Using Reinforcement Learning to Combine Green Light Optimised Speed Advisory and Responsive Traffic Control Systems with Non-Autonomous Vehicles: Effects of Imperfect Training and Incomplete Information

William Paine

4th June 2025

https://orcid.org/0000-0003-4894-9680

Abstract

The advantages of Green Light Optimal Speed Advisory (GLOSA) include improvements in travel times, fuel consumption, battery life, emissions, the number of stops, and ride smoothness for cars, buses, HGVs, HEVs, and EVs in scenarios ranging from isolated junctions to connected networks.

However, GLOSA is a detriment when paired with responsive traffic control systems (RTCs), which are presently widespread, greatly reducing the number of locations where the advantages of GLOSA could be obtained. This is because RTCs cannot provide accurate long-term future signal plans without the introduction of some mitigating solution.

However, as Reinforcement Learning methods can, by trial and error, identify unseen solutions to complicated problems, there appears to be no reason they couldn't be used to create frameworks designed to combine GLOSA and RTCs and benefit (Connected Non-Autonomous Vehicles) CNAVs, allowing the advantages of GLOSA to be obtained with the present traffic and traffic control make-ups.

Therefore, this thesis investigates the performance of such frameworks in both ideal and non-ideal conditions. To achieve this, an initial framework was constructed and evaluated on a simulated isolated junction. It was found that the initial framework had a positive impact on stopping time and junction entry speed when traffic densities were at 55% of saturation levels but a negative impact at traffic densities of 70% as the Reinforcement Learning Traffic Control (RLTC) system was unable to reach states where vehicles could approach the junction without the obstruction of queues regularly enough. Also, in the 55% scenario, when the training penetration rate (TPR) was lower than 20%, performance was degraded if the evaluation penetration rate (EPR) was increased, as vehicle behaviour differed greatly from what the models were expecting.

To expand testing to arterial flows, a revised framework was created and tested on a simulated arterial testbed. Alterations were also made to the framework to account for lessons learned from the isolated junction experiments.

It was found that the revised framework had a positive impact at all traffic densities, with vehicle speed increasing and stops and waiting time decreasing compared to benchmark systems, even when performance was degraded by the absence of vehicle route information. However, performance was still degraded at higher EPRs when the TPR was lower than 40%.

Overall, this research shows that GLOSA can be combined with Reinforcement Learning Traffic Control (RLTC) systems in a way that benefits CNAVs in terms of waiting time, number of stops, junction entry speed and average speeds, and demonstrates that GLOSA could be installed despite present traffic and traffic control make-ups. It also highlights the need for future frameworks to be tested at a variety of EPRs and with/without route information to avoid decreases in performance or outright failures as conditions at the site of deployments change over time.

Contents

List of Figures

List of Tables

Abbreviations

A2C	-	(Synchronous) Advantage Actor-Critic Implementation
A3C	-	Asynchronous Advantage Actor Critic
CAV	-	Connected Autonomous Vehicle
CNAV	-	Connected (or GLOSA equipped) Non-Autonomous Vehicle
DQN	-	Deep Q Networks
DTSE	-	Discrete Traffic State Encoding
EPR	-	Evaluation Penetration Rate
FTTCs	-	Fixed Time Traffic Control systems
GLOSA	-	Green Light Optimal Speed Advisory
PER	-	Prioritised Experience Replay
PPO	-	Proximal Policy Optimisation
PRC	-	Practical Reserve Capacity
RLTC	-	Reinforcement Learning Traffic Control
RTCs	-	Responsive Traffic Control system(s)
RPS	-	Raw Pixel Snapshots
TPR	-	Training Penetration Rate
TSCs	-	Traffic Signal Control system(s)

1 Introduction

The efficient management of urban traffic has become critical in the modern age. A common infrastructure-based solution for traffic management is Responsive Traffic Control systems (RTCs) which can reduce travel time, delays, stops [1; 2; 3], and stopped times, as well as increase average speeds; reduce emissions [4]; and increase safety [5] by using live traffic information to estimate incoming traffic flows and frequently [6; 7] calculate near-optimal future signal plans that maximise or minimise some objective function [8].

There also exists a vehicle-based solution, called Green Light Optimal Speed Advisory (GLOSA), that informs vehicles approaching a signalised junction or pedestrian crossing of future signal timings and advises drivers of the optimal speed at which they should approach the intersection to arrive at a green light [9]. This achieves benefits like reduced stopping time, reduced travel time [10], improved fuel consumption or energy efficiency, less stops, better ride smoothness [11], and reduced CO_2, CO, NO_x, and particulate emissions [12] for all vehicles, including public transport [11], heavy-duty diesel trucks [10], and electric vehicles [13].

However, as GLOSA requires accurate future signal plans, it appears fundamentally incompatible with RTCs [14]. Furthermore, due to the lack of compatible vehicles and the comparatively higher implementation costs, GLOSA has been sidelined in favour of RTCs, which are currently ubiquitous. Accordingly, the advantages of GLOSA will remain unavailable to road users until such time as a joint control framework, which enables GLOSA to be compatible with RTCs, can be deployed.

Existing joint control frameworks all fall into one of three categories: Phase Prediction Algorithms, Junction Controlled Vehicles, and Reinforcement Learning Traffic Control Systems. However, each of these presently has limitations.

1.1 Phase Prediction

Phase Prediction Algorithms use methods like Markov chains [15], using transition graphs (see figure 1), or reinforcement learning [16] to make predictions about the future signal plans of oncoming junctions. It has been shown that phase change algorithms can make predictions accurate to within two seconds on average using loop detector data [16] or data recorded through mobile phones [15]. These predictions can then be used to activate GLOSA. However, with an average error of 2 seconds, some vehicles will be forced to make unnecessary stops, while others may run red lights if the driver is not paying attention because the green phase was later than expected.

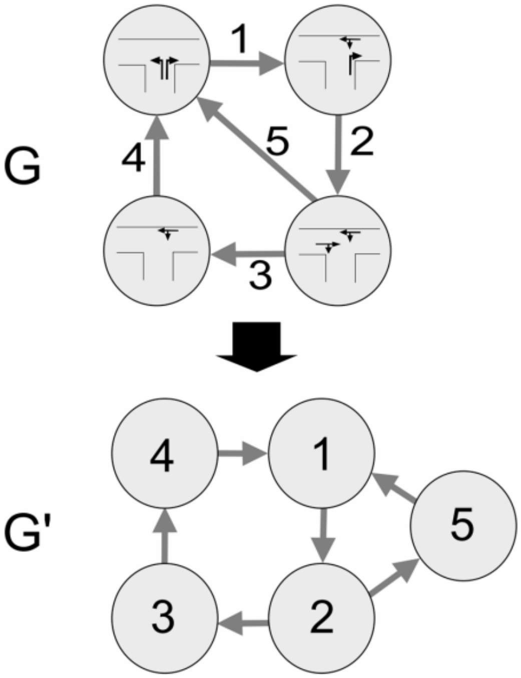

Figure 1: Example of a state graph of the traffic light controller (G) and the corresponding transition graph focusing on signal changes and their occurrence probability (G') to be used for Markov chain based traffic phase prediction [15].

1.2 Junction Controlled Vehicles

Junction Controlled Vehicles strategies involve the infrastructure at the junction issuing trajectories to Connected Autonomous Vehicles (CAVs), either using a time slot-based approach or machine learning. In CAV-only environments, research has shown that vehicles approaching the junction can be assigned a time slot in which they must traverse the junction, allowing them to share the junction with and pass between other vehicles that are coming from conflicting directions (see figure 2) [17]. These slot-based intersections are extremely efficient, achieving better throughput, with fewer stops and less delay using these systems compared to a traditional traffic management method [18; 19; 20; 21; 22] as it removes the need for a red phase. However, such systems could be dangerous if CAVs are not properly maintained, the infrastructure fails, or there are communication errors between the CAVs and the infrastructure. Also, such systems are not compatible with Connected Non-Autonomous Vehicles (CNAVs) or other vehicles, as drivers would be incapable of driving at a suitable precision. This is an issue as they will likely make up the vast majority of vehicles on the road for the foreseeable future [23].

However, some ideas from slot-based intersections have also been applied in mixed traffic scenarios. This has chiefly been achieved by implementing a CAV passing rule into otherwise standard RTCs such that CAVs can group up on approach to the junction and then be served by slot-based intersection style phases, while the remainder of vehicles are served by traditional phases [24]. However, as with slot-based intersections, these junctions could be dangerous when the CAV passing rule is active. Also, while this kind of solution is compatible with CNAVs and traditional vehicles, they are not directly advantaged by the inclusion of the CAV passing rule.

In other Junction Controlled Vehicles strategies, CAVs are given precise commands, in terms of acceleration or deceleration values, by the junction using machine learning [25]. As the junction operates with RTCs, this method can be deployed in mixed traffic flows. However, it does not benefit CNAVs or other traditional vehicles directly.

Overall, Junction Controlled Vehicles strategies are not deployable in the present day or in the foreseeable future because they do not offer any benefits to CNAVs or other traditional vehicles.

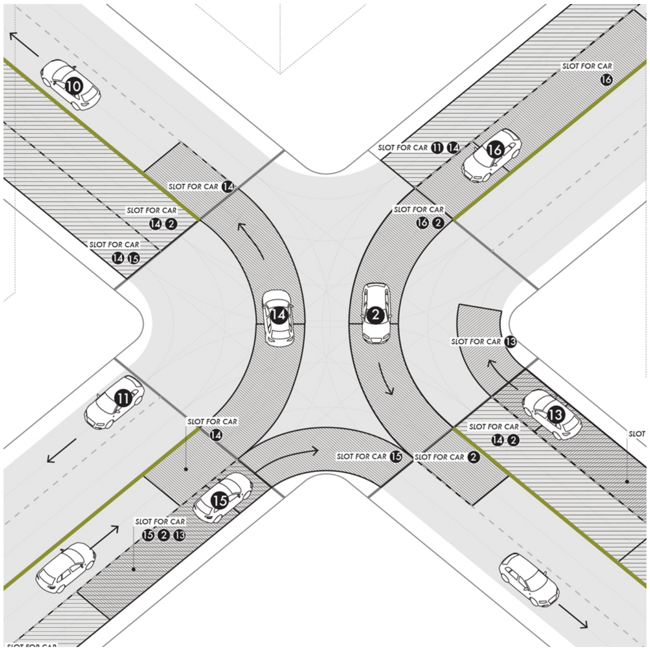

Figure 2: Example of a slot-based intersection managing CAVs to allow vehicles from conflicting directions to traverse the junction simultaneously[26].

1.3 Reinforcement Learning Traffic Control Systems

Reinforcement Learning Traffic Control (RLTC) systems, where an agent trained by reinforcement learning (see figure 3) controls the junctions, have also been used in frameworks that aim to grant GLOSA-like advantages to CAVs [27] either by adding junction controlled vehicle functionality, either issuing discrete or continuous commands [25][28], or by implementing a form of CAV compatible GLOSA [29].

In these existing frameworks, CNAVs and other traditional vehicles can be accommodated [28]; however, as with Junction Controlled Vehicles strategies, such vehicles receive no benefits, making these frameworks unworkable in the present day or in the foreseeable future.

However, in existing literature, no reason has been found that would bar RLTC systems from being deployed with GLOSA and GLOSA compatible CNAVs, creating a joint control framework which can benefit present traffic flows, as vehicles would only need to be connected while placing no requirements on their autonomy. Therefore, the performance of such a system is an open question.

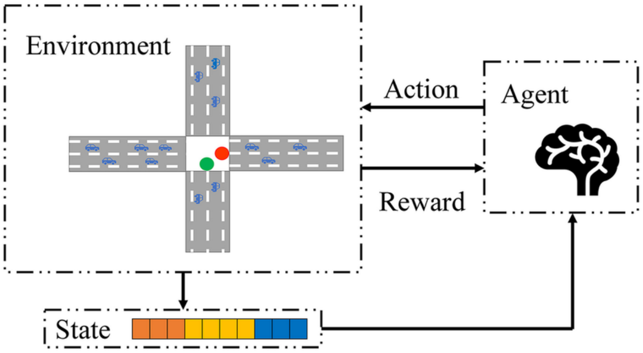

Figure 3: Example of a typical Reinforcement learning structure in traffic signal control system [30].

1.4 Problem, Aim and Objectives

While each of the existing types of joint control frameworks has limitations, RLTC systems appear to have the greatest potential for adaption that could lead to implementation in the near future. However, further research is needed, and therefore, the aim of this thesis is to understand the potential for using reinforcement learning to combine RTCs and GLOSA in CNAV-dominated traffic scenarios. To achieve this aim, several objectives will be satisfied:

- The first objective of this thesis is to understand the existing approaches and their limitations by fully reviewing the state of the art in GLOSA, RTCs, and existing combinations of the two.
- The second objective is to design a framework that incorporates RTCs and GLOSA for

CNAV-dominated traffic scenarios.

- The third objective is to understand the potential performance of the framework in a range of scenarios, by performing a series of experiments.
- The fourth objective is to understand the impact of operation in imperfect conditions, e.g. when there are differences in penetration rates of GLOSA when training (as in the Training Penetration Rate or TPR) and when deployed (as in the Evaluation Penetration Rate or EPR) or when vehicles do not provide route information to the framework, by performing further experiments.
- The fifth objective is to understand the limitations of this research and the direction future research should take.

It should be noted here that the scope of this work is limited to simulation tests modelling the real world instead of empirical data. It, therefore, relies on established models of GLOSA and car-following operations and excludes pedestrians and vehicles such as HGVs, emergency vehicles, and public transport.

The remainder of this thesis shall be arranged as such:

- Chapter 2 will comprise a literature review covering traffic control, GLOSA, and combinations of both, as well as existing phase prediction algorithms, junction-controlled vehicle frameworks, and RLTCs. Over the course of this review, the reader will be given a grounding in all the topics needed to understand this research, and several gaps in knowledge will be defined.
- After that, Chapter 3 will contain a Methodology which will describe how the aims and objectives of this thesis will be achieved, and how the gaps in knowledge will be addressed.
- In chapter 4, a RLTC based joint control framework and isolated junction testbed will be designed with the aim of gathering experimental data relating to the potential performance of the framework in a range of scenarios.
- In chapter 5, the RLTC based joint control framework will be evaluated in isolated junction experiments and the results discussed.
- In chapter 6, an arterial testbed will be designed. Also, the RLTC framework will be updated, both to allow for operation on an arterial flow and to address issues uncovered in the isolated junction experiments.
- In chapter 7, the RLTC based joint control framework will be evaluated in arterial flow experiments and the results discussed.

- Finally in chapter 8, the results, limitations, and conclusions of this thesis will be reviewed and used to make recommendations for future work in this area of research.

2 Literature Review

To achieve the first objective laid out in the introduction, a wide-ranging literature review must be undertaken.
Firstly, existing GLOSA implementations must be reviewed to gain an understanding of their requirements for successful deployment. As such, their workings will be detailed and their benefits and weaknesses quantified. In particular, the performance of GLOSA in scenarios where accurate long term future signal plans are unavailable will be examined to ascertain what benefits are lost in such situations.

Secondly, existing TSCs (Traffic Signal Control systems) must be reviewed to gain an understanding of the relationship between the flexibility of the control schemes (how often, suddenly, and drastically signal plans change) and the benefits they provide. A particular focus will be placed on those systems which are in current widespread use.

Thirdly, previous joint control frameworks and other related topics, including Phase Prediction algorithms, Junction Controlled Vehicle methods, and CAVs, must be reviewed to understand their potential performance but also their limitations. The focus will be on the results that have been achieved by existing frameworks as well as the issues faced by such solutions that presently make them impractical to deploy.

fourthly, RLTC systems must be reviewed to gain an understanding of their workings, potential performance, current limitations, and the research avenues for development. This will include both a wider look at RLTC systems in the latter part, as well as reviewing their use alongside GLOSA in the former part.

Finally, at the end of this section, several gaps in knowledge will be outlined.

2.1 GLOSA

GLOSA systems use Infrastructure to vehicle communication to inform vehicles approaching signalised junctions or pedestrian crossings of future signal timings. This information can then be used to reach the junction during a green phase and achieve a more efficient speed profile over the course of the journey.

