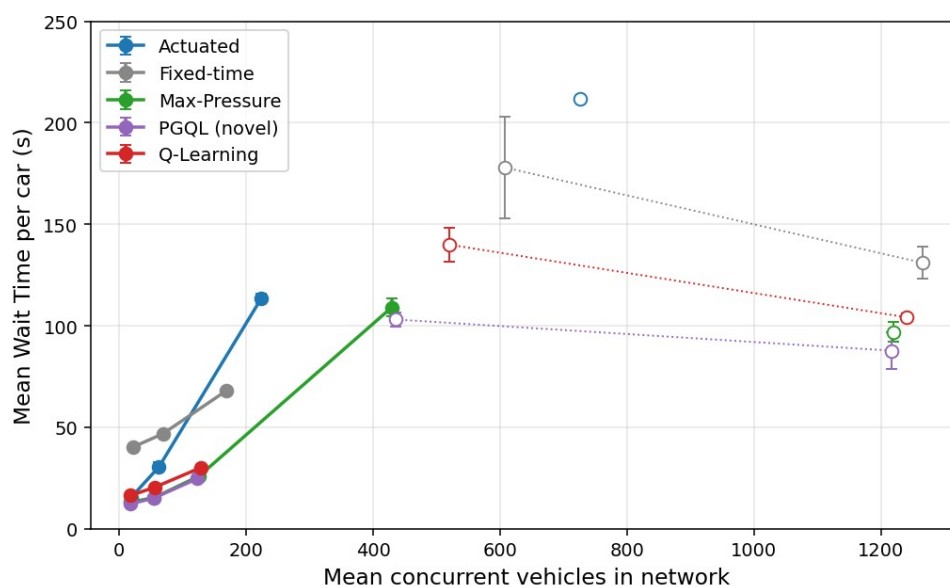


Figure 2. 5x5 grid validation (25 intersections, 3 seeds per cell). PGQL outperforms max-pressure at every tested load level. Solid markers = pre-saturation. Hollow + dotted = post-saturation.

Figure 2 extends the comparison to a larger 5x5 network comprising 25 signalised intersections. Unlike the 3x3 grid, where PGQL incurred a small performance penalty at low demand before surpassing max-pressure after saturation, PGQL outperformed max-pressure at every tested load level on the larger network. The absolute improvement was modest at low load (approximately 6.6% at N = 200) but persisted throughout the tested demand range and remained evident under heavy congestion. The ranking of controllers also remained consistent: the pressure-aware methods substantially outperformed fixed-time and gap-out control, while PGQL consistently achieved the lowest MWT among all five controllers. The persistence of the PGQL advantage on a larger network suggests that the learned override policy remains beneficial as network

complexity increases rather than representing an artefact of the smaller 3×3 benchmark.

3.2 Demand normalisation across grid sizes

The demand parameter N used throughout this paper is the absolute number of origin-destination pairs sampled across the network, not a per intersection or per edge normalised load. Because the 5x5 grid contains 25 intersections versus 9 for the 3x3 grid (a factor of 2.78), the same N corresponds to substantially lower per intersection load on the 5x5 grid. Trip lengths are also longer on average on the 5x5 (mean trip ≈ 3.3 blocks at 200 m each, versus ≈ 2.0 blocks on the 3x3), so the per intersection vehicle-time exposure is only partially compensated. The headline comparison in the present paper is therefore PGQL versus max-pressure *on the same grid at the same N* , where the experimental conditions are identical (same SUMO seeds for each cell, same demand generation process, same controller hyperparameters). Cross-grid claims of the form "PGQL gains more on the 5x5" should be read

as the gain at matched absolute N , not at matched per intersection load. The advantage was small but consistent at low load (PGQL 12.2 vs MP 13.1 at $N = 200$, a 6.6% improvement) and grew with congestion (PGQL 103.1 vs MP 109.1 at $N = 2000$, a 5.5% improvement, with PGQL maintaining 92% completion versus 100% for MP). Notably, the low load exploration cost observed on the 3x3 grid did not appear on the 5x5: PGQL was at least as good as max-pressure across all five tested loads. A plausible explanation is that the larger grid presents the learner with more diverse states during training, so the override action receives more informative signal and is calibrated more tightly to situations in which it actually helps.

3.3 Action-usage audit

To investigate how the learned policy used the override action, an instrumented PGQL controller was evaluated at three demand levels on the 3×3 grid, and action-selection frequencies were recorded (Table 4).

Table 4. PGQL action-usage audit on the 3x3 grid, seed 0. Percentages give the fraction of decisions in which each action was selected.