Early GLOSA systems were typically implemented with roadside message signs placed ahead of the signals. However, due to the significant costs and maintenance issues, only a small number of these GLOSA systems were installed worldwide [31]. More modern GLOSA implementations typically rely on communication through technologies such as 4G-LTE and the DSRC via 802.11p protocol [9]. However, this means that their performance can be negatively affected by high rates of packet loss [32].

The information sent to vehicles during the implementation of GLOSA can consist of geometric information of one or more intersections and the entire programming of the traffic lights (times and manoeuvring directions attached) and their current phases[9].

The simplest GLOSA implementations for Fixed Time Traffic Control systems (FTTCs) routinely calculate a target speed U_t using an assumption that acceleration will be continuous, such that,

$$U_t = \frac{2d}{T_T L} - U_0, \tag{1}$$

where U_0 is the current speed, d is the distance to stop line, and T_{TL} is the time until the light turns green [33].

However, while calculating a target speed will allow vehicles to reach the junction while the lights are green, it may not provide them with the most optimal speed profile. Therefore, some GLOSA research has focused on searching for the optimised speed curve [34; 35].

An example of a common approach for finding optimal speed curves is using a genetic algorithm where a number of candidate options are generated and then assessed according to a fitness function linked to fuel economy, travel time, etc. [36]. After this, the most successful options are crossed and mutated to produce offspring solutions, which should describe better optimised advisory speed profiles [36].

This can be combined with a car following model to improve the results in multi-vehicle situations [36]. On simulated isolated intersections, this kind of speed curve optimisation can further decrease fuel consumption and trip time compared to traditional GLOSA implementations [36].

GLOSA can also be used to optimise the advised speed profile for several or all junctions along the proposed route. This is called multi-segmented GLOSA (see figure 4) [37]. As the search space of all target speeds for each segment through the arterial is exceptionally large, searching exhaustively is not considered optimal [37]. Therefore, it is often done using a genetic algorithm [37]. With multi-segmented GLOSA, significant differences in advised speeds or speed profiles can be avoided, reducing inefficient acceleration and deceleration between junctions [37].

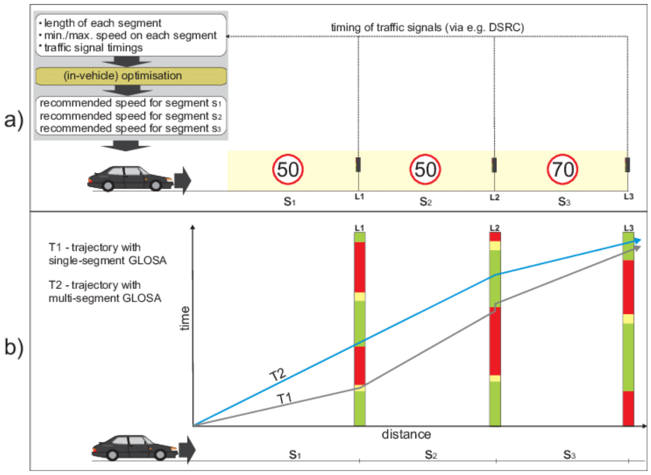
a)
• length of each segment
• min./max. speed on each segment
• traffic signal timings
(in-vehicle) optimisation
recommended speed for segment s1
recommended speed for segment s2
recommended speed for segment s3
timing of traffic signals (via e.g. DSRC)
50
50
70
S1
L1
S2
L2
S3
L3
b)
T1 - trajectory with single-segment GLOSA
T2 - trajectory with multi-segment GLOSA
time
T2
T1
distance
L1
L2
L3
S1
S2
S3

Figure 4: An example of a GLOSA speed profile for both single segment and multi segment systems [37].

2.1.1 Overview of Benefits of GLOSA

GLOSA has also been shown to have a positive impact on the experience of public transport where, on simulated arterials (with 2 to 8 junctions), fuel consumption, number of stops and ride smoothness were improved [11].

When applied to heavy-duty diesel trucks, GLOSA has been shown to increase the passing rate of vehicles and lead to a smoother driving style with increased cruise time and reduced travel time [10]. Aggressive acceleration and deceleration behaviour near signalised intersections are also significantly reduced, as well as fuel consumption and CO_2 emissions [10].

Plug-in hybrid vehicles combined with GLOSA see a significantly reduced frequency of engine starts and throttle openings by reducing the number of stops and starts and the intensity of acceleration [12]. Other benefits include reduced energy consumption and pollutant emissions, including a reduction in CO_2, CO, NO_x and particulate emissions [12].

Similar benefits can be seen on all-electric vehicles, with reductions seen in energy consumption, if the speed profile of the route is designed to improve the efficiency of the driving motor and regenerative braking [13].

A decrease in rear-end collisions has also been observed, as well as an improvement in pedestrian safety [38].

The impacts of GLOSA on fuel and traffic efficiency when using this algorithm have been shown to be positive for a simulated 100m test bed of Guildford town centre, in the United Kingdom, with a single route arterial flow with two junctions [33]. Testing determined the effects of different penetration rates of GLOSA-equipped vehicles, activation distances and traffic densities. It was found that the higher the GLOSA penetration rate is, the more benefits are observed [33]. Also, as the density decreases, the benefits for fuel efficiency are reduced, while the benefits to traffic efficiency are increased [33]. Lastly, the optimal activation distance, where the GLOSA application should begin advising the driver of the optimal speed profile, is near 300m from the traffic lights, but it depends slightly on the road network [33].

GLOSA has also been discussed as an alternative to re-timing FTTCs, an expensive, time-consuming, and frequently required process where the cycle plan of a FTTCs is updated to account for long-term changes in typical traffic patterns that must be undertaken to keep a junction running efficiently [39].

However, it was found that, while GLOSA and similar strategies may sometimes bring more benefits than re-timing traffic signals, the signal retiming process remains an irreplaceable part of the overall improvement of arterial traffic operations and with or without GLOSA, re-timing of the signals brings significant benefits [39].

It was also found that GLOSA brings additional benefits for fixed time signals which are recently re-timed or optimised [39].

While GLOSA typically provides benefits for equipped vehicles, this can come at the expense of any unequipped vehicles. One study examined the effects of GLOSA on delay, capacity, and surrogate safety measures for vehicles that arrive from unsignalised side driveways and access roads [40]. On a Vissim simulation model of a five-intersection corridor, GLOSA was found to often have a significant effect on the delay of the side-street traffic [40]. However, in general, it only had a minor impact on the number of conflicts [40]. However, the road geometry and proximity of the signalised intersections affected the impact of GLOSA on the side-street traffic [40].

As stated earlier, GLOSA becomes less effective as traffic density increases. This is caused in part by the formation of queues at junctions, which, if not considered, will block an equipped vehicle from progressing in accordance with the speed advice. To address this, a vehicle queue length estimation method, based on vehicle to infrastructure communication technology, has been proposed to predict the effective green light time [41]. When compared to a traditional GLOSA system (without queue length estimation), the improved GLOSA system was shown to reduce energy consumption in simulation and real-vehicle tests [41].

Other researchers have demonstrated that accounting for downstream traffic also delivers promising improvements in terms of capacity, the number of stops, queue lengths, delay, and travel time [42].

The issue of other vehicles obstructing the advised path can also be addressed by using a reinforcement learning-based approach to predict downstream traffic [43].

Another variable of GLOSA performance is the Red to Green Time (RGT) ratio of the traffic lights and the type of assistance offered. When the RGT ratio is almost 1 (red time and green time are almost equal), a partial assistance mode where GLOSA is only active during the red signal is recommended [44]. However, when the RGT ratio is less than 1 (more green time than red time), a full assistance mode, which assists the vehicle regardless of the signal phase, performed better [44].

Strategies like GLOSA have also been applied to foot traffic by using a smartphone app to recommend walking speeds and routes to pedestrians [45]. It can not only provide pedestrians with dependable speed recommendations but also the optimal route to follow [45]. It was found that pedestrians using the app made better time than naive pedestrians (pedestrians not using navigation) and route planning based on the earliest available green time for pedestrians [45].

However, when performing field tests of GLOSA, it often underperforms compared to simulation results. It still shows an improvement in fuel consumption versus the scenario without GLOSA. However, factors like the driver's response times and inability to keep the vehicle at a precise speed reduce GLOSA's effectiveness [46]. Therefore, some research has focused on finding the best methods for communicating speed advice to drivers, with many different ideas and solutions being trialled [46].

Several factors have been determined to affect a driver's response to speed advice, including the activation distance, the difference between the advised speed and the instantaneous speed of the vehicle, and whether the driver was accelerating when the speed advice was delivered [47].

Drivers typically respond slower, and less often, to speed advice if the activation distance is increased [47]. Simulations also show that a short activation distance with a long cycle length will significantly restrict the positive impacts of GLOSA [47]. However, with longer activation distances, the speed compliance of drivers who did respond to GLOSA was improved, and their deceleration was smoother [47]. This suggests that the activation distance should be neither too close to nor too far from the stop-line [47].

When there was a high difference between the instantaneous and the advised speed, the chances of drivers showing a response or adhering to the speed instructions decreased [47]. Consequently, even if they show a response, they would be forced to come to a stop at the intersection [47]. Because of this, the researcher recommended that the speed of the driver should be considered before offering speed advice and in cases where the difference was higher than $20km/h$, speed advice should not be provided, and instead, the driver should only be provided with an instruction to come to a gradual stop to avoid hard braking near the intersection [47].

Lastly, drivers who were accelerating at the instant when the speed instruction was provided were more likely to show a response in terms of speed reduction. However, the response times were higher in such cases [47]. The researchers also stated that factors such as time of the day and the number of lanes available also influenced the response of drivers [47].

One algorithm called driver-centric GLOSA (DC-GLOSA) was proposed where drivers would be advised of the degree of acceleration or braking [48]. It was field tested at the Pusan National University campus, where it reduced intersection stopping time [48]. The researchers also argued it had a positive effect on fuel efficiency, driving comfort, and driver focus [48].

Another algorithm called GLOSA-RMM provided drivers with screen and voice instructions to minimise the driver's distraction [49]. In field tests, it was able to reduce stopping time and fuel consumption [49].

The French public-funded Co-Drive project, which focused on V2X technologies with the aim of reducing emissions and improving traffic flow, performed a GLOSA field test on a circular test track with two sets of traffic lights where drivers were presented with speed advisory data in the form of a speedometer with green and red zones dynamically marked [50]. They found that when GLOSA was activated, stopping time, emissions, and travel time were decreased [50]. Furthermore, GLOSA had a slightly greater positive effect on emissions when the speed limit was increased from 50km/h to 70km/h [50].

Another strategy that has been trialled for conveying speed advice to drivers is using a heads-up display to project a green wave onto the road ahead [51]. So long as the vehicle remains on the green wave, the vehicle can clear the junction without stopping [51]. Such a system would have safety benefits compared to using dashboard displays as the driver would not have to take their eyes off the road [51]. It also has a positive effect on fuel economy and emissions whilst not negatively affecting travel times and waiting times [51].

This work was expanded on to confirm the safety benefits with a simulator study where an increase in the time headway of the traffic flow was observed as well as a decrease in the severity of deceleration [52; 53]. It was therefore concluded that the proposed GLOSA system achieved safer traffic flows in the simulated real-world signalised intersection without deteriorating the traffic flow efficiency [52; 53].

Several of these methods of conveying speed advice to drivers have been the subject of a comparative real-world study in which drivers were shown either a speed range on the speedometer, a target speed, or a graphic interface. In some tests, this was accompanied by a voiceover [54].

Due to the time taken for drivers to react and alter their speed, it was determined that voice prompts were most efficient when they were given the advice 3.1 seconds in advance of when the advice needed to be adhered too. Also, no matter which dashboard display mode was used, the actual driving speed of the driver was often higher than the recommended speed [54]. Therefore, advising a lower speed than necessary is recommended [54].

It was also concluded that voice prompts were more important than the dashboard display as the driver receives the information more directly from the voice prompt and does not need to focus on the dashboard, improving driving safety [54]. Among the three kinds of visual displays assessed, the recommended method gave users a speed range on the speedometer, as it required the least time for the driver to absorb the information [54]. The graphical display method was more conducive to reducing the driving speed error but came at a much higher visual and mental burden to the driver [54].

Aside from the issue of distracting the driver with speed advice, GLOSA also comes with another safety concern. When an equipped car is advised to slow down to avoid a red light, other drivers in unequipped vehicles might become frustrated, leading to a road rage incident, or a potentially dangerous attempt to overtake the equipped vehicle [55].

One simulator study attempted to combat these issues by informing drivers, of unequipped vehicles, of the equipped nature of the car in front [55]. However, they found that this had no effect on the level of frustration felt by drivers in unequipped vehicles [55]. Also, while many drivers adapted their driving behaviour in accordance with the assisted driver, they also tended to drive much closer to the equipped vehicle when it slowed down approaching green lights [55]. It was thought that drivers in unequipped vehicles might have associated GLOSA with a higher degree of safety and thus felt safer in riskier positions [55].

In a large-scale simulation study, including a network comprising eight vertical and thirty-eight horizontal roads with a speed limit of 50 km/h covering an area of about 5.6 km by 1.6 km, the impact of different equipment rates of both traffic lights and vehicles was studied [56].

At low traffic densities, these systems can reduce CO_2 emissions, fuel consumption, waiting times, and the number of stops. It was also found that these benefits grow linearly with the number of equipped traffic lights or vehicles (or both) [56].

However, with denser traffic, the performance of such a system deteriorates as vehicles affected by other vehicles can no longer choose an optimal driving strategy [56]. In simulations, drivers of unequipped vehicles cause queues at the junctions [56]. The resulting congestion leads to higher CO_2 emissions for unequipped vehicles and increased waiting times, travel times, and stops for all vehicles [56].

The impacts of multi-segment GLOSA have been evaluated in simulations of between 3 and 15 junctions (although without cross traffic at those junctions), and it was found that, as long as traffic conditions allow drivers to select a wide range of speed (e.g. during free-flow driving), multi-segment GLOSA results in much better performance when compared with single-segment GLOSA in the metrics of travelling time and fuel efficiency [31].

Multi-segment GLOSA has also been shown to have a positive impact on the experience of public transport where, on simulated arterials (with 2 to 8 junctions), fuel consumption, number of stops, and ride smoothness were improved compared to single-segment methods [11].

Another method for finding the optimal speed curve for multi-segmented GLOSA is to discretise the search space and use the branch and bound algorithm [57]. Using this method, a discretised range of target speeds is created [57]. These speeds are then evaluated to see if they would allow the vehicle to reach the following junction during a green phase [57]. Speeds that would allow for arrival during green time are retained, while the rest are discarded as they are not optimal [57]. The process is then repeated for each following junction, with speed profiles only retained when they arrive at green phases at each junction [57].

When a collection of near-optimal solutions has been created they can be run through an objective function to allow for the choosing of the optimal solution [57]. In field tests conducted on a three-junction arterial, a reduction in fuel consumption, trip times and stop times were observed when the system was activated [57].

Another algorithm called Dynamic-GLOSA attempted to solve the multi-segment GLOSA optimisation problem by choosing speeds to minimise an objective function and then, if the vehicle would arrive on red at any junction, used that information to greatly reduce the search space before repeating the search [58]. It was compared to brute force optimisation, genetic algorithms, maximum speed GLOSA and naïve drivers [58]. Dynamic-GLOSA performed almost as well as the brute force approach, but with a much shorter calculation time, and outperformed the other existing methods for every number of segments while being much faster to calculate the optimal speed than the genetic algorithm [58].

Similar work has been done to calculate optimal speed profiles for hybrid electric vehicles (HEVs) by altering the fitness function according to fuel consumption models designed for HEVs [59]. When implemented on a three-junction arterial, there are significant advantages in reducing fuel consumption and intersection passing time compared to a single segment algorithm [59].

Multi-Segment GLOSA has also been evaluated on simulated non-arterial networks. In one study, thirteen signal-controlled junctions in the city centre of Trento, Italy, were modelled [9]. In this scenario, GLOSA achieved reduced fuel consumption and emissions when the activation distance was above 300m [9]. At an activation distance of 250m, fuel consumption and CO_2 emissions were reduced, but CO and particulate emissions were increased [9]. When the activation distance was decreased further, GLOSA caused an increase in emissions and fuel consumption [9]. It was also observed that a lower minimum advised speed led to decreased emissions and fuel consumption [9].

Multi-segment GLOSA has also been explored on buses. A Green Light Optimal Speed Advisory (GLOSA) system for buses, called B-GLOSA, was developed, which was implemented on diesel buses and field tested to validate and quantify the potential real-world benefits [60]. A moving-horizon dynamic programming problem solved using the A-star algorithm was used to compute the energy-optimised vehicle trajectory through signalised intersections [60]. The proposed B-GLOSA system was able to smooth the bus's trajectory while traversing signalised intersections and simultaneously reduce fuel consumption and travel times compared to an unequipped bus [60].

In heavy traffic or on single-lane roads, vehicles whose paths diverge after the next intersection, and therefore have slower fuel-optimal speeds, may block the other faster vehicles from meeting their optimal speed [61]. This issue can be addressed by performing a negotiation where all vehicles involved agree upon a compromise speed [61]. The resulting cooperative speed advice schemes reduced the total fuel consumption of the involved vehicles, approximating the global optimum [61]. It was shown that such cooperative schemes reduced fuel consumption compared to other GLOSA schemes [61].

2.1.2 Inaccurate or Insufficient Future Signal Plans

Based on the literature reviewed in the previous section, GLOSA offers a wide range of advantages for vehicles navigating FTTC junctions. However, as GLOSA works by informing vehicles of the timings of traffic signals on their route, they face a complication when they are installed alongside traffic control systems which supply inaccurate or insufficient future signal plans [14]. When future signal plans are inaccurate or insufficient, GLOSA has suboptimal performance regarding CO_2 emissions and fuel usage [62]. While some algorithms manage to reduce the time vehicles are standing for, total travel time typically increases [62]. Also, results have pointed to further negative impacts for diesel vehicles compared to petrol vehicles on the measures of emissions and fuel consumption in this scenario [62].

Using such systems with GLOSA, without modification, often performs worse than just using either of the systems [14]. This result has been shown to hold for a range of GLOSA algorithms [62]. In conclusion, because they cannot be used together, and because RTCs are cheaper than GLOSA to install and do not require connected vehicles, RTCs are ubiquitous and GLOSA has been sidelined. Accordingly, the benefits of GLOSA will remain unavailable until the issues of inaccurate or insufficient future signal plans can be addressed.

2.2 Traffic Signal Control Systems

Clearly the choice of TSCs used on a junction or in a network influences the performance of any GLOSA implementation and therefore it is required for this thesis to review the existing TSCs. In search of ways to, safely and efficiently, accommodate ever-greater volumes of road traffic, researchers have spent decades developing countless TSCs and other similar strategies. This section is split into two parts. The first part will cover FTTCs, and the programs used to optimise their timings before deployment. The second part will cover widely deployed RTCs.

For a grounding in traffic signal control and terms specific to this field, please see "A Concise Introduction to Traffic Engineering: Theoretical Fundamentals and Case Studies" [63].

2.2.1 Fixed Time Traffic Control systems

Fixed Time Traffic Control systems operate signal-controlled junctions according to a repeating preset signal plan. This plan includes: a cycle time, the total amount of time required to complete all stages/phases at a junction; and split, the green time allocated to a phase; and offset times, the delay between green times of subsequent phases, for each phase.

These systems can be deployed as isolated units or optimised in groups over a network. In the case of network installations, the plans throughout the network can be coordinated to create green waves, where traffic lights along an arterial turn green in succession, allowing a platoon of vehicles to navigate the network without stopping [64].

Fixed Time Traffic Control systems may also be programmed with multiple signal plans that are activated at certain times of day, on certain days of the week, or when special events are scheduled. However, fixed-time plans cannot respond automatically to traffic accidents, roadworks, or moment to moment changes in live traffic conditions [64]. Over time, the performance of fixed-time plans tends to degrade as traffic patterns change, and therefore, it is imperative that the signal plans are kept up to date [65].

Examples of software packages used for calculating fixed time signal plans include: MAXBAND [66], TRANSYT [67], LinSig [68], and Synchro [69].

2.2.1.1 MAXBAND

MAXBAND is a portable, offline computer program for calculating near optimal fixed time signal plans [66]. MAXBAND tries to maximise the weighted sum of the bandwidths in both directions along an arterial flow [70]. To maximise this metric, MAXBAND selects cycle times, offsets, and order of left-turn phases along the arterials [71]. Inputs for MAXBAND include network geometry, green splits or traffic flows and capacities, left turn signal patterns, a queue clearance time, and the range of vehicle speeds on each link [72].

The user may specify at any intersection in either direction, a queue clearance time which MAXBAND uses to adjust the signal times so that the queues, formed by turning movements onto the artery at previous intersections, clear and do not impede the flow of vehicles through the arterial [72].

The first version of MAXBAND could manage problems on networks with only three arteries and up to seventeen traffic signals [73]

Later, the MAXBAND model was extended to create MULTIBAND [71] which can also optimise splits [74]. MULTIBAND increased the flexibility of green waves on arterial flows by allowing them to use more green time at junctions where it was available [75]. MULTIBAND produced improvements in several performance metrics (delay, stops, speed and miles per gallon) compared to MAXBAND [71].

More recently, an updated version of MULTIBAND called AM-BAND was proposed [73]. AM-BAND further increased the flexibility of green waves on arterial flows by removing a symmetry requirement of the green wave [76]. MULTIBAND and AM-BAND both support optimising signals for public transport, with the option to apply different weights/coefficients to buses (as opposed to other traffic) in the objective function [77].

2.2.1.2 TRANSYT

TRANSYT is a software suite used for optimising fixed traffic signal timings at single junctions and large traffic networks of mixed control (signalised or priority) [67]. It consists of a traffic model and a signal optimiser [78]. In the traffic model, TRANSYT simulates the movement of traffic through the network and then calculates a weighted sum of estimated delays and stops on all the streets, which will then be used as a performance index by the signal optimiser [79; 78].

The signal optimiser adjusts the signal timings before the traffic model is rerun with the updated signal timings [78]. If the updated signal timings lead to an improved performance index, they are adopted, or else they are rejected [78]. The process is then repeated until ”optimal timings” have been found [78]. This method, which is characterised by trial-and-error, is called a hill-climbing technique [78; 79].

Splits and offsets are optimised simultaneously by the hill climbing procedure, while the cycle time can be manually or automatically selected based on its effect on junction saturation and delay [79]. Junctions can also be double cycled or have repeated greens [79]. TRANSYT can also optimise the phase sequence, has separate models for buses/trams and normal traffic, and can model pedestrian crossings [80].

2.2.1.3 LinSig

LinSig is a computer software package for the assessment and design of traffic signal junctions [68] for use on stand-alone junctions, multiple traffic signal junctions, complex compound junctions (such as signalled roundabouts), and road networks, which can include traffic signal pedestrian crossings and priority junctions [68].

LinSig uses a combination of geometric layout, traffic behaviour, and controller modelling to accurately represent existing junctions and their effects [80]. Given this information, LinSig can optimise the offsets and green splits for a phase sequence and cycle time chosen by the user [80]. LinSig's optimisations are performed with the aim of increasing capacity or minimising delay [68]. LinSig can model both cyclists and pedestrians at signalised intersections but only has a single traffic model for all vehicle classes [80].

2.2.1.4 Synchro

Synchro is a macroscopic traffic simulation and optimisation tool that is widely used for the performance analysis of signalised intersections and roundabouts [69]. The software optimises cycle lengths, green splits, offsets, and phase sequences [69] to improve a performance index combining stops and delay [81], which is calculated using a formula based on the Webster model [82]. When optimising cycle lengths, the software attempts to find the shortest cycle length that can clear the critical percentile of traffic throughput [69]. Traffic simulations for Synchro are run in the microscopic traffic simulation software SimTraffic, which is coupled with it. [69]

2.2.1.5 Overview

Fixed-time traffic control systems are very predictable which means GLOSA, which requires accurate future signal plans, operates well with them. However, fixed-time plans cannot respond automatically to traffic accidents, roadworks, or changes in live traffic conditions [64] or longer-term changes in traffic patterns [65]. When these things happen the performance of FTTCs is reduced, leading to greater congestion which will negatively impact the performance of GLOSA.

2.2.2 Responsive Traffic Control Systems

The alternative to FTTCs, RTCs, use measured data about traffic conditions to adjust signal plans in real time. RTCs have been used broadly since the eighties [83] with a large number of systems having been designed and deployed, including SCOOT [83], LA ATCS [84], InSync [85], ACS-Lite [86]. Actuated or adaptive traffic control systems vary hugely in their workings, from systems that produce an updated signal plan every 10 minutes [2], to systems that make decisions on a second-by-second basis [87].

In this section, a large number of RTCs have been reviewed with the aim of understanding how they work and what information they collect. This is done to create an understanding of how these systems might interact with GLOSA and to assess which systems might benefit from having GLOSA deployed with them. In many cases, the exact workings of these systems have never been published, and very few field studies exist which directly compare two or more commercial systems [88]. Therefore, this section will, at times, rely on promotional material released by the developers of the systems being reviewed.

2.2.2.1 Split Cycle Offset Optimisation Technique

Split Cycle Offset Optimisation Technique (SCOOT) is a centralised traffic adaptive signal control system developed in the United Kingdom in the early eighties by the Transport Research Lab that has since been installed extensively worldwide [83].

SCOOT uses inductive loops in the carriageway (among other sensor types) [89] to monitor and detect traffic volumes, flow for every group of incoming cars, and the average speed of vehicles approaching the traffic signals (see figure 5) [90]. This data is then used to calculate a performance index (a composite measure of delay, queue length, and stops in the network), which SCOOT then attempts to minimise by adjusting signal timings [83].

It adjusts signal timings by updating the split (the durations of green phases for each signal), cycle (the time taken for one complete sequence of the operation of traffic signals) and offset (the time that each junctions cycle is disjointed compared to a reference cycle) times in the upcoming traffic plan [91].

The Split time is optimised before every stage (a period when one or more non-conflicting phases are given a green signal at the same time) and can either be changed temporarily (by up to four seconds) on any of the inbound roads or permanently (by up to one second) based on the degree of saturation on each inbound road. The Offset time is optimised once per cycle for each junction. It operates by looking for patterns in platoons of vehicles arriving at the junction and then tries to line up the green time with the arrival of platoons. It can be advanced or delayed in four-second increments [91].

The Cycle time (the time taken for one complete sequence of the operation of traffic signals) is optimised at most once every two and a half minutes. The cycle length can be changed by four, eight, sixteen or 32 seconds but cannot exceed 120 seconds or decrease below thirty seconds. In general, the aim of this optimisation is to ensure that the most heavily loaded intersection operates at a maximum degree of saturation of about 90%. If the junction is under-saturated, the cycle time is decreased to reduce the time vehicles wait for a green light. If the junction is over-saturated, the cycle time is increased to lower the proportion of time spent changing phases and therefore maximise the amount of time in which vehicles are moving through the junction [91].

SCOOT also features: public transport and pedestrian priority [92]; accident management, where it registers accidents and breakdowns and takes such information into account to control the traffic signals at the surrounding intersections as per the new situations [90]; and Pedestrian crossing timers [93].

However, SCOOT is unable to manage signals which are closely spaced separately due to its specific requirements of detection configuration [94]. If two or more junctions are too close together, SCOOT will control them as a single junction. TRL's marketing material claims SCOOT typically decreases delay by 15% compared to a fixed time systems [92] while research papers have put that figure closer to 20% to 30% [95; 96; 97; 98].

The flexibility of SCOOT to frequently change the future signal plans, especially the split times, which are optimised after every stage, allows it to better deal with short term changes in traffic flows. However, no consideration is made for approaching GLOSA-equipped vehicles when signal plans are updated and there is only limited delay between the calculation of new signal plans, and their implementation. As such, depending on the way the changes in signal plans are managed, there are two issues that vehicles will face.

If signal timings that assume there will be no changes will occur are sent out, vehicles may often have to adjust their speed curve on approach to correct for any changes that do occur, which would be inefficient and potentially dangerous. Alternatively, if only signal timings that are not subject to change are sent out, many vehicles will only receive relevant signal timings once they are too close to the junction to adopt a more efficient course.

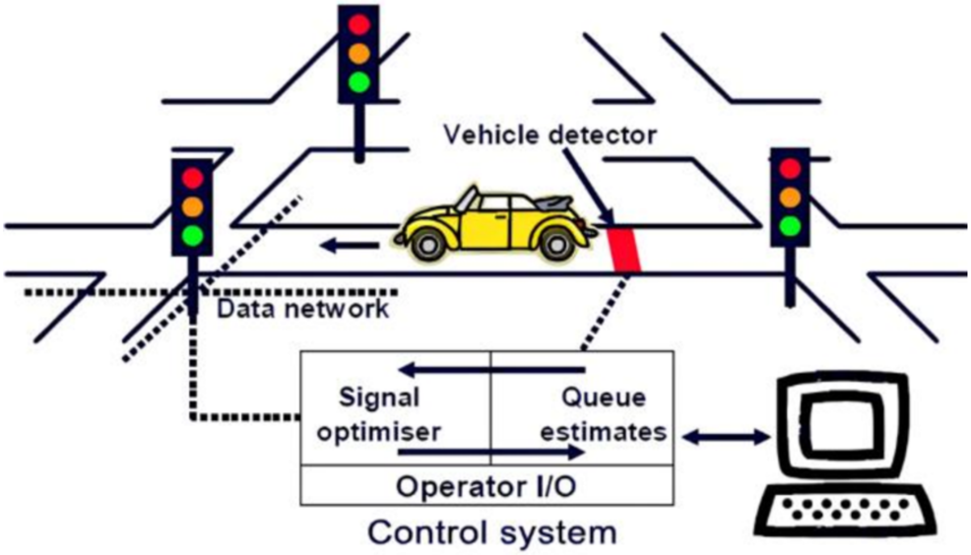

Figure 5: Diagram of the SCOOT traffic control system [99].

2.2.2.2 Adaptive Control Software Lite

Adaptive Control Software Lite (ACS Lite) was developed by the American Federal Highway Administration [86], in partnership with Siemens [3], to provide a “widely deployable” adaptive traffic control system. This system performs optimisation by changing the splits and offsets of signal control patterns, typically by between 2 and 5 seconds (depending on the settings chosen by the operator/installer) [3], roughly every 10 minutes to accommodate changes in traffic conditions [2]. Split adjustments are based on measures of the utilisation of each phase and are subject to minimum green times, pedestrian interval requirements, and maximum green times [100]. Offset adjustments are made to increase the proportion of traffic arriving at a green light [3]. This system’s design focused on lower installation and operation costs, which have been a limiting factor in the deployment of adaptive traffic control systems [100].

This system is decentralised [101] with each intersection operating individually and autonomously [102]. This makes it scalable and easy to apply to large networks. ACS Lite requires a detector at the stop line as well as a detector 80 to 170m upstream [101].

Up until 2012, ACS Lite deployments tended to only include a small number (less than 20) of intersections. A commonly cited issue was that ACS-Lite could not modify the cycle time [1] to deal with fluctuation in traffic demands, making it inefficient in areas where traffic volume changed [86], and it did not perform well when applied to grid networks. More recently, ACS Lite has been updated so that cycle time could be changed using a time-of-day schedule, alleviating some earlier issues [2; 103]. However, this schedule requires maintenance from a traffic engineer as long-term traffic patterns change [2].

Reported benefits of this system include: a 5 to 25% reduction in arterial travel time, 5 to 50% reduction in delays on side streets, a 15% reduction in stops and a 15% reduction in overall delays [1; 2; 3] although one study indicated that, while ACS Lite can potentially improve traffic flow within its own system, it can cause large delays or issues at the boundary intersections in some situations, with the issue attributed to ACS Lite restricting flow into its part of the network [86].