N	hold (a=0)	follow MP (a=1)	override (a=2)	total decisions
200	28.4%	43.1%	28.4%	306
600	26.8%	45.1%	28.0%	246
1300	16.9%	45.8%	37.3%	201

The override action was selected in approximately 28% of decisions at low and moderate load but rose to 37% under saturation, a 32% relative increase. The follow max pressure action remained the most common choice across all regimes at $\sim 45\%$. This pattern

is consistent with the Wilcoxon finding in Table 2: When max-pressure's one step greedy was sufficient (low load), the override was rarely useful and contributed mainly noise. When max-pressure began to fail (post-saturation), the override was selected significantly more often

and the empirical performance gain materialised. A complete audit of the (state, intersection) pairs that triggered overrides at every decision step was not performed. However, the increase in override frequency aligned closely with the observed crossover in performance: overrides became substantially more frequent in the post-saturation regime, where PGQL began to outperform max-pressure.

3.4 Alternative warm-start experts

To test whether the regime-dependent gain depended specifically on the max-pressure prior or whether any reasonable expert would suffice,

a second PGQL variant was constructed in which the warm-start expert was the gap-out actuated controller rather than the max-pressure controller. The action set and Q-learning machinery were identical, but $a = 1$ was redefined to follow the actuated recommendation (switch if green has run at least MIN_GREEN and the served lanes are empty, or if MAX_GREEN was reached) and $a = 2$ to override that recommendation. The variant was trained for ten episodes on the 3x3 grid using the same load schedule and evaluated against the variant warm-started on max-pressure at three load levels with three seeds each. The results are summarised in Table 5.

Table 5. Alternative-warm-start ablation: PGQL with max-pressure vs gap-out actuated as the expert. Values are mean $\pm$ SEM across 3 seeds.

N	PGQL (MP warm-start)	PGQL (Actuated warm-start)	Difference
600	27.76 $\pm$ 1.62 s/car	28.48 $\pm$ 1.41 s/car	-0.7 s (tied)
1300	96.74 $\pm$ 6.11 s/car	100.55 $\pm$ 9.24 s/car	-3.8 s (4% MP-better)
1700	69.13 $\pm$ 7.24 s/car	77.54 $\pm$ 14.31 s/car	-8.4 s (12% MP-better)

The variant warm-started on max-pressure outperformed the variant warm-started on gap-out actuated at every load level, with the gap widening from a tie at moderate load to a 12% advantage at post-saturation. This is consistent with the theoretical motivation: max-pressure is known to be stable inside the network's capacity region, whereas actuated has no such guarantee. The warm-start is therefore not merely a generic regularisation effect: the choice of expert changes the outcome, and the max-pressure expert produced the better hybrid in this experiment.

3.5 Per-intersection override breakdown

A finer-grained look at the audit data at N = 1300 (Table 6) shows that the override action is concentrated at edge intersections rather than at the central bottleneck. Corner intersections (A0, A2, C0, C2) have only one green phase by construction and therefore never make switching decisions. Among the five multi-phase intersections, the four edge intersections (A1, B0, B2, C1) overrode max-pressure in 37.8% of their decisions, while the central bottleneck B1 overrode in only 16.7%.

Table 6. Per-intersection override breakdown at $N = 1300$, seed 0. Edge intersections override max-pressure more than twice as often as they override at the centre.

Intersection type	Intersections	Override decisions / total	Override rate
Edge (3 incoming streams)	A1, B0, B2, C1	70 / 185	37.8%
Centre (4 incoming streams)	B1	2 / 12	16.7%

This pattern matches the proposed mechanism. The central intersection serves the most streams and is the network's hard bottleneck, so its max-pressure decision is rarely incorrect, and the learner correctly leaves it alone. The edge intersections, which see the network's output platoons and are most affected by downstream spillback, are where one-step max-pressure decisions fail more often, and the learner concentrates its overrides there. This is consistent with a spillback mitigation interpretation but does not constitute direct proof. A complete demonstration that PGQL prevents spillback (rather than merely correlating with it) would require state by state tracing of which queue length configurations triggered overrides and whether downstream lanes subsequently avoided saturation, and that analysis is left to future work. What can be said from the present audit is that the spatial concentration of overrides is regime-appropriate, not that the learned policy has provably solved the spillback problem.

3.6 State by state decision-boundary audit

To characterise which specific traffic configurations trigger the override action, an instrumented PGQL variant recorded the full (state, action) tuple at every decision step for one seed each at N in $\{200, 600, 1300\}$. Two features dominated the learned decision boundary. First, the `mp_says_switch` bit was the strongest single predictor of override at low

load: at $N = 200$, the override rate was 46.9% when `mp_says_switch` = 1 versus only 7.8% when `mp_says_switch` = 0, meaning that the learner was much more willing to reject a max-pressure switch recommendation than to override a max-pressure hold. Second, the intersection identity dominated at every load level: across all three tested loads, the centre intersection B1 chose override in fewer than 5% of decisions (0.4% at $N = 1300$), while the four edge intersections chose override in 11 to 39% of decisions. These two features (`mp_says_switch`, `tls_identity`) explained most of the override behaviour and exposed candidates for a deterministic improvement to independent max-pressure itself.