As ACS Lite only updates its signal plans every 10 minutes, it would, for most of the time, work quite well in combination with GLOSA and will also be able to adapt to longer term changes in traffic conditions. However, the presence and location of GLOSA-equipped vehicles are not considered in the timing of the optimisation, which would often lead to vehicles receiving inaccurate signal information or no signal timings. Also, it is unable to respond to phase by phase or second by second changes in traffic conditions, and therefore it will suffer from many of the shortcomings previously discussed when examining the combination of fixed time traffic control systems with GLOSA.

2.2.2.3 Balancing Adaptive Network Control Method

Balancing Adaptive Network Control Method (PTV BALANCE), which was developed at the Technical University of Munich by Prof Dr Bernhard Friedrich in the late nineties [104], uses a genetic algorithm to optimise future signal plans [105; 2] for an entire network [106].

It uses a macroscopic traffic model that estimates network traffic flows in accordance with detector data, a control model, and a mesoscopic traffic flow model to calculate the effects of a specific signal plan and, most importantly, different optimisation algorithms [106; 107]. It follows a cyclic control strategy [108], meaning this system never changes the order of the phases in its cycle and instead just alters the cycle, split and offset times [109; 106].

As of 2017, BALANCE has been set up in Hamburg, Ingolstadt, and other cities with a total of more than one hundred light signals. BALANCE runs every 5 minutes and calculates optimised signal plans for the following 5-minute period [106]. One advantage of BALANCE is it is an open system and thus not bound to any particular manufacturer's equipment or detectors [2]. Similar to ACS Lite, Balance only updates its signal plans every 5 minutes. So, for most of the time, it would work quite well in combination with GLOSA. However, the presence and location of GLOSA-equipped vehicles are not considered in the timing of the optimisation, and so optimisations would often lead to vehicles receiving inaccurate signal information or no signal timings. Also, Balance is unable to respond to phase by phase or second by second changes in traffic conditions, and therefore it will suffer from some of the shortcomings previously discussed when examining the combination of fixed time traffic control systems with GLOSA.

2.2.2.4 Entire Priority Intersection Control System

Entire Priority Intersection Control System (PTV EPICS) is for isolated intersections [106]. Using information from detectors (both current and historical), oncoming public transport (via radio or vehicle to infrastructure communication), and queue length estimators, EPICS estimates the traffic inflow to the intersection every second for the next 100 seconds (see figure 6) [104]. Using this prediction, optimisations are performed every second [106; 104].

It uses a two-step approach. First, a time-ordered shortest-way algorithm chooses the right stage sequence. Second, a hill-climbing approach fine-tunes the starting points of the intergreen times. The performance index to be minimised is the weighted sum of the delay for all approaches and traffic modes detected at this intersection [104].

EPICS is often used in conjunction with BALANCE [110; 111]. EPICS can function with any detector position, but optimal detection is at a distance of about 50-80 meters in front of the stop line [110]. Additionally, stop line detection can be added, but they are not crucial [110].

As PTV EPICS optimises signal plans every second, it can leverage a huge amount of flexibility to adapt to live changes in traffic condition. However, as GLOSA-equipped vehicles are not considered by this system, those vehicles would frequently receive incorrect signal timings or no signal timings. This means that the benefits of implementing GLOSA are not available on PTV EPICS controlled junctions.

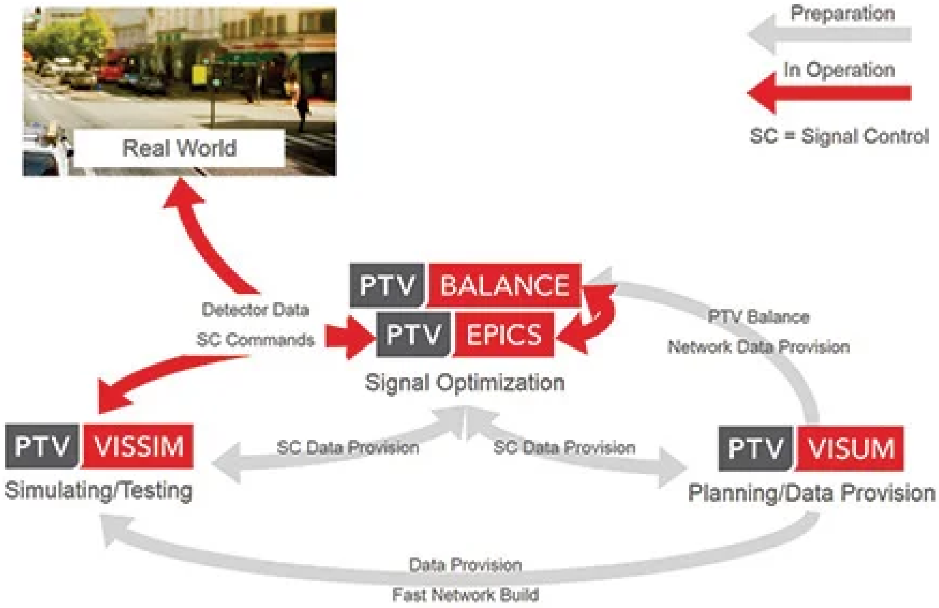

Figure 6: Diagram of the PTV BALANCE and PTV EPICS traffic control systems [111].

2.2.2.5 InSync ATCs

InSync, developed by Rhythm Engineering, performs local optimisation and global optimisation [85]. At the global level, InSync creates green tunnels, where platoons of vehicles gather and are then released all together along the entire corridor [85; 2]. By coordinating with each other, the signals anticipate the green tunnel's and the platoon's arrival, so the platoon can pass through without slowing down or stopping [85; 2]. The duration, period (time between green tunnels) and frequency of the green tunnels can vary to best support traffic conditions [85; 2]. The period is decided by looking at queue lengths and percentage of occupancy for each phase at similar times of day and days of the week over the last four weeks [112].

When green tunnels are not active, each junction acts as a fully actuated traffic signal control system [112], and local optimisation serve the side streets [85; 2]. Locally, it uses integrated digital sensors to know the exact number of cars demanding service at an intersection and the duration for which they have been waiting. Based on this queue and delay data, approaches are given phasing priority [85; 112]. However, priority will be given to the side streets if a green tunnel has just ended and conversely given to the main route if a green tunnel is due [112]. For further reading on the workings of InSync ATCs, see Chandra et al. [113] and Chandra et al. [114].

As of November 2015, InSync is operational in 2,300 traffic signals in thirty-one states and 160 municipalities in the U.S. [85]. The benefits of InSync ATCs are cited as decreased travel times, stops, and stopped times; increased average speeds; reduced emissions [4]; and increased safety [5]. For more details on the performance of Insync ATCs, see Dakic et al. [115], Stevanovic et al. [116] and Selinger & Schmidt [1].

InSync selects which signal phases will be run after a currently running green ends and dynamically adjusts the time between green tunnels [112; 117]. Because of this, InSync can leverage a lot of flexibility to adapt to live changes in traffic conditions. However, as GLOSA-equipped vehicles are not considered by this system, those vehicles would frequently receive limited or no signal timings, leading the system to be inefficient when used in combination with GLOSA during normal operation.

2.2.2.6 Sydney Coordinated Adaptive Traffic System

Sydney Coordinated Adaptive Traffic System (SCATS) is a two-level hierarchical adaptive traffic signal control system (see figure 7) [118; 119] developed by the Department of Main Roads (Roads and Traffic Authority) of New South Wales in Australia [120]. SCATS uses information from a video imaging processing system called Autoscope, located in each lane immediately in advance of the stop line, to detect vehicles queued at the traffic signal, along with other traffic flow parameters [120]. This information is used to calculate Degrees of Saturation and Link Flows [118; 119] which are measured each cycle and used to calculate cycle lengths, splits, and offsets for the following cycle [119]. The SCATS strategy assumes that higher cycle lengths increase intersection capacity and splits proportional to approach demand and provide longer offsets for increased traffic volumes [118; 119]. For saturated and over-saturated traffic conditions, SCATS usually abandons the concept of splits proportional to saturation and provides more green time for higher traffic flows on major roads [119].

SCATS has a hierarchical structure. At the highest level there is a central management computer, which serves primarily as a repository for traffic volume data and networks all the regional computers together into one system, making all junctions accessible through the SCATS user interface.

Then below that there are several regional computers (each one capable of controlling up to 250 intersections) [121; 122]. The junctions, each regional computer controls, are split up into groups, often referred to as subsystems, where each junction will have a common cycle time [121; 122]. These subsystems can then be connected by links designed to synchronise the flow between them [121; 122].

For further reading on the workings of SCATS, see: McCann [121] and Sims & Dobinson [122]. The SCATS system has been used in Hong Kong, Sydney, Melbourne and Oakland County, Michigan [120]. For details on the performance of SCATS, see: Chong-White et al. [123], Hunter et al. [124], Dutta et al. [120], and Slavin et al. [125].

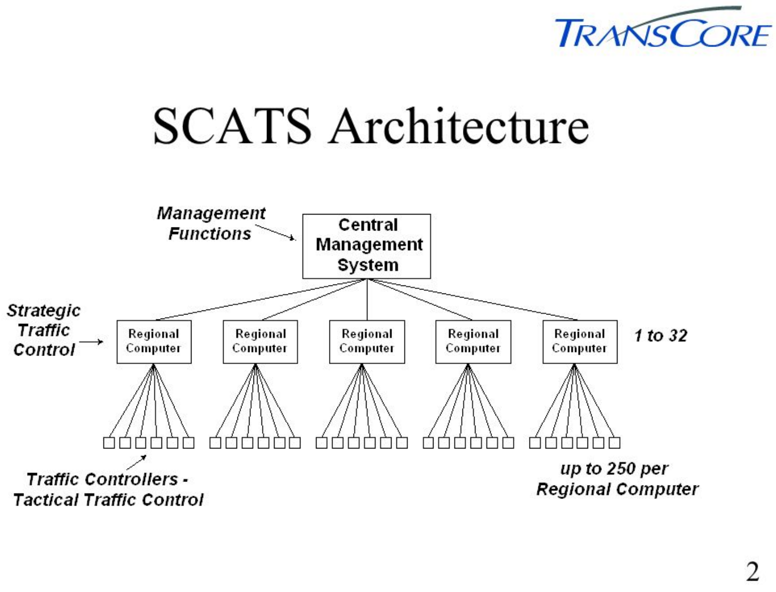

Figure 7: Diagram of the SCATS traffic control systems architecture [126].

2.2.2.7 Los Angeles ATCS

Los Angeles ATCS (LA-ATCS) was developed in 2001 by the Los Angeles Department of Transport [84]. As of 2017, LA-ATCS operated over three thousand intersections in the city of Los

Angeles [2]. The LA-ATCS updates cycle lengths, splits, and offsets at each intersection once per cycle based on prevailing traffic conditions [127]. The adaptive adjustments of the signal timings are based on changes in detector data, such as changing volumes and occupancies, at each intersection [128], which are collected every second but used every cycle [127]. Changes to these parameters are incremental, although the system can vary the size of these increments up to a certain threshold, depending on how quickly traffic conditions are changing [128]. Splits are based on traffic volumes and occupancies of each approach [128]. Offsets are based on minimising the number of stops for the approach with the highest flows (with special priority given to coordinated directions) [128]. A section-wide cycle length is based on the minimum time needed to keep all signals in a particular section operating below saturation [128]. For minor intersections, the system runs double-cycled operations to reduce unnecessary delays [127]. The optimisers used for splits and offsets are referred to as "Critical Intersection Control" and "Critical Link Control," respectively [101]. Each optimiser can function independently of the others [127].

Detectors are usually located 200 to 300 ft upstream of the intersection, which allows the system to measure platoon arrival patterns and collect a set of useful traffic metrics such as volume, occupancy, speed, stops, queue, and delay [127**?**]. LA ATCS requires at least one detector per lane for each phase [127]. Bluetooth units, installed in several of the deployments, anonymously collect the unique identifiers of discoverable Bluetooth devices within range (about 100 ft) every 5 seconds; this information is then used to estimate travel times [128]. Downtime is cited at only 1% [1].

LA-ATCS only updates its signal plans once per cycle time, and therefore cannot respond to second to second or phase to phase changes in traffic as well as more flexible RTCs with more frequent cycle/plan updates. However, for the vast majority of the time, it would work quite well in combination with GLOSA, although it does not consider the presence and location of GLOSA-equipped vehicles in the timing of the optimisation process, and so optimisations would often lead to vehicles receiving inaccurate signal information or no signal timings.

2.2.2.8 Method for the Optimisation of Traffic Signals in Online-Controlled Networks

Method for the Optimisation of Traffic Signals in Online-Controlled Networks (MOTION) is a two-level RTCS, which produces new network-wide single plan frameworks every 10 to 20 minutes [129], while local controller adapts these plans (within set limits) to suit local traffic conditions [130], however, it is unclear how much the local plans can change the more strategic network plan [131]. Optimisation normally aims at minimising delays and stops in the network [132]. At the network level, optimisation typically starts with the dominant traffic stream through the network and attempts to construct a grid of Green Waves, considering modelled (or measured) platoons in the links [132].

MOTION optimises the cycle time, stage sequence, offset, and splits for each junction [132]. These optimisations are then implemented, but only if the calculation determines that there would be a significant improvement in the overall performance of the network [133]. This avoids frequent minor changes [133]. To avoid severe disruptions in traffic flow due to the network-wide change of signal plans, a smooth transition is performed [132]. Once the optimal signal plan has been implemented, some bandwidth remains available for the subsequent local optimisation [132]. How much the local controllers can change depends on the remaining spare time per intersection and the constraints of the optimised offsets [132].

MOTION uses detectors placed 10 to 50m and 50 to 200m upstream of each intersection [101] that collect data on traffic volumes, platoons, and occupancies [132]. MOTION can give public transport priority by limiting the range of options for local optimisation of stage sequence, split and offset to those that provide a green time window for public transport vehicles at their expected arrival times [134].

While the 10 to 20 minutes of future signal plans produced by MOTION's higher-level systems would be good for GLOSA, the lower level is allowed to adapt these signal plans. This allows MOTION to better adapt to short term changes in traffic conditions but reduces the reliability of future signal plans.

2.2.2.9 Optimised Policies for Adaptive Control

Optimised Policies for Adaptive Control (OPAC) was developed by PB Farradyne [135] based on research presented by Professor N. Gartner in 1983 [136] at the University of Lowell under the sponsorship of U.S. Department of Transportation [137]. It was originally developed for individual intersections [137], with the aim of reducing total delay [138] and stops [137]. However, later versions of OPAC were expanded to include an option for coordination between intersections, which is suitable for implementation on arterials and networks [136], as part of the RT-TRACS project [139].

OPAC is acyclic and makes phase-switching decisions at fixed time intervals [137]. Using simplified Dynamic Programming techniques [137] and Rolling Horizon approach [136], OPAC continuously produces new plans for the next 50 to 100 seconds [140; 137; 141]. Because of this, the exact remaining duration of the current phase is never pre-specified and depends solely on the prevailing traffic flow conditions [137; 141].

Each 50 to 100-second plan must include at least one and no more than three-phase changes [140]. The plan is constructed based on: data from upstream detectors, between 130 to 200m away from the junction; data from detectors located at the stop line; and historical data [101], including measurements of queue lengths, arrivals and departures [138] from the past 50 to 100 seconds [136]. For further reading on the workings of OPAC, see: Gartner [138].

As OPAC performs updates continuously it has an extremely high degree of flexibility, allowing it to manage phase by phase and second by second fluctuations in traffic flow. However, such flexibility without considering GLOSA-equipped vehicles means that future signal plans are constantly subject to change.

2.2.2.10 Real-Time Hierarchical Optimised Distributed Effective System

Real-time hierarchical Optimised Distributed Effective System (RHODES) is a three-level decentralised system (see figure 8) [101], developed by the University of Arizona in 1991 [**?**], that uses dynamic programming to optimise a given performance criterion selected by the user (such as average delays, stops, and throughput) [142].

At the highest level, details like network geometry, network demand and the typical route selection of travellers are considered [142]. Using this information, RHODES produces estimates for the load on each road in vehicles per hour [142]. Typically, these estimates are updated every hour [87].

At the middle level, often referred to as “network flow control” [143], predictions of platoon flow are created and used with the estimated loads generated by the highest level to create target signal timings [87]. This happens every 200 to 300 seconds approximately [87].

Finally, at the lowest level, often termed “intersection control” [143], predictions of vehicle flow are created and used to alter the target signal timings so that they take local conditions into account, as well as the approaching platoons predicted at the previous level [87]. This happens on a second-by-second basis [87]. Intersection control considers a time horizon of 45 to 60 seconds over which it assigns time to phases in a fixed order [143]. Phases can be skipped if the system decides to assign them no time [143]. To do this, upstream detectors are usually placed 30 - 50 m from the stop lines of each intersection [101; 141].

RHODES has a bus priority module, referred to as “BUSBAND”, which uses the exact locations of the buses in the network and the passenger counts of those buses to assign the buses a weight that depends on the number of passengers and whether the bus is behind schedule [144].

As RHODES optimises signal plans every second, it has an extremely high degree of flexibility to allow for better managing phase by phase or second by second fluctuation in traffic flow. However, as GLOSA-equipped vehicles are not considered by this system, those vehicles would frequently receive incorrect signal timings, leading the system to be inefficient when used in combination with GLOSA.

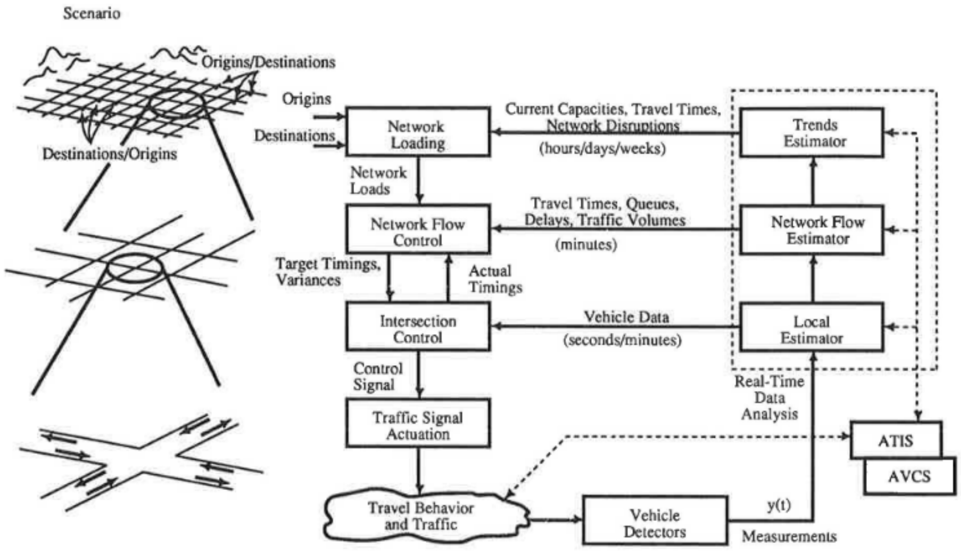

Figure 8: Diagram of the RHODES three-level decentralised system [145].

2.2.2.11 Composite Signal Control Strategy System

Composite Signal Control Strategy System (CoSiCoSt) is an urban traffic control system developed by CDAC (Centre for Development of Advanced Computing) in India [94; 146], which optimises a weighted combination of delay and number of stops in real-time [147].

CoSiCoSt is designed to cater to typical Indian driving behaviours and traffic conditions such as poor lane discipline and unpredictability [147]. It does not make predictions of incoming traffic or classify approaching vehicles by type [148]. Instead, relying only on detectors near stop lines, which are preferred to upstream detectors, as they help overcome the issues of non-lane following traffic and intrusion from uncontrolled side roads and parking [148].

This system divides the network into arterial flows that share a common cycle time (see figure 9). Along these arterial flows, offsets are set to provide green waves [146; 2]. The split times are decided in a two-level process [146; 2]. A central controller chooses split timings for each intersection based on the demand trend analysis [146; 2]. After this, the local controllers can modify the split timings slightly based on actual measured traffic conditions [146; 2].

During a link's green phase, but after the minimum green time has elapsed, the system will check for the presence of vehicles at the stop line every second [148]. If the stop line is vacant for two consecutive seconds, the phase terminates [148]. If this condition is not met, the phase will continue until the maximum green time has elapsed [148]. If queues are still detected after a phase has finished, the system will increase the split time awarded to that phase in the next cycle [146; 2]. Should this fail to clear the queues, the cycle time of the arterial flow will be increased [146; 2].

CoSiCoSt's aptitude for use with GLOSA is like that of LA-ATCS. However, the two-second rule, employed to decide when phases end, adds a level of uncertainty to future signal plans that would negatively affect both equipped vehicles approaching an active green phase, which can't determine if the lights will still be green when they arrive, and equipped vehicles approaching an upcoming green phase, which can't determine when exactly the green phase will begin.

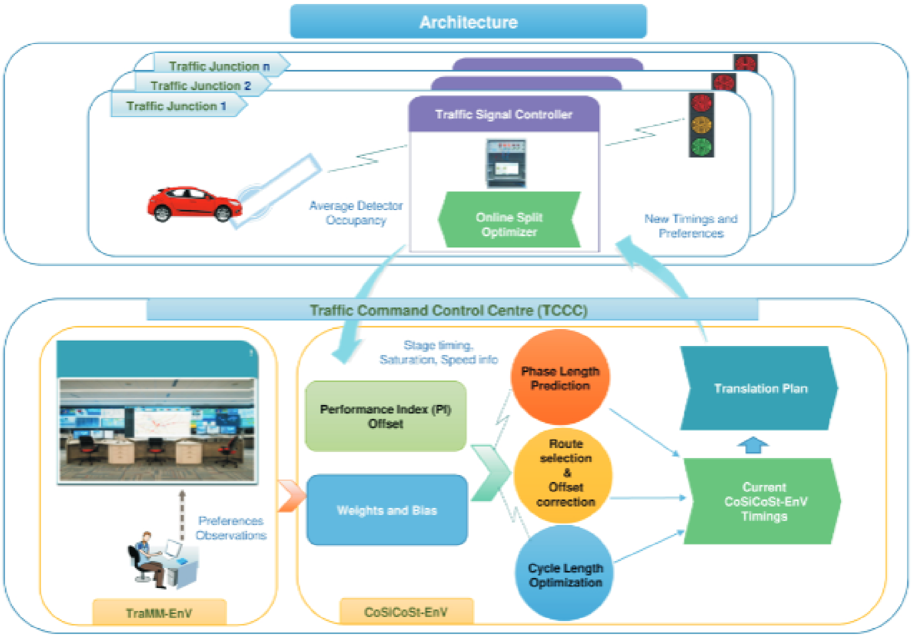
Architecture
Traffic Junction n
Traffic Junction 2
Traffic Junction 1
Traffic Signal Controller
Average Detector Occupancy
Online Split Optimizer
New Timings and Preferences
Traffic Command Control Centre (TCCC)
Stage timing, Saturation, Speed info
Performance Index (PI) Offset
Weights and Bias
Phase Length Prediction
Route selection & Offset correction
Cycle Length Optimization
Translation Plan
Current CoSiCoSt-EnV Timings
Preferences Observations
TraMM-EnV
CoSiCoSt-EnV

Figure 9: Architecture diagram of CoSiCoSt [149].

2.2.2.12 TASS

The Traffic Actuated Signal Plan Selection (TASS) system, developed by Siemens [150], is a decentralised area traffic control system [151] that selects one out of six pre-defined signal plans (each with different cycle lengths, splits, and offsets) every 15 minutes, depending on the current traffic conditions in the network [152] and transfers the selected plan to the junction controllers for application [152].

The junction controllers may modify (within certain limits) the received signal settings by application of a simple traffic actuation logic based on local traffic measurements [152]. The cycle times of the available plans are typically 60 to 120 seconds [153]. Double-cycling can be enabled on boundary junctions with moderate loads [153].

As TASS only updates its signal plans every 15 minutes, it would, for much of the time, work quite well in combination with GLOSA. However, the presence and location of GLOSA-equipped vehicles are not considered in the timing of the optimisation, which would often lead to vehicles receiving inaccurate signal information or no signal timings. Also, TASS is not able to adapt to phase by phase or second by second fluctuation in traffic flow.

2.2.2.13 Microprocessor Optimised Vehicle Actuation

The Transport Research Laboratory developed Microprocessor Optimised Vehicle Actuation (MOVA) [154] as an actuated signal control system initially designed for use on isolated junctions [155].

The system generates signal timings once per cycle but varies these timings continuously according to the latest traffic condition [156]. There are two operational modes specified for uncongested and congested conditions [156]. In the uncongested mode, delay and stop are minimised, while in the congested mode, capacity is maximised [156; 157]. MOVA evaluates its signal plans every half second [156].

Although MOVA was initially developed to control isolated junctions, it was later upgraded to enable links between sets of signals, enabling its use on large gyratory and grade-separated signal roundabouts [154]. MOVA is thought to be installed at half of all the UK junctions where it could be installed, with 200 to 300 installations being added annually [158].

As MOVA optimises signal plans every half second, and GLOSA-equipped vehicles are not considered by this system, those vehicles would frequently receive incorrect signal timings, leading the system to be inefficient when used in combination with GLOSA.

2.2.2.14 PRODYN

PRODYN is a real-time hierarchical [159] traffic control system developed by CERT/ONERA in France that has been implemented in three French cities and Brussels [160], which attempts to minimise the total delay [161]. At a network level, it uses decomposition coordination methods, and at the intersection level, it uses forward dynamic programming [162].

PRODYN controls traffic signals by frequently deciding whether to switch the phase of the traffic lights of each intersection [159]. These decisions are made every 5 seconds with a 75-second rolling horizon considered for optimisation [160], with constraints on minimum and maximum stage times also considered [161].

PRODYN uses two magnetic loop sensors in each lane (one at the entrance of the link and another fifty meters from the stop line) [161] and manages coordination between junctions by exchanging platoon forecasts between upstream and downstream intersections [160]. PRODYN can provide priority to buses using a GPS-based system and the vehicle's odometer to determine each bus's positions [160].

Due to its brief time between decisions, PRODYN can adjust well to short term changes in traffic flows, but it is limited to producing only 5 seconds of future signal plans and does not consider GLOSA-equipped vehicles in its optimisations. Therefore, it is not suitable for use in combination with GLOSA.

2.2.2.15 Urban Traffic Optimisation by Integrated Automation

Urban Traffic Optimisation by Integrated Automation (UTOPIA) is an adaptive traffic control system that has been used in Turin since 1984 and has since spread to other areas in Europe as well as North America [163; 100]. UTOPIA has a three-tiered hierarchical architectural system [131]. The area level consists of a central system responsible for the medium and long-term forecasting and control over the entire deployment [164]. At this level, signal reference plans and conditions for adaptive coordination are calculated. In addition, diagnostic activity for each local controller is continuously conducted at this level [164]. The area level has access to the average speed of vehicles through the area and the level of saturation of each junction and produces forecasts and conditions every 30 minutes [165].

The local level consists of local controllers responsible for determining the optimal sequence and length of traffic light phases based on the conditions determined by the upper level and traffic information supplied by controllers both locally and at adjacent intersections [164]. Optimisation at the local level is conducted over a time horizon consisting of the next 120 seconds and is repeated every 3 seconds [165].

The third level, called the town supervisor level, uses a macroscopic model to integrate congestion information with data from other systems, such as bus travel times [131]. UTOPIA provides bus priority by shifting planned green windows to coincide with the anticipated arrival times of buses [131]. Bus location technology is used far upstream of signalised junctions, and the system can gradually adapt the junctions to match the arrival times [131].

UTOPIA uses loop detectors in the network, located just downstream of upstream junctions [131]. The benefits due to the implementation of UTOPIA have been recorded as an increase of 15% in average speed for private vehicles and, when the public transport module is installed [100], a 28% increase in average speed for public transport with priority [166]. Travel times have also been decreased by 10% for cars and, with the public transport module, 2 to 7% for public transport [163].

As UTOPIA optimises signal plans every three seconds, it can better adjust to short term changes in traffic conditions. However, as GLOSA-equipped vehicles are not considered by this system, those vehicles would frequently receive incorrect signal timings, leading the system to be inefficient when used in combination with GLOSA.

2.2.2.16 Other Systems

A few other RTCs exist, including: Sitraffic FUSION [167], Adaptive Traffic Light Timer Control (ATLTC) [168], Intelligent Traffic Area Control Agent (ITACA) [169], TUC [170], System D [171], and Green Link Determining (GLIDE) [172]. However, these RTCs all behave similarly to the RTCs reviewed previously and, therefore, are incompatible with GLOSA for the reasons previously discussed in this section.

2.2.2.17 Overview

As responsive traffic control systems can adjust to changes in live traffic conditions in real-time, they can reduce travel time, delays, stops [1; 2; 3], and stopped times, as well as increase average speeds; reduce emissions [4]; and increase safety [5]. However, it is clear from this review that the existing responsive traffic control systems are incompatible with GLOSA without the implementation of some mitigating solution.

2.3 Recap and Discussion 1

While GLOSA offers clear benefits, including better fuel consumption, less stops, improved ride smoothness, reduced travel times, and improved capacity, for a wide range of vehicles, including cars, buses, lorries, HEVs, EVs and emergency vehicles, these benefits are currently out of reach because GLOSA is incompatible with the already widely deployed RTC systems.

However, as it has been demonstrated that GLOSA could have significant benefits to road users if it could be implemented with RTC systems, the next part of this review will look at two existing potential solutions, Phase Prediction algorithms and Junction Controlled Vehicle methods, focusing on the results that have been achieved by existing frameworks and highlighting the issues faced by such solutions that make them impractical to deploy.

2.4 Phase Prediction Algorithms

Phase Prediction Algorithms use methods like Markov chains [15] or reinforcement learning [16] to make predictions about the future signal plans of oncoming junctions. These predictions can then be used to activate GLOSA.

The older Markov chain based methods used graph theory, to create a transition graph of signal changes, and calculated occurrence probabilities for those transitions using real-life observations and recorded detector data of the traffic light [15]. This could be used to predict the future state of the traffic lights to within a safe degree of accuracy more than 80% of the time [15].

When this method was explored further it was found that a time-aware system that uses empiric data from similar time slots to predict future signal changes outperformed general approaches [173]. Furthermore, the consideration of traffic detectors improves the overall forecast accuracy; therefore, it was recommended that controllers should continuously report the detector readings back to the prognosis system [173]. This system was validated by means of simulations and real-life test drives in the Travolution testbed in Ingolstadt, Germany [173].

Predictive systems have also been designed which use traffic signal controller high-resolution event data, from similar time-of-day and day-of-week periods, to calculate phase probability

[174]. These methods eliminate the need for real-time communications from the vehicle to infrastructure, removing data-loss-related issues [174]. A proof of concept was conducted by simulating drives through a test route composed of an arterial that had historical high-resolution traffic signal event logs for a series of actuated-coordinated traffic signals [174]. A significant reduction was observed in the amount of hard braking and the number of crossings through red lights [174].

A system called GreenDrive used similar ideas of collecting data to use in forecasting traffic lights but did so through the tracking of mobile phones that were running an app [175]. The system recorded GPS, Accelerometer and Magnetometer data from the phones and used this information to build a model of traffic control systems, which could then be used for signal timing predictions [175]. In simulations and real-world experiments, GreenDrive learned to predict phase durations with an average error of fewer than 2 seconds and was able to reduce the fuel consumption of vehicles. However, travel time was increased [175].

SpeedAdv, another predictive GLOSA, used machine and reinforcement learning to both predict future states of traffic lights and advise vehicles on the optimal speed using current traffic conditions, historical signal data and other information collected by vehicles [176]. SpeedAdv was implemented and evaluated with a field test which demonstrated improvements in travel time, energy consumption, safety, and comfort compared to the GreenDrive method [176].

Researchers working on the Universal Traffic Management Systems (UTMS) project have also begun incorporating similar predictive methods into their designs [177].