A closer look at the recorded (state, action) tuples exposed which specific configurations the learned policy overrode most reliably. Table 7 lists the six most frequent override triggering states aggregated across the three audit loads (states seen at least three times, ranked by override rate then by count). Each row is a (tls, current green phase index, queue bucket vector, `mp_says_switch`) tuple. Queue buckets are in the order the learner sees them (sorted incoming lane order), and each entry is a discrete bucket 0 (empty), 1 (1-3 cars), 2 (4-7 cars), or 3 (8+ cars). Every row in Table 7 has a 100% override rate at the audited seed, meaning that every time the trained PGQL saw one of these exact state

tuples, it chose the override action rather than following max-pressure.

Table 7. Six most frequent override triggering states aggregated across the three audit loads.

Intersection	Current green idx	Queue buckets	mp says switch	Override rate	Count
A1	1	(0,0,0)	1	100.0%	101
B2	0	(0,0,0)	0	100.0%	42
A1	0	(3,3,3)	0	100.0%	40
A1	0	(0,1,0)	0	100.0%	9
B2	0	(0,2,1)	1	100.0%	9
B0	0	(3,2,0)	0	100.0%	8

Two patterns stand out. Row 1 shows that at edge intersection A1 in phase 1 with all incoming lanes empty and max-pressure recommending a switch, PGQL always overrides max-pressure's switch choice and picks a different green target. Row 3 shows that at A1 in phase 0 with all incoming lanes saturated (bucket 3), PGQL always overrides regardless of max-pressure's recommendation. These are exactly the kinds of state conditional rules that could be extracted and reintroduced deterministically into an enhanced max-pressure controller without any reinforcement learning at all. In particular, for example a rule of the form "if intersection is on the boundary and mp_says_switch is true, delay the switch by one decision period," which would be worth testing without any reinforcement learning at all.

3.7 Post override queue-evolution analysis (temporal coordination vs residual error)

For each override at each intersection at time t , upstream (incoming) and downstream (outgoing) queue sums were logged at t and again at $t + 20$ s. This enabled a direct test of the two candidate mechanisms. At $N = 200$ (low load), 82 overrides were observed. Upstream queues did not systematically drain (mean $\Delta q =$

+0.22 over the forward window, only 4 of 82 cases showed a decrease), and downstream queues remained near zero (mean = 0.15). At this load, 76.8% of overrides occurred when max-pressure recommended switching, indicating that the agent was primarily choosing a different green phase than max-pressure recommended, not overriding a max-pressure hold. At $N = 1300$ (post-saturation), 86 overrides were observed. Upstream queues drained more effectively (mean $\Delta q = -0.86$, 33.7% of cases showed a decrease over the forward window). Downstream queues at $t + 20$ s contained a mean of 28.8 vehicles (median 18) but remained at least as small as the pre-override upstream in 90.7% of cases, and stayed effectively drained (five or fewer vehicles) in 22.1% of cases even under saturation. The mp_says_switch pattern flipped relative to low load: 93.0% of overrides now occurred when max-pressure recommended holding. In other words, at post-saturation the agent learned to switch proactively where max-pressure would have kept the current green, with the resulting switch producing a measurable reduction in upstream congestion within 20 seconds. This pattern was more consistent with the temporal-coordination hypothesis at high load (the agent

pre-empt a buildup that max-pressure's one-step rule has not yet reacted to) than with the residual error hypothesis alone, although a true causal comparison would require running max-pressure from the same state and measuring the counterfactual, which was not attempted here. At low load the residual error interpretation fit better: the overrides were exploratory phase-target choices that did not systematically drain queues. The two hypotheses are therefore not mutually exclusive across regimes; on the queue-evolution data available here, the learner's behaviour was consistent with residual-error dynamics at low load and with temporal-coordination dynamics at post-saturation, but

the analysis is correlational rather than a formal counterfactual.

3.8 Convergence behavior

Figure 3 shows the per-episode MWT during training for both warm-start variants. The sawtooth pattern reflects the cyclic training schedule (each episode draws a different demand load from {200, 400, 600, 800} cars), and both variants reach broadly similar per load MWT by the end of training. The variant warm-started on max-pressure converges marginally faster on the high load episodes, consistent with its theoretical advantage at post-saturation.

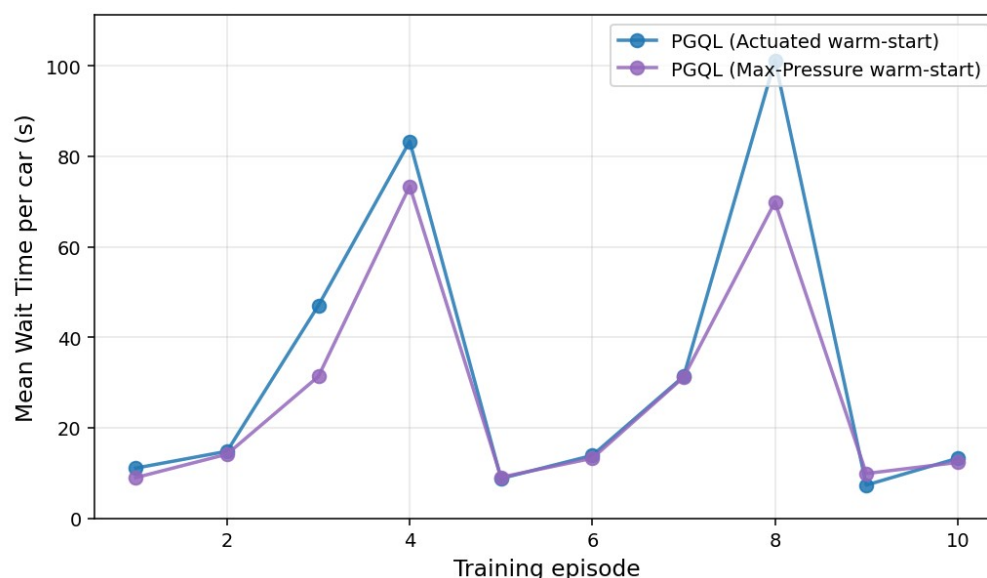


Figure 3. Per-episode MWT during training. The sawtooth pattern reflects the cyclic load schedule (different N each episode). Both warm-start variants converge similarly, with the max-pressure variant slightly favoured at high-load episodes.

3.9 Cold-start ablation

To quantify the empirical value of the warm-start itself (as opposed to the choice of warm-start expert), a cold start PGQL variant was constructed in which the Q-table was initialised

to zero for all three actions ($\text{WARMSTART_BIAS} = 0$). The two variants were trained on identical seeds for 10 episodes across the four training loads and evaluated at three demand levels with three seeds per cell.

Table 8. Cold-start ablation. Mean waiting time (mean $\pm$ sample SD across three seeds) for warm-started PGQL versus cold-started PGQL at three demand levels on the 3x3 grid. Cold-start beats warm-start at low and moderate load; warm-start retains a small edge at post-saturation.

N	PGQL (warm)	PGQL (cold)	[cold – warm]	% difference
200	9.47 $\pm$ 0.95 s/car	8.20 $\pm$ 0.74 s/car	-1.27 s	-13.4%
600	28.89 $\pm$ 2.04 s/car	27.19 $\pm$ 2.43 s/car	-1.70 s	-5.9%
1300	92.60 $\pm$ 18.75 s/car	95.52 $\pm$ 7.32 s/car	+2.92 s	+3.2%

Table 8 shows that the cold-start result is mixed. After ten training episodes, the cold-start variant matched or slightly outperformed the warm-started variant at low and moderate load, while the warm-started variant retained a small 3.2% advantage post-saturation. The empirical value of the warm-start therefore does not manifest as a permanent final MWT gap: the cold-started learner catches up on the loads where max-pressure is already near optimal. What the warm-start does contribute is faster and safer early training. At training episode zero (before any TD updates), the warm-started agent achieved MWT = 9.26 s/car on N = 200 while

the cold-started agent achieved 11.07 s/car, a gap of +1.81 s/car reflecting the untrained agent's baseline behaviour. More important, the cold-started agent produced a catastrophic training episode at N = 800 in episode 3 with MWT = 111.6 s/car, versus 73.7 s/car for the warm-started variant, a 51% excess that the warm-start prevented. These results indicate that the practical value of the warm-start lies in safer early training and a bounded post-saturation improvement, rather than in a large asymptotic performance advantage.