Supervised machine learning can also be used to make predictions about future signal plans on fully-actuated signal control systems, and it has been demonstrated that by utilising live data from loop detectors, as well as information about the current state of the traffic signals, the time to green can be correctly predicted to within two seconds around 80% of the time [16].

On the whole these methods seem promising however they are currently held back by their error rate, with the systems we have discussed making a significant error in their prediction 10 to 20% of the time. If a prediction puts the green phase later than reality, the phase may let less vehicles through than it otherwise would, negatively effecting the operation of the junction. However, more importantly, if a prediction puts the green phase earlier than reality, vehicles will be forced to stop sharply and needlessly at the red lights, potentially causing accidents. Even worse, a distracted driver could fail to observe that the lights are still red and cause an accident. Because of this, phase prediction algorithms are not presently ready for real world implementation.

2.5 Junction Controlled Vehicles Frameworks

In Junction Controlled Vehicles frameworks, CAVs (Connected and Autonomous Vehicles) approaching a junction are controlled by the junction's RTCs framework directly instead of just informed of the optimal time to arrive [178]. One advantage of using Junction Controlled Vehicles and CAVs is that it avoids the issues of human drivers being unable to keep absolutely to the speed advice they are given.

Junction Controlled Vehicles frameworks can be built using dynamic programming to find near-optimal signal plans and minimise vehicle delay, and optimal control theory to calculate near-optimal speed profiles for vehicles to minimise fuel consumption and emissions [179]. These frameworks have been shown to reduce both vehicle delay and emissions under a variety of demand levels compared to FTTC and RTC when vehicle trajectories are not optimised [179].

One of the earliest dynamic programming-based Junction Controlled Vehicles frameworks was AGLOSA [180]. It worked by producing near-optimal traffic signal plans up to a rolling time horizon and assigning vehicles an arrival time [180].

AGLOSA performs well in terms of minimising average time loss compared to fixed time with GLOSA and RTCs. However, it can produce excessive maximum time loss values in high asymmetrical demand scenarios, as the objective function minimises average time loss with no regard to maximum time loss and no constraints on maximum phase durations [180].

Another Junction Controlled Vehicles frameworks reduced travel time and fuel consumption even with low levels of penetration. In mixed traffic scenarios, system performance improves with increasing market penetration rates [181]. It was also demonstrated that the framework could be implemented in real-time [181].

Similar Junction Controlled Vehicles frameworks, for traffic signals and vehicle trajectories, were able to reduce gasoline consumption, transportation emissions and vehicle delay in scenarios including mixed traffic flows [182; 183].

Another system developed for this situation was R-GLOSA, which has been tested on a simulated four junction arterial. It uses a genetic algorithm to decide on the optimal speed profile for either one or many junctions along a route [184]. The simulation results showed that R-GLOSA outperformed a non-autonomous car equipped with GLOSA in terms of travel time and waiting time in either mode [184]. In addition, R-GLOSA in multi-segment mode could provide improved fuel efficiency and reduced emission CO_2 at some vehicle densities [184].

A Junction Controlled Vehicles frameworks designed for electric vehicles was able to reduce both energy consumption and travel time in a real-time simulation of a busy corridor in the City of Edmonton, Canada [185].

Junction Controlled Vehicles frameworks can also employ reinforcement learning to control the CAVs. For example, some frameworks train vehicle agents with the aim of producing eco-driving behaviours [186; 187; 188]. It has demonstrated that these frameworks can improve energy consumption compared to a behavioural car-following model, with only marginal compromise to travel efficiency [186; 187].

2.5.1 Slot-based Intersections and Passing Rules

A more extreme version of the above systems is Slot-based intersections where vehicles are allowed to cross the intersection through the gap between two vehicles that are coming from conflicting directions [17]. This is done by assigning each approaching vehicle a precise arrival time at which to reach the junction and a specific speed profile to follow on route to the junction to ensure their arrival and clearance at the aforementioned times [17]. This, in theory, allows vehicles to safely share the junction with vehicles on conflicting courses, or routes, and cross the junction with minimal change in speed [17]. When evaluated in isolated junction simulations, it has been shown that Slot-based intersections have increased capacity compared to junctions controlled by RTCs [17]. The capacity can also be increased by decreasing the defined minimum safe gap [17]. Although, one problem with Slot-based Intersections is that any drop or delay in communications between the infrastructure and vehicles could result in a high-speed collision.

Other researchers have achieved better throughput, reduced number of stops, reduced delay and lower trip delays using these systems compared to a traditional traffic management method [18; 19; 20; 21; 22].

These strategies were further expanded for operation on connected networks where, in simulations, it has been shown to reduce the travel time, average delay, average number of stops, and average delay at stops, compared to the case that only signal timing parameters are optimised [189]. Importantly, the method demonstrated was able to perform the required optimisations in real-time [189]. Other schemes showed similar promise by reducing vehicular emission reduction [190].

One issue with joint control frameworks for CAVs is they can present privacy concerns as they require knowledge of the positions, speeds, and routes of every approaching vehicle. To tackle this issue, some researchers have proposed privacy-preserving adaptive traffic signal control methods which can calculate key traffic quantities without having to reveal the confidential data of road users [191]. Evaluation results have shown that the systems can preserve privacy with only a marginal impact on control performance [191].

However, Slot-based intersections cannot serve non-CAVs and would have to be used in CAV dedicated zones (see figure 10). However, their ideas can be implemented in mixed traffic scenarios. For example, deep reinforcement learning (DRL) powered control systems can be used with a passing rule for CAVs [24]. The traffic signal control strategy allows traffic lights to adaptively adjust their phase and duration based on real-time traffic information, while the passing rule allows CAVs that meet certain safety constraints to form platoons and pass through the intersection in a coordinated manner regardless of traffic signals [24]. It has been shown that such systems can reduce travel time and fuel consumption under a low CAV penetration rate but also enlarge its advantages with the increase of CAV penetration rate [24].

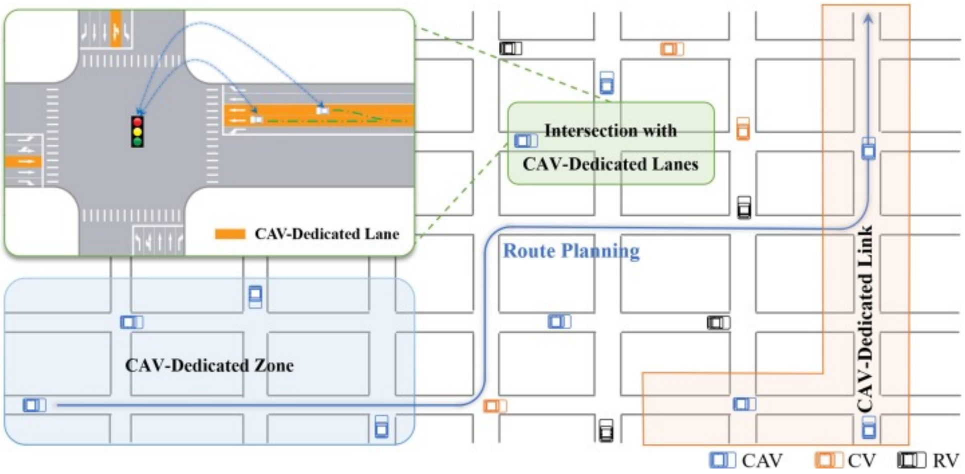

Figure 10: A mixed traffic control scenario including both Slot Based Intersections in CAV-dedicated Zones, and junctions with CAV passing rules using CAV-Dedicated Lanes [178].

2.5.2 CAV Development

While Junction Controlled Vehicles frameworks have thus far recorded impressive results across the board in simulation testing, their timetable for real world use is dependent on the creation and widespread uptake of CAVs, as these vehicles need to be widespread for Junction Controlled Vehicles frameworks to have any performance benefits, and need to saturate the market to make Slot-based Intersections implementable. However, current estimates on when CAVs will be released and widely adopted vary massively. The most optimistic predictions suggest that CAVs will make up all vehicle sales by 2030 [192], while the most pessimistic predict it could take until 2100 for all road vehicles to be CAVs [194]. More commonly, it is predicted that CAVs will make up most road vehicles by the 2050s [193; 192] but that is still 25 years away.

Therefore, Junction Controlled Vehicles frameworks are unimplementable for the foreseeable future.

2.6 Recap and Discussion 2

So far, it has been demonstrated that GLOSA offers clear benefits that are currently out of reach because such systems are incompatible with the already widely deployed RTC systems. Also, the methods discussed so far, Phase Prediction algorithms and Junction Controlled Vehicles frameworks, which attempt to tackle this issue are not suitable for deployment at this time.

Phase Prediction algorithms, which are typically only accurate to within 2 seconds 80% of the time, will cause vehicles to make unnecessary stops, while others may run red lights if the driver was not paying attention because the green phase was later than expected. Meanwhile, Junction Controlled Vehicles frameworks depend on the widespread adoption of a technology that some predict could be 75 years away. Therefore, these options will be rejected as potential solutions and this review will continue to another existing potential solution, Reinforcement Learning Traffic Control (RLTC) systems. As before, the focus shall be on the results that have been achieved by existing frameworks and as well as highlighting the issues currently faced in this field.

2.7 Reinforcement Learning Traffic Control Systems

RLTC systems are a subset of RTCs that use reinforcement learning, a form of machine learning that allows an AI-driven system (referred to as an agent), to learn traffic control through a combination of trial and error, as well as feedback from a predefined reward function. An advantage of Reinforcement Learning methods is that they can find near-optimal strategies for problems that would be highly time-consuming or impossible to solve and code exhaustively [**?**]. Because of this, In recent years a large amount of research time has been put into developing RLTC systems (see table 6 in the Appendix for examples).

However, Reinforcement Learning's ability to find solutions to complex problems, means it can also be used to create flexible traffic control strategies that can take into consideration the effects that different future signal plans might have on GLOSA-equipped vehicles, and can therefore be used as part of joint control frameworks.

In most cases, these frameworks deploy multiple reinforcement learning agents, with each agent either controlling the traffic signals at a single junction or the movements of a single vehicle [27]. In early examples, the vehicle control agents were limited to only three actions: accelerate, decelerate, and maintain speed [27].

However, the CoTV framework used an expanded vehicle action space that allowed for a continuous range of acceleration values to be selected, allowing for more specific control of vehicles [25]. However, CoTV only cooperates with one CAV (the nearest to the traffic light controller) on each incoming road [25]. CoTV can significantly improve traffic efficiency (i.e., travel time, fuel consumption, and CO_2 emissions) as well as traffic safety (i.e., time-to-collision), outperform other DRL-based systems that control either traffic light signal or vehicle speed, and non-DRL joint control methods based on GLOSA [25]. This was validated in various grid maps and realistic urban scenarios. The robustness of CoTV was also validated under different penetration rates of CAV [25].

While these RLTC joint control frameworks can operate in mixed traffic flows, they will be unable to instruct non-autonomous vehicles on the optimal course [28]. Another issue is a lack of investigation into how the system might perform if the proportion of GLOSA equipped vehicles changes. Also, in the case of arterial flow or connected network scenarios where the framework controls a large number junctions, the framework assumes CAVs are broadcasting their intended routes and it is not explored how the system might behave if the intended routes of the vehicles are unknown to the framework, either because of communication errors, or a wish for privacy by drivers/passengers. In this case the framework may perform worse due to having to make more predictions about the approaching traffic at each junction. Also, while these frameworks do demonstrate that RLTC systems can work as part of joint control frameworks in mixed vehicle condition, as these frameworks work in part by having the junction pilot CAVs, they will be unimplementable until such time as there are a sufficient number of CAVs on the roads.

Another similar framework combined traffic signal control and vehicle routing in signalised road networks using Multi-Agent Deep Reinforcement Learning [28]. Signal control agents were employed to establish signal timings at intersections, whereas vehicle routing agents were responsible for selecting vehicle routes [28]. It was demonstrated that the integration of signal control and vehicle routing outperforms controlling signal timings or vehicles' routes alone in enhancing traffic efficiency [28]. This kind of framework could be implemented without CAVs, by advising drivers of routes via their navigation system, but lacks the speed advisory information. Also, it was not explored how drivers ignoring the instructions, and effectively introducing vehicles whose routes were unknown, would affect the systems performance.

One unique framework used Advantage Actor Critic (A2C) to control a simulated isolated four-leg junction in a mixed vehicle scenario [29]. The tests were performed at saturated and oversaturated traffic densities and at a range of penetration rates [29]. It was compared to a well-tuned fixed time plan (with optimal speed advisory in a CAV environment) and actuated signal control [29]. It was found to reduce stop delay significantly under all scenarios and was comparable to other control strategies in queue length [29]. While the RLTC framework used here was tested in an environment containing CAVs, there seems to be no obvious reason why RLTC frameworks like this one, could not be combined with separate GLOSA to create joint control frameworks which can work in scenarios where vehicles only need to be connected, with no requirements placed on autonomy. However, while this approach appears promising, it cannot be stated for certain at this time that this is possible, as no research presently exists that constructs an RLTC-based joint control framework for connected non-autonomous vehicles. Also, this work once again contained a lack of investigation into how the system might perform if the proportion of GLOSA equipped vehicles changes, or if the intended routes of those vehicles are unknown.

At this point, the first objective has been satisfied, as GLOSA, RTCs and existing combinations of the two have been reviewed and assessed. During this work, this method has been identified as showing some promise for the creation of a reliable joint control framework that do not require CAVs and therefore could be implemented with present traffic makeups. However, to design and construct such a framework and fulfil the remaining objective of this thesis, it is required that a review of Reinforcement Learning methods and existing RLTC systems be undertaken.

Specifically, it is critical to the objectives of this thesis to understand the design and decision-making processes of existing RLTC systems. There are typically five defining features: the RL method used for training; the hyperparameter optimisation method used; the control strategy, which comprises the action space (the set of options available to the RLTC system) and the frequency with which decisions are made; the state space, which defines the information that the RLTC systems use to make decisions; and the reward function, which sets the priorities of the RLTC system or the metric to be maximised or minimised.

2.7.1 Reinforcement Learning Methods

Several Reinforcement Learning algorithms exist which have been used for the training of RLTC systems. The most successful of the Q-learning algorithms is DQN [195], which was developed by Deepmind in 2014. This algorithm explores the task and keeps track of the actions it has taken, the states it has visited and the rewards it has received. It then uses this information to learn Q values or action values using one or more neural networks. The earliest implementations of DQN were plagued by stability issues and required a great amount of fine-tuning to achieve satisfactory results. However, since then, it has received near-constant attention from researchers, leading to the creation of multiple variants of the algorithm, including Dueling-DQN and Double-DQN, which have improved performance and stability.

The most popular Policy Optimisation algorithms are (Synchronous) Advantage Actor-Critic Implementation (A2C), Asynchronous Advantage Actor Critic (A3C) [196] and Proximal Policy Optimisation (PPO) [197]. These are Actor Critic algorithms that work by learning both the optimal policy and the value of the current policy separately. This is done with an actor function that takes the current state as input and returns a probability distribution over the available actions and a critic function that takes the current state as well as the selected action and returns the expected value of that action. The critic is updated based on experienced events, and the actor is updated based on estimates of value provided by the critic.

2.7.1.1 Deep Q Networks

In the DQN method, the agent maintains two value function approximators: the policy function $Q_\phi(S, A)$, which takes the parameters ϕ, the current state s and possible actions a as inputs and returns the expected long term reward; and a target function $Q_{\phi_t}(S, A)$, which is employed to improve stability of optimisation.

The neural networks are made up of linear layers which take a vector, $x \in \mathbb{R}^p$, as input and perform $y = wx + b$, where $w \in \mathbb{R}^{q \times p}$ is the weight matrix, $b \in \mathbb{R}^q$ is the bias vector, and $y \in \mathbb{R}^q$ is the output layer.

When training begins, the parameters ϕ, which are the arrays w and b for each layer, are chosen randomly, and the target parameters, ϕ_t, are set equal to ϕ. On each time step, the agent explores the environment using an ϵ-greedy strategy, where an action is chosen either greedily or randomly. Random actions are chosen with probability ϵ. Greedy actions are the actions believed to have the highest value at the time of selection. As time goes on, ϵ is decreased, meaning that the agent will explore more widely early on and narrow down its exploration as it converges to the optimal policy.