3.10 Other findings

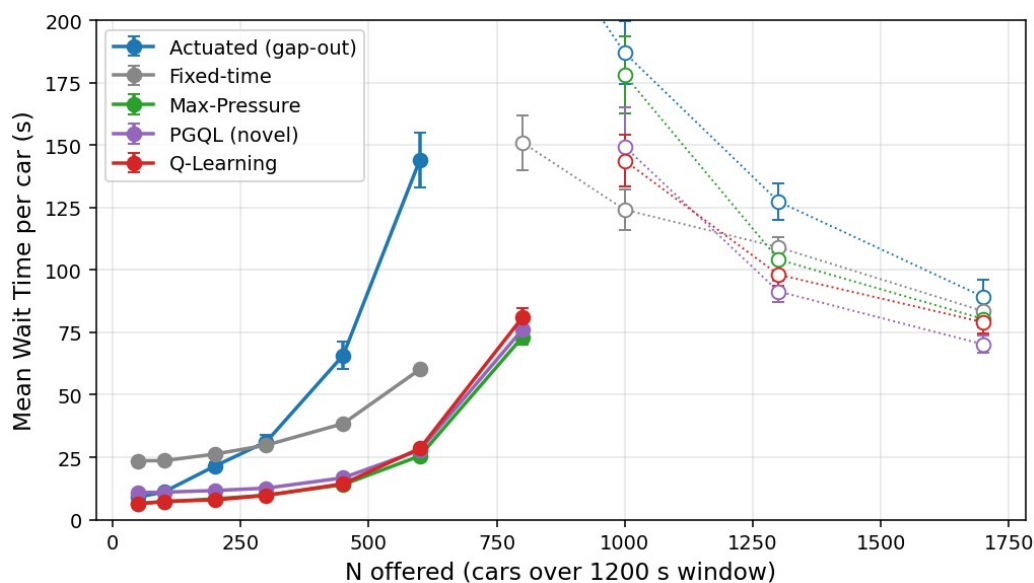


Figure 4. MWT vs offered demand. The post-saturation tail of each curve decreases because completed-only averaging biases MWT downward when the network rejects cars. Compare with Figure 1 for the unbiased view.

Figure 4 shows the same MWT data on the 3x3 grid plotted against offered demand rather than cars in system. The post-saturation tail of each curve appears to decrease, a well-known measurement artefact, because at high offered demand the network rejects excess cars and MWT, computed only over completed vehicles, drops as only the ‘lucky’ escapees contribute. Figure 5 shows the throughput collapse. The 95% completion threshold occurs near $N = 600$ for fixed-time and near $N = 800$ for max-pressure, Q-learning, and PGQL.

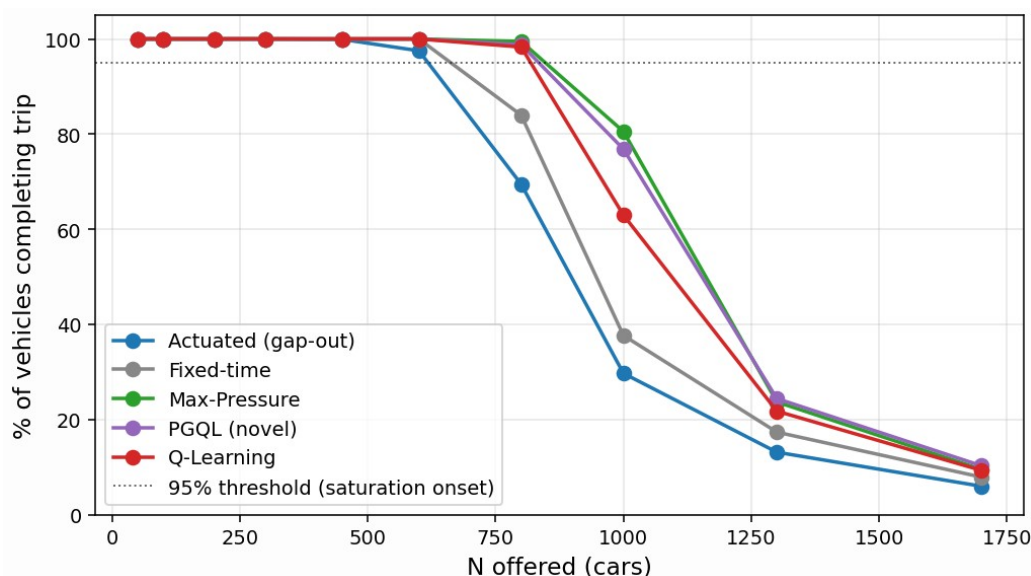


Figure 5. Throughput collapse: fraction of offered vehicles completing the trip before sim-end. The 95% threshold (dotted) marks the effective offered-demand capacity.

A counter intuitive secondary finding emerges from the moderate-load regime: gap-out actuation underperforms fixed-time at N between 450 and 800. At $N = 600$, fixed-time achieves $MWT = 60.2$ s/car while actuated achieves 144.1 s/car. This effect arises because the gap-out logic terminates greens whenever the served lanes are momentarily empty, but on a small grid the platoons are short and the cross-traffic queue is small, so the controller wastes about three seconds of yellow on each switch whose marginal benefit is negligible.

4. Discussion

The findings provide several insights into the

design and evaluation of hybrid traffic-signal controllers.

First, the comparison of five controllers across a wide demand range and two grid sizes found that max-pressure and the two learning based controllers are statistically tied across most of the pre-saturation regime on the 3x3 grid, but that PGQL outperforms max-pressure across all tested loads on the 5x5 grid. The implication is that the inductive bias provided by max-pressure is close to optimal on small networks where its one-step greedy is rarely incorrect, but its myopia appears more expensive on the larger

5x5 benchmark, leaving more room for the learner to discover beneficial deviations.

Second, PGQL was shown to match max-pressure by construction at initialisation, while the Wilcoxon analysis and 5x5 results together reveal a genuinely regime-dependent behaviour. On the 3x3 grid PGQL is significantly worse than max-pressure at low load, tied near the saturation knee, and significantly better post-saturation. On the 5x5 grid the low-load cost disappears and PGQL is uniformly best. The disappearance of the low-load penalty on the larger network suggests that the benefit of the hybrid controller depends on network scale. One possible explanation is that the 5x5 grid exposes the learner to a richer and more diverse set of traffic states, allowing the learned override policy to generalize more effectively while reducing unnecessary deviations from max-pressure. Alternatively, larger networks may present more opportunities for long-term coordination, making the one-step greedy decisions of max-pressure increasingly suboptimal and thereby increasing the value of learned overrides. Distinguishing between these mechanisms will require additional experiments, but the consistent improvement across all tested demand levels indicates that the warm-start strategy remains effective as network complexity increases. The warm-start reduces the initialisation risk relative to an unconstrained random RL policy, and the learned override delivers a measurable improvement in the regime where max-pressure's one-step rule begins to fail. For operators whose main worry is gridlock during peak periods, that trade off matches the regime where the gain matters.

Third, the carrying capacity framework introduced here, with vehicles concurrently in the network as the load axis, addresses a measurement artefact common in the literature. The single fitted parameter N_{crit} is a physically meaningful summary of network capacity, expressed in vehicles rather than in vehicles per hour.

The action-usage audit and the per-intersection breakdown together suggest a mechanism for the regime dependent improvement. The override action is selected $\sim 28\%$ of the time at low and moderate load but increases to 37% under saturation, indicating that the learner has discovered a state-dependent policy in which max-pressure's recommendation is followed when it is reliable and overridden when it is likely to be myopic. This is a more transparent mechanism than is typical of deep RL traffic controllers, where the learned policy is opaque to inspection.

The counter-intuitive finding that gap-out actuation underperforms fixed-time at moderate load deserves additional comment. Gap-out parameters are typically tuned for highway like flow with long platoons and large MIN_GREEN values. On a small dense grid this tuning is mismatched, and a single premature gap-out wastes that overhead with no compensating benefit.

A note on the operational significance of the observed gains is warranted. Figure 5 shows that PGQL and max-pressure cross the 95% completion threshold at very similar offered demand, and the post-threshold throughput collapse curves are nearly indistinguishable. The PGQL advantage reported in this paper is

therefore not a fundamental extension of the network's capacity region (PGQL does not raise N_{crit} appreciably above max-pressure in Table 1) and is not a fundamentally different congestion management capability. Rather, the contribution of PGQL is an incremental reduction in mean waiting time within the post-saturation regime, statistically significant under a paired Wilcoxon test, but bounded in magnitude (~ 5 to 12% on the configurations tested). Operators considering PGQL should weigh this incremental post-saturation improvement against the additional engineering and computational complexity of the reinforcement-learning component. The case for PGQL is strongest in deployments where (i) the network frequently operates at or beyond the capacity knee, (ii) a small post-saturation MWT reduction is operationally meaningful, and (iii) the warm-start guarantee that the trained controller is initialised at the classical baseline is itself a deployment requirement.

A final point concerns the relationship to prior reinforcement-learning studies. Several prior RL traffic-control studies (1, 2) have reported strong nominal performance but also noted instability under congestion. The PGQL framework directly addresses this concern at the deployment level: even if Q-learning convergence guarantees are violated in practice (because the environment is non-stationary, exploration is limited, or training data is sparse), the warm-started agent's behaviour is bounded near max-pressure as long as the learned overrides remain infrequent. It is a much weaker statement than asymptotic optimality, but it reduces the initialisation risk that an unconstrained random RL policy would carry.