The environment then returns a reward R as well as an updated state S'. Now, S, A, R, and S' are stored in memory as experiences. Then M experiences are selected randomly from memory (M is referred to as the batch size), with each experience given an equal chance of selection, and for each experience, the expected long-term return, $\mathcal{R}_i$, is calculated,

$$y_i = R_i + \gamma \max_{A'} Q_{\phi_t}(S'_i, A'), \tag{2}$$

where γ is a hyperparameter termed the discount factor. The policy's parameters, ϕ, are then updated using a gradient descent algorithm with the aim of minimising the loss,

$$L = \frac{1}{M}\sum_{i=1}^{M}(y_i - Q_\phi(S_i, A_i))^2. \tag{3}$$

The target's parameters are updated each time step according to,

$$\phi_t = \tau\phi + (1-\tau)\phi_t, \tag{4}$$

where τ is a smoothing factor parameter.

DQN can also be supplemented by implementing the following methods, which aim to achieve faster and more stable training.

2.7.1.1.1 Double DQN

In the version of DQN outlined above, DQN predicts future returns by finding the maximum Q value for the current state, $\max_{A'} Q_{\phi_t}(S'_i, A')$. However, due to the maximum operator, this value tends to be an overestimate of the actual expected value of future returns, leading to instability in the learning process [198].

It has been shown that using the target network to select the action A' used in the term $Q_{\phi_t}(S'_i, A')$ reduces the number of overestimations [198]. When doing this, the expected long-term return, y_i, is calculated using the target function to estimate the value of taking the greedy action as selected by the policy function in the updated state [199],

$$\mathcal{R}_i = R_i + \gamma Q_{\phi_t}(S'_i, \arg\max_{A'} Q_\phi(S'_i, A')). \tag{5}$$

2.7.1.1.2 Prioritised Experience Replay

Another typical strategy in simple DQN implementation is to sample memories during learning with equal chance. However, this is inefficient as not all memories are helpful in constructing an accurate Q function. The alternative is Prioritised Experience Replay (PER), where memories are sampled more frequently if they lead to a higher loss function [200]. The chance of the ith experience being selected from the memory is defined as,

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}, \tag{6}$$

where α is a hyperparameter which determines how much prioritisation is used, with $\alpha = 0$ corresponding to the uniform cases, and $p_i > 0$ is the priority of transition i. For the first experience, p_1 is set to 1, and all experiences after the first have their priority p_i set to $\max_{j<i} p_j$. Then, after the memory has been sampled each timestep, the priorities of the sampled experiences are updated according to the following equations:

$$p_i = |\delta_i| + \epsilon_{prior} \tag{7}$$

$$\delta_i = \mathcal{R}_i - Q_\phi(S_i, A_i) \tag{8}$$

where ϵ_{prior} is a hyperparameter. However, doing this also introduces a bias, which must be addressed by using the values of $w_i\delta_i$ for the Q-learning update, instead of δ_i, where,

$$w_i = \left(\frac{1}{N} \cdot \frac{1}{P(i)}\right)^\beta, \tag{9}$$

β is a hyperparameter, and N is the size of the memory [200].

2.7.1.1.3 Dueling Network

Dueling Networks work by splitting into two streams in their later layers. The first of these two streams' outputs the advantage of each available action, while the second stream outputs the value of the state. These two values can then be combined to get the Q value for each action that the normal neural network would generate [201]. An example of the structure of a neural network, both with and without the extra layers used for Dueling Networks can be seen below in figure 11.

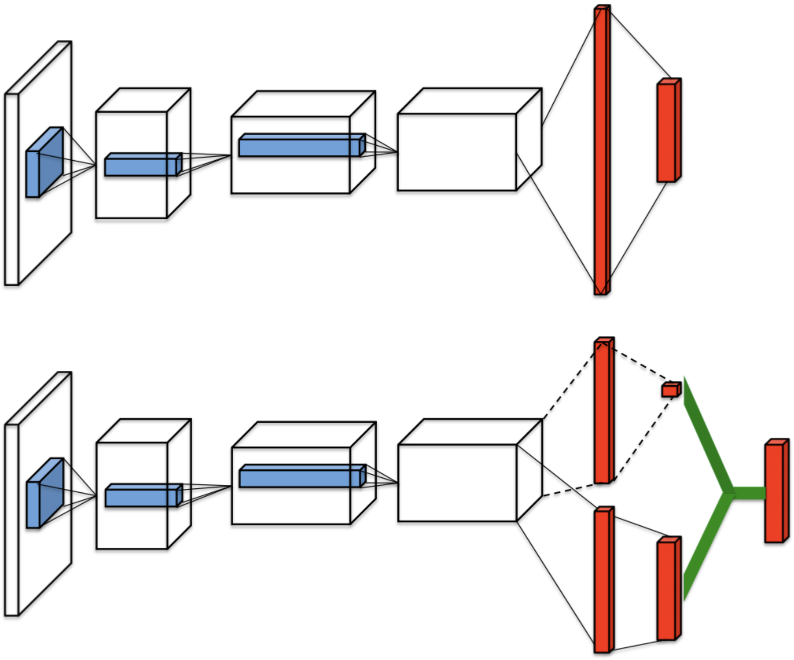

Figure 11: Example of the structure of a Dueling Deep Q Network (bottom) compared to a standard Deep Q Network (top). In the standard Dueling Deep Q Network, the last two layers of nodes in the neural network split to provide estimates for both the value of the current state and the advantage of each available action. While in the standard Deep Q Network the last two layers do not split and only provide a Q value for each state action pair [201].

With this implemented, the network outputs the value of the current state, $V_\phi(s)$ and the advantage of each action in the current state $A_\phi(s, a)$. The Q value for each action is then calculated as,

$$Q_\phi(S, A) = V_\phi(S) + A_\phi(S, A) - \frac{\sum_{A'} A_\phi(S, A')}{|\mathcal{A}|}, \tag{10}$$

where $|\mathcal{A}|$ is the size of the action space. Adding this type of structure to the network means that the network can learn which states are (or are not) valuable without having to learn the effect of each action for each state. This is particularly useful in states where the actions have minimal effects on the environment [201].

2.7.1.1.4 Noisy Networks

Previously, the ϵ-greedy strategy for exploration was outlined. In that method, actions are chosen randomly with probability ϵ and greedily with probability $1 - \epsilon$, where ϵ is decreased as training goes on. This means that training starts with wide-ranging exploration but tends toward exploitation later, with exploration being in theory limited to situations where exploration might lead to an improvement in performance.

However, the noise ϵ-greedy adds is decorrelated and independent from the current state [202], which makes it unlikely to lead to the large-scale behavioural patterns needed for efficient exploration in many environments [203].

The Noisy Network is an alternative approach where learned perturbations of the network weights are used to drive exploration, leading to a consistent, and often complex, state-dependent change in policy over multiple time steps [202]. To implement this, the linear layers of the neural network, represented by $y = wx + b$, are replaced by Noisy layers, represented by

$$y = \left(\mu^w + \sigma^w \odot \epsilon^w\right) x + \left(\mu^b + \sigma^b \odot \epsilon^b\right), \tag{11}$$

where: $\mu^w \in \mathbb{R}^{q\times p}$, $\sigma^w \in \mathbb{R}^{q\times p}$, $\mu^b \in \mathbb{R}^q$, and $\sigma^b \in \mathbb{R}^q$ are to be learned; $\epsilon^w \in \mathbb{R}^{q\times p}$ and $\epsilon^b \in \mathbb{R}^q$ are noise random variables; and $\odot$ is the Hadamard product (element-wise product of matrices) [202].

Here ϵ^w and ϵ^b will be created using factorised Gaussian noise, where two vectors $\omega^p \in \mathbb{R}^p$ and $\omega^q \in \mathbb{R}^q$ are created. Each element of ω^p and ω^q is a random variable distributed according to the unit normal distribution. Then $\epsilon^w_{i,j} = f(\omega^q_i)f(\omega^p_j)$ and $\epsilon^b_i = f(\omega^q_i)$ where $f(x) = sgn(x)\sqrt(|x|)$.

2.7.1.1.5 Categorical DQN

The implementation of DQN previously described outputs only the expected value and advantage of each action in a given state. However, it has been shown that learning the distribution of the value of each state and the advantage of each action leads to better performance [204].

This is done by expanding the value output layer and advantage output layer to produce an approximate distribution of the return instead of the expected return. This is achieved by having the outputs be of size N_{atoms} and $|\mathcal{A}| \times N_{atoms}$, respectively.

Each group of N_{atoms} outputs represents the likelihood values,

$$z_i = V_{min} + (i-1)\frac{V_{max} - V_{min}}{N-1} \text{ for } i \in \{1, ..., N_{atoms}\}, \tag{12}$$

where V_{min}, V_{max} and N_{atoms} are hyperparameters. Using the Softmax function,

$$y_i = \frac{e^{x_i}}{\sum_j e^{x_j}}, \tag{13}$$

the likelihoods output by the network can be converted to probabilities, which then allows for the calculation of the expected return of each action in a given state [204].

2.7.1.1.6 N-Step Learning

One final improvement that can be applied to this implementation of DQN is multi or N-Step Learning. Simple DQN implementations (using the 1-step method) will compare the reward they get for implementing an action to the change in expected return. However, with N-Step Learning, the total reward of several actions is compared to the change in expected return. This is more efficient than the 1-step method due to faster propagation of the reward signal and reduced overestimation problems [205]. In this multi-step version of DQN, the aim is to minimise this alternative formulation of loss,

$$\left(r_t^{(n)} + \gamma^n Q_{\phi_t}(S_{i+n}, \arg\max_{A'} Q_\phi(S_{i+n}, A')) - Q_\phi(S_i, A)\right)^2, \tag{14}$$

where $r_t^{(n)}$ is the Truncated N-Step Return, $r_t^{(n)} = \sum_{k=0}^{n-1} \gamma^k R_{t+k+1}$ [206].

2.7.2 Hyperparameter Optimisation

While Reinforcement Learning algorithms can quickly learn near-optimal strategies, their learning performance is dependent on how well-tuned their hyperparameters are [207]. If the hyperparameters are poorly tuned, the policy can converge too slowly, converge at a local optimum, fail to converge at the global optimum, or diverge away from the global optimum. While tuning can be done by hand, policies are time-consuming to train and therefore it is important to gain an understanding of the hyperparameter optimisation algorithms, which can automate this tuning and improve results, as well as reinforcement learning methods before RLTC systems can be carefully reviewed.

All hyperparameter optimisation algorithms aim to achieve the same thing, finding the values of hyperparameters, x, that minimise the loss of the agent over a testing set, but there are many different approaches to this problem. Examples include Grid Search, Random Search and Bayesian Optimisation.

2.7.2.1 Bayesian Optimisation

Bayesian Optimisation creates an approximate probability distribution of the loss, given hyperparameters as input, and then uses that distribution to decide what values of hyperparameters to try next. It then repeats this process until stopped.

The process starts with a brief random search of the hyperparameter space, trialling with n_0 sets of hyperparameters, $\{x_1, x_2, ..., x_{n_0}\}$, and calculating their loss values, $\{f(x_1), f(x_2), ..., f(x_{n_0})\}$. Once this is complete, the approximate probability distribution of the loss will be constructed. This is done using Gaussian Process Regression, which assumes that,

$$f(x)|f(x_{1:n}) \sim Normal(\mu_n(x_{1:k})m\Sigma_n(x_{1:k}, x_{1:k})), \tag{15}$$

given that,

$$\mu_n(x) = \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x), \tag{16}$$

$$\sigma_n^2(x) = \Sigma_0(x, x) - \Sigma_0(x, x_{1:n})\Sigma_0(x_{1:n}, x_{1:n})^{-1}\Sigma_0(x_{1:n}, x), \tag{17}$$

$$x_{1:n} = [x_1, x_2, ..., x_n], \tag{18}$$

$$f(x_{1:n}) = [f(x_1), f(x_2), ..., f(x_n), \tag{19}$$

$$\mu_0(x_{1:n}) = [\mu_0(x1), ..., \mu_0(x_k)], \tag{20}$$

$$\Sigma_0(x_{1:k}, x_{1:k}) = \begin{bmatrix} \Sigma_0(x_{1:k}, x_{1:k}) & \cdots & \Sigma_0(x_{1:k}, x_{1:k}) \\ \vdots & \ddots & \vdots \\ \Sigma_0(x_{1:k}, x_{1:k}) & \cdots & \Sigma_0(x_{1:k}, x_{1:k}) \end{bmatrix}, \tag{21}$$

where μ and Σ are the Mean and Kernal functions to be chosen by the user.

Once this approximate probability distribution has been created an Acquisition Function is used to select the next hyperparameters to test. Many different Acquisition functions exist; some examples are: upper confidence bound, which selects the hyperparameters corresponding to the lowest confidence bound of the loss value; probability of improvement, which selects the hyperparameters with the highest chance of scoring a lower amount of loss; and expected improvement, which selects the hyperparameters with the highest expected decrease in loss.

For more details on Bayesian Optimisation, Gaussian Process Regression, Acquisition functions, Mean and Kernal functions, as well as more advanced techniques, including running multiple function evaluations in parallel, multi-fidelity and multi-information source optimisation, expensive-to-evaluate constraints, random environmental conditions, multi-task Bayesian optimisation, and the inclusion of derivative information, see [208; 209; 210].

2.7.3 Control Strategy

Existing reinforcement learning ATC frameworks can mostly be split into two types: the first type allows the next phase to be selected, and the second allows frameworks to select how long the current phase will continue.

Although frameworks in the first category can choose which phases will follow the current phase, the time between these actions is fixed, usually between 1 and 10 seconds [211]. Setting the time between actions to be shorter allows for more flexible control strategies; however, it will also reduce the available reliable future signal plans as the traffic lights will behave less predictably with shorter times between actions. In the case of short times between actions, limits must be included on the minimum length of phases. As there are a finite number of actions available to an agent in this framework, Deep Q Networks (DQN) is usually employed as the reinforcement learning method.

Frameworks in the second category can choose the length of the current phase but cannot change the order of the phases. These frameworks can be further split according to how the phase length is selected. In most frameworks in this category, the agent decides between either extending or ending the current phase every second or few seconds [213], while a rarer approach has the agent choose how long a phase will last at the beginning of the phase [215]. Some frameworks can skip a phase by assigning it no time [217].

Having the agent choose to extend or end phases allows for more flexible control strategies, which should perform better in the absence of GLOSA. However, this also reduces the available reliable future signal plans as the traffic lights will behave less predictably with shorter times between actions. Also, limits must be included on the minimum length of phases [219].

Having the agent decide how long the phase will last is more predictable and does not require limits, as phase lengths considered too short are not given as actions. These can also have either finite or continuous action spaces. With a finite action space, the agent is given a set list of options for the duration of the next phase, usually 5 or 10 seconds apart and ranging between 5 to 90 seconds. With a continuous action space, the agent will be able to select any amount of time for the phase within a preset range, allowing for a lot of flexibility. However, overall, this approach is less flexible than choosing between extending or ending the phase, but it does supply more reliable future signal plans, up to two phases worth if skipping phases is not allowed.

A third, less common, type of framework selects an operating mode or a full cycle worth of signal plans. This can be done in a number of ways. The simplest frameworks in this category have a number of preplanned operating modes or cycle plans, which the agent can choose from, typically at the beginning of each cycle [221]. Other frameworks make small adjustments (typically up to 5 to 10 seconds per phase) to the one or each phase in the cycle plan at the end of each cycle [223]. A framework that selects a cycle worth of signal plans would supply a lot of future signal plans but would need to be able to frequently update the cycle plan and be able to construct a broad range of signal plans to be flexible.

The final type of framework combines the first two types by having the agent select a phase and then its duration, either by choosing between extend or end [225], or by selecting its duration at the beginning [227]. Thanks to not having a fixed phase order or length, these frameworks are very flexible but do not allow for a large amount of future signal plans.