4.1 Limitations

Several limitations should be acknowledged. First, the Q-learning and PGQL agents were trained for only 20 and 16 episodes respectively on the 3×3 grid (8 episodes each on the 5×5 grid). Although this was sufficient for the controllers to converge on a coarse low-load policy and to discover post-saturation override behaviour, the per-intersection state space contains roughly 1000 entries and many states were not visited at low training-load. A larger training budget would substantively tighten the headline claim. Second, all simulations assumed perfect vehicle detection, instantaneous signal actuation, and simplified driver behaviour. Real deployments would need to address sensor noise, network latency, pedestrian activity, weather effects, and incidents. Third, only two warm-start experts (max-pressure and gap-out actuated) were tested. Alternative experts such as Webster-optimal fixed, learned classical policies, or model predictive control would each define a different PGQL variant, and those variants are not assessed here. The max-pressure vs actuated comparison in Table 5 indicates that the choice of expert affects performance, and that controllers with capacity-region stability proofs (such as max-pressure) are the natural choice. Fourth, the largest grid evaluated was 5×5 , and only homogeneous topologies (uniform block length, single-lane, symmetric) were tested. Whether the PGQL gain transfers to asymmetric networks, multi-lane roads with turn pockets, heterogeneous block lengths, real rush hour O-D patterns, incident disruptions, or networks of 49 or more intersections has not been established by this work. Fifth, although the warm-start bounds the controller at initialisation, the formal stability proof of max-

pressure does not transfer to the trained PGQL policy once override actions become common. The post-training behaviour was checked from the simulation results. Sixth, all comparisons in this paper are against independently operating max-pressure controllers. Whether comparable performance gains could be obtained from a hierarchical or regionally coordinated max-pressure architecture (without any reinforcement learning) was not tested. The observed gains should therefore be read as improvements over independent max-pressure, not as improvements over all max-pressure-based architectures. Seventh, although a cold-start ablation was performed (Table 8), the training budget of ten episodes may not be sufficient to reveal the asymptotic gap between cold and warm PGQL. A longer training run would tighten the empirical claim about warm-start value.

5. Conclusion

This study compared five traffic-signal controllers (fixed-time, gap-out actuated, max-pressure, independent tabular Q-learning, and the proposed hybrid Pressure-Guided Q-Learning) on a simulated 3×3 SUMO grid across demand levels ranging from light traffic to gridlock, and validated the principal findings on a larger 5×5 grid. The three pressure-aware controllers reduced mean waiting time by approximately threefold relative to fixed-time on the 3×3 grid, while PGQL outperformed max-pressure at every tested load on the 5×5 grid. PGQL augments an independent Q-learner with the max-pressure recommendation as an action and warm-starts the Q-table so that the untrained agent initially behaves like max-pressure. Paired statistical analysis on the 3×3

grid revealed regime-dependent behaviour: PGQL was significantly worse than max-pressure at low load, tied near saturation, and significantly better post-saturation. The action-usage audit further showed that the override action was selected approximately 28% of the time at low load and 37% under saturation, consistent with the regime in which PGQL provides its greatest benefit.

A secondary contribution is the carrying-capacity evaluation framework. Plotting mean waiting time against the mean number of vehicles concurrently in the network avoids the post-saturation measurement artefact common in the literature and permits a clean fit to a closed-form M/M/1-style capacity model. The fitted parameter N_{crit} provides a physically meaningful summary of network capacity.

Several avenues for future work remain. The initial cold-start ablation, state-by-state decision-boundary audit, and post-override queue analysis leave important mechanistic questions unresolved. Immediate extensions include (a) longer cold-start versus warm-start training to determine whether the small post-saturation advantage persists after convergence; (b) incorporating the learned decision boundary into a deterministic enhancement of max-pressure (for example, *mp_says_switch* AND *edge_tls* $\rightarrow$ delay switch) to test whether the observed gain can be reproduced without reinforcement learning; (c) systematic comparison of additional warm-start experts; (d) regime-dependent architectures that select among multiple expert controllers rather than overriding a single expert; and (e) comparison with hierarchical or regionally coordinated max-

pressure controllers to determine whether the observed gains require reinforcement learning or can be achieved through classical coordination alone. The carrying-capacity framework introduced here provides a natural basis for evaluating such extensions.

This work presents a unified comparison of five traffic-signal controllers across two network sizes, introduces a warm-started hybrid controller that combines reinforcement learning with a stability-proven classical policy, proposes a carrying-capacity framework based on vehicles concurrently present in the network, and identifies a counterintuitive failure mode of gap-out actuation on small dense grids.

Together, these findings show that classical adaptive control and reinforcement learning can be combined so that deployment begins from a well-understood baseline while allowing learned deviations to improve performance under heavy congestion. At initialization, PGQL inherits the behavior of max-pressure; once learning modifies the policy, however, performance must be established empirically rather than through the stability guarantee of the underlying controller.

Acknowledgments

Grammatical tools including Grammarly were used in the preparation of this manuscript.

6. References

1. Wei, H., Zheng, G., Yao, H., & Li, Z. (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). ACM. <https://doi.org/10.1145/3219819.3220096>
2. Mannion, P., Duggan, J., & Howley, E. (2016). An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In *Autonomic road transport support systems* (pp. 47–66). Springer. https://doi.org/10.1007/978-3-319-25808-9_4
3. Haydari, A., and Yilmaz, Y. (2022). Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1), 11-32. [doi:10.1109/TITS.2020.3008612](https://doi.org/10.1109/TITS.2020.3008612)
4. Noaen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Abad, Z. S. H., Bazzan, A. L. C., and Far, B. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, 199, 116830. [doi:10.1016/j.eswa.2022.116830](https://doi.org/10.1016/j.eswa.2022.116830)

5. Varaiya, P. (2013). The max-pressure controller for arbitrary networks of signalised intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*, (pp. 27-66). Springer. [doi:10.1007/978-1-4614-6243-9_2](https://doi.org/10.1007/978-1-4614-6243-9_2)
6. Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., & Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4), 3414–3421. <https://doi.org/10.1609/aaai.v34i04.5759>
7. Wei, H., Chen, C., Zheng, G., Wu, K., Gayah, V., Xu, K., & Li, Z. (2019). PressLight: Learning max pressure control to coordinate traffic signals in arterial network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1290–1298). ACM. <https://doi.org/10.1145/3292500.3330949>
8. Zheng, G., Xiong, Y., Zang, X., Feng, J., Wei, H., Zhang, H., Li, Y., Xu, K., & Li, Z. (2019). Learning phase competition for traffic signal control. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1963–1972). ACM. <https://doi.org/10.1145/3357384.3357902>
9. Watkins, C. J. C. H., and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4), 279-292. [doi:10.1007/BF00992698](https://doi.org/10.1007/BF00992698)
10. Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011). SUMO, simulation of urban mobility: An overview. In *Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011)* (pp. 63–68). IARIA. ISBN 978-1-61208-169-4
11. Webster, F. V. (1958). *Traffic signal settings* (Road Research Technical Paper No. 39). Her Majesty's Stationery Office.

Appendix

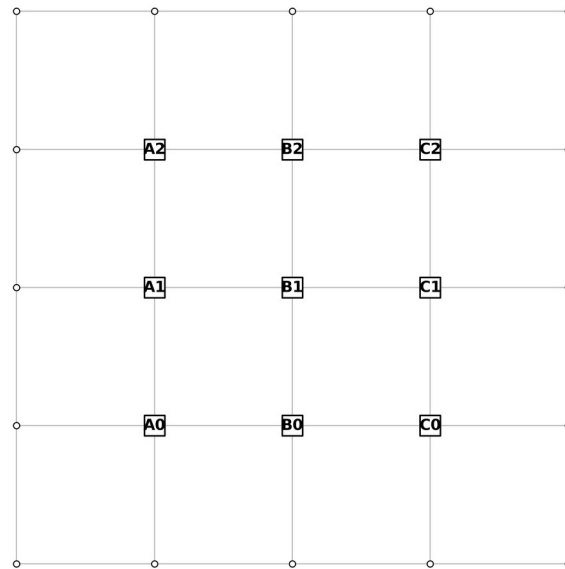


Figure A1. 3X3 grid identification of intersections

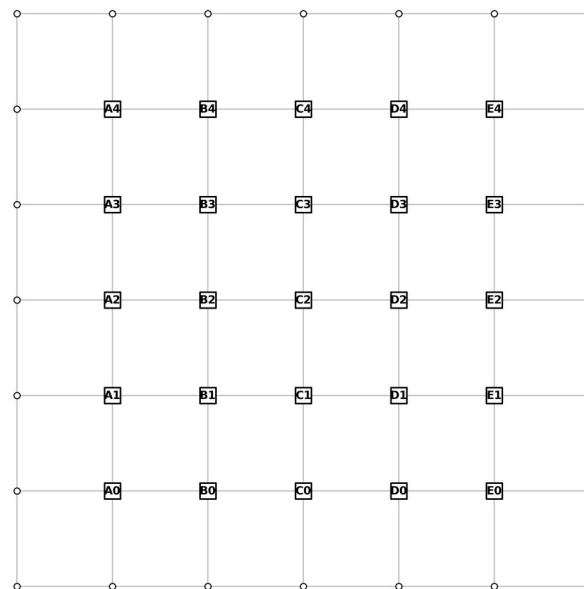


Figure A2. 5X5 grid identification of intersections