2.7.4 State Space

In the published literature, many different state spaces have been designed and tested, including Queue Length [228], Vehicle Position [229], Distance to the nearest vehicle at each approach [213], Vehicle Density [229], Speed [230], Delay [231], Vehicle Waiting Time [232], Vehicle Emissions [231], Red Green Yellow Timing [211], Yellow Phase Indicator [213], Current Traffic Phase [233], Current Time [213], Raw Pixel Snapshot [234], Number of Waiting Pedestrians [235], Distance Between the Lead Car and the Stop Line [236].

Queue length, or number of stopped vehicles, has been a very commonly used state space, appearing in eighty-five of the papers surveyed. However, its popularity has decreased in recent times, with it appearing in around 37% of papers after 2017, as opposed to 64% before 2018. Queue length is typically defined as the number of slow-moving (less than $0.1m/s$) vehicles. When used as part of a state space, the queue lengths for each approach lane are usually arranged into a vector [237]. However, in some earlier Q-Learning RLTC systems, the queue lengths were expressed as a level of congestion (typically a 0 to 3 scale) [238] or stopped counting above a certain number of queued vehicles [239], to reduce state space dimensionality. Its common use may be partly attributed to the ease with which this information can be approximated in real-world scenarios by placing loop detectors at the stop line and shortly upstream [240]. It also represents a metric that is commonly optimised for RTCs. However, it does not contain information about upstream vehicles, and so it does not help agents react to approaching vehicles or platoons. This can be addressed by adding further metrics to the state space, but this increases dimensionality, training time, and training cost.

A similar metric is approaching vehicles which was used in fifty-three of the surveyed papers and appeared in 37% of all papers published after 2017. Here, all vehicles on incoming lanes within a given distance of the junction are counted (moving or halted). Just like Queue Length, this state space can be expressed as a vector or in a more compressed method to reduce state space dimensionality. It can also be approximated with a loop detector.

However, unlike Queue Length, which is effectively blind to the traffic on lanes that are currently being served, the Approaching Vehicles metric is not affected by the current phase. This means it should be able to make more informed decisions about whether to continue a phase.

However, the downside to Approaching Vehicles is that it can waste green time by attempting to serve vehicles that are too far away if the detection range is too large. Similarly, It can also fail to detect and respond to oncoming platoons if the detection range is too short. A balance can be achieved by discretising the incoming lanes and counting the vehicles inside each cell [241]

These state spaces can also be combined with information about emergency vehicles [242], public transport or other higher priority traffic [243], as well as the queue lengths or number of vehicles on downstream links, in multi-agent connected network settings [244].
Other less frequently employed alternatives include: Occupancy [245], whether or not there are vehicles queueing at the stop line; Traffic Flow [102], the number of vehicles to use each lane over a given time period; Density [246], the number of vehicles per a given distance on incoming lanes; and Gap Between Vehicles [245], the gap between the last two vehicles to reach a loop detector.

Often, information about the previous phase [247] and how long it has been going [248] is included in the state space. Both help the agent distinguish between a lane that has light traffic conditions and a lane where the previously queueing vehicles have begun moving and become spread out during a green phase, therefore allowing the agent to make better decisions about whether to continue a phase. The latter also helps the agent avoid the negative behaviour of keeping vehicles on side roads stationary for extended periods of time when paired with the correct reward functions. An extension to the latter is the time since each phase or the time since each lane was served [249].

In multi-agent connected network settings this state space can be expanded to include the signal states of surrounding junctions [250]. This extension to the state space can provide the agent with information about upstream and downstream traffic conditions, allowing it to better serve incoming vehicles and avoid becoming backed up.

Another type of state information that achieves a similar effect as phase information is waiting time. This is typically expressed for each lane as the cumulative delay of every vehicle in that lane [235] but could also expressed as the waiting time of the lead vehicle in that lane [248]. This state space can also be expanded to pedestrians [251]

As reinforcement learning methods have improved, their capability to manage large state spaces has increased. With the introduction of the Deep Q Networks (DQN) method, it became possible to use state spaces like discrete traffic state encoding (DTSE) [229] and Raw Pixel Snapshots (RPS) [252].

The former creates a tensor representing things like vehicle positions, speed, acceleration, phase, and waiting time. Each layer of the tensor contains a different type of information. Each row inside a layer pertains to a different incoming lane (sometimes outgoing lanes are included [235]), and each element in a row pertains to a section of that lane, which is slightly longer than a car's length, usually called a cell.

Position is typically expressed by putting a one in all cells occupied by a vehicle and a zero in other cells. Speed, acceleration and waiting time are expressed in a comparable manner, with elements relating to the occupied cells being set equal to the Speed, Acceleration or Waiting Time of the vehicle. Phase is typically expressed by setting all the elements in all the rows that correspond to lanes currently receiving green time to one, while all other cells are set to 0.

The biggest issue with DTSE is the scale of the data collection. To get a complete and correct DTSE representation, every vehicle would need to broadcast its position and speed, and there would have to be a very low rate of packet loss.

The alternative is an RPS state space, where the state is described by one or more images. In simulator studies, this is typically a screenshot of the simulator GUI. Whereas in the real world, this would be live camera feeds of the junction (or junctions) and approach roads. A Convolutional Neural Network can then process these images. Unlike DTSE, RPS can be set up with far less infrastructure and without vehicles having to broadcast any information. However, agents using RPS are very time-consuming to train as this state space has extremely high dimensionality, and lots of the information in an RPS state space is of no value or just noise. Because of this, they are still quite rare.

2.7.5 Reward Functions

Many reward functions have been designed and trialled in published literature. The metrics they are based on typically include: Queue Size [253], Delay [213], Vehicle Travel Time [254], Vehicle Waiting Time [231], Approaching Vehicles [219], Intersection Throughput [217], Fuel or Energy Consumption [255], Vehicle Emissions [256], Penalty for Emergency Stops [257], and Phase Changes [217].

Queue Length is the most used reward function metric, appearing in 48% of all surveyed papers. Once again, its common use may be partly attributed to the ease with which this information can be approximated in real-world scenarios by placing loop detectors at the stop line and shortly upstream.

As in the state space, it is typically measured as the number of slow-moving (less than $0.1m/s$) vehicles. There are three ways in which queue length might be incorporated into the reward function. The first is to penalise the agent for every queuing vehicle at every decision step [238]. The second is to reward the agent for reducing the queue length by applying a reward for every vehicle it manages to clear from the queues while penalising it for increasing the queue length by applying a penalty whenever a car begins to queue [228]. These two can be used to create agents that minimise the number of queueing vehicles at any given time.

However, the third way is to reward the agent for keeping all the queues close to even or punish the agent if a certain approach backs up far more than others [258]. The aim of this method is to ensure that vehicles on side roads do not spend a disproportionately long time stuck at a red light, as agents trained to try to minimise the total queue length can decide against releasing side road traffic because of the short-term queues stopping the main road creates.

The second, third, fourth and fifth most common reward function metrics are delay, waiting time, vehicles served, and Approaching Vehicles, which appear in 25%, 24%, 20% and 16% of all surveyed papers. Waiting time is the amount of time vehicles spend stopped (moving at less than $0.1m/s$), while the delay is the time lost when a vehicle is travelling below the speed limit of the road, and Approaching Vehicles is the number of vehicles on incoming lanes. Vehicles Served is the number of vehicles that have passed through the junction since the last action began.

Waiting time, Delay and Approaching Vehicles can be applied in the same ways as Queue Length, with some modification, but Vehicles Served can only be applied as a positive reward when vehicles pass the stop line.

Waiting Time can be estimated by using loop detectors to estimate Queue Length [240] but, as it cannot be known exactly when the car entered and left the queue, doing this adds a further level of error to the estimation. In contrast, Delay can be estimated with loop detectors [259]. Approaching Vehicles and Vehicles Served are easier to measure than Queue Length.

While the reward metrics we have discussed so far are typically aimed at improving overall performance, some reward metrics have more specific goals. One such reward metric is Phase Change, which involves giving a small penalty to the agent whenever the phase changes, with the aim of stopping a common light-flickering behaviour of RLTC systems, where the agent changes the phase too often and wastes a large amount of time switching between phases. Another is Collisions or Emergency Stops, which aims to reduce the number of times the agent causes an accident or narrow escape by applying a large penalty.

An important note is that as a reinforcement learning agent is trying to find a way to maximise its reward, it may happen across a solution (or local maximum) that was not intended by the designer and performs poorly overall but achieves a high return from the reward function. One common strategy for avoiding these local maximums is to create reward functions that are weighted sums of many metrics, as this ensures that the agent cannot neglect some metrics and overall performance to maximise one metric.

2.8 An Overview of the Literature and the Gap In Knowledge

The first objective of this thesis is to fully review the state of the art in GLOSA, RTCs, and existing combinations of the two, to build a complete understanding the existing approaching and their limitations. With the conclusion of this literature review, this has been achieved, and a list of knowns and unknowns can be constructed on the topics of RTC systems (including RLTC systems), GLOSA, and combinations of both.

As both RTCs and GLOSA are well-researched, their benefits are well-known. Early attempts to implement these systems together were unsuccessful, but some strategies have been found that allow these systems to be combined. These strategies have centred on three approaches: predicting future signal plans to activate GLOSA, using dynamic programming or like schedule the arrival times of vehicles approaching the junction, and combining a RLTC system with either GLOSA or reinforcement learning controlled vehicles.

The latter type of solution has been shown to be effective for CAVs and may also be effective for connected non-autonomous vehicles. However, no research exists that constructs a RLTC-based joint control framework for connected non-autonomous vehicles. Also, none of the existing research into RLTC-based joint control frameworks examines the effects of differences between the penetration rates of evaluation and training or of vehicle route information being unavailable. This satisfies the first objective.

The above can be rewritten into three gaps in knowledge. Firstly, it is unknown how a joint control framework that uses reinforcement learning traffic control and is designed for non-autonomous vehicles will perform. Secondly, it is unknown how a joint control framework that uses reinforcement learning traffic control will perform if the penetration rates of GLOSA at the site of deployment are different from those in the training scenario. Thirdly, it is unknown how a joint control framework that uses reinforcement learning traffic control will perform if route information is unavailable (due to communication issues, or a wish for privacy from road users).

3 Methodology

To address these gaps in knowledge and achieve the aims and objectives of this thesis, three phases of research will be required.

3.1 Phase 1: Isolated Junctions

To begin addressing these gaps in knowledge and the aims and objectives of this thesis, a better understanding of how RLTC based joint control frameworks, designed for non-autonomous vehicles, will behave on isolated junctions is needed. To get that better understanding, experimental results must be gathered which demonstrate the performance of such a joint control framework on an isolated junction.

However, getting that experimental data requires that a joint control framework, for non-autonomous vehicles that uses RLTC, be designed and have its performance validated. This design work must include the selection of a reinforcement learning method, a neural network architecture, a state space, an action space, and a reward function, as well as the tuning of some hyper-parameters. A suitable test bed must also be found or built, as this will be needed to allow for: the definition of some details of the joint control framework, the validation the performance of the joint control framework, and the conducting of experiments to gain the required experimental data. Furthermore, GLOSA must be implemented on the test bed, so that the required experimental data can be obtained, and benchmarks to compare the joint control framework against must be chosen. The design and selection of these details will be covered in chapter 4.

Once the framework and test bed design work is complete, it remains to get the experimental results, which will require some experimental design work. Full design details will be discussed, before presentation of results and discussion of conclusions, at the beginning of chapter 5. However, as the experimental data required must include demonstrating that the joint control framework out performs benchmark systems, quantifying how the joint control framework will perform if the penetration rates of GLOSA at the site of deployment are different from those in the training scenario, and training the RLTC system, it is clear that two experiments will be needed.

The first experiment of this chapter will be to train a number of RLTC agents at a range of GLOSA penetration rates (called training penetration rates or TPR). Training an agent with a TPR of 0% is equivalent to training a RLTC system without GLOSA. Then, the agents will be evaluated with the evaluation penetration rates (EPR) set equal to the TPR and compared to the benchmark. During this and all other evaluations, the number of stops, waiting times, vehicle speeds, and junction entry speeds will be recorded. This step will train the RLTC system for use in the second experiment, validate the performance of the joint control framework against the benchmark systems, and provide the results required to address the first gap in knowledge for isolated junction scenarios.

The second experiment will evaluate the agents trained in the first experiment in scenarios where the EPR is different from the TPR on the isolated junction testbed. This step will provide the results required to address the second gap in knowledge for isolated junction scenarios.

3.2 Phase 2: Arterial Flow

After this, a better understanding of how RLTC based joint control frameworks, designed for non-autonomous vehicles, behave on isolated junctions will have been obtained. However, to fully address the stated gaps in knowledge, as well as the aims and objectives of this thesis, further work must be done and experimental data gathered. Chiefly, knowledge must be expanded to a greater number of scenarios.

This can primarily be achieved by gaining experimental data for the performance of the joint control framework on an arterial flow. This will also allow for the creation of understanding of how the joint control framework will perform if route information is unavailable. Getting this new experimental data requires that a suitable arterial flow test bed be found or built. Also, the joint control framework will need to undergo modification for use on the new testbed and have its performance validated on that testbed. Furthermore, GLOSA must be implemented on the test bed, so that the required experimental data can be obtained, and benchmarks to compare the joint control framework against and validate its performance must again be chosen. The design and selection of these details will be covered in chapter 6.

Once the framework and test bed design work is complete, it remains to get the experimental results, which will require some experimental design work. Full design details will be discussed, before presentation of results and discussion of research limitations, at the beginning of chapter 7.

However, the experimental data must demonstrate that the joint control framework outperforms benchmark systems and quantify how the joint control framework will perform if the penetration rates of GLOSA at the site of deployment are different from those in the training scenario. Also, the experiments must include the training of the RLTC system. Therefore, two experiments, numbered 3 and 4, will be needed.

In experiment three, a number of groups of cooperative RLTC agents will be trained on the arterial network testbed at a range of TPRs. Then, the agents will be evaluated on an arterial network testbed with the EPR set equal to the TPR and compared to a fixed time benchmark. This step will provide the results required to address the first gap in knowledge for arterial network scenarios.

In experiment four, the reinforcement learning traffic control system will be evaluated in scenarios where the EPR is different from the TPR on the arterial network testbed. This step will provide the results required to address the second gap in knowledge for arterial network scenarios.

3.2.1 Phase 2.1: Arterial Flow with Unknown Routes

However, experimental data is also needed to assess how a joint control framework that uses reinforcement learning traffic control will perform if vehicles don't provide their routes to the joint control framework in advance, meaning junctions will not know a vehicle is approaching until after it has cleared the previous junction. Also, the experiments must include the training of the RLTC system for this scenario. Therefore, two further experiments, numbered 5 and 6, will be needed.

In experiment five, a number of groups of cooperative RLTC agents will be trained on the arterial network testbed at a range of TPRs without vehicles providing any information about their intended route. Then, the agents will be evaluated on an arterial network testbed with the EPR set equal to the TPR and compared to a fixed time benchmark. This step will provide the results required to address the third gap in knowledge.

In experiment six, the reinforcement learning traffic control system will be evaluated in scenarios where the EPR is different from the TPR on the arterial network testbed without vehicles providing any information about their intended route. This step will provide the results required to address the second gap in knowledge for arterial network scenarios with unknown vehicle routes.

3.3 Phase 3: Final Conclusions and Recommendations

Once this experimental data has been obtained, the gaps in knowledge will have been addressed. However, the last objective, which was to understand the limitations of this research and the direction future research should take, and therefore also the aim of this thesis will remain uncompleted. To address this, a concluding chapter is included containing an overview of the key results obtained, as well as a discussion on the limitations of this research conducted in this thesis. Based on these results and limitations, recommendations will be made for the direction of future research.

3.4 Overview

See table 1 for the makeup of each experiment and figure 12 for a pre-requisite chart, showing the tasks to be completed and the order in which they must be undertaken.

		Training	Evaluation	
			$TPR = EPR$	$TPR \neq EPR$
Isolated Junction		Experiment 1		Experiment 2
Arterial	Known Routes	Experiment 3		Experiment 4
	Unknown Routes	Experiment 5		Experiment 6

Table 1: Experiments to be conducted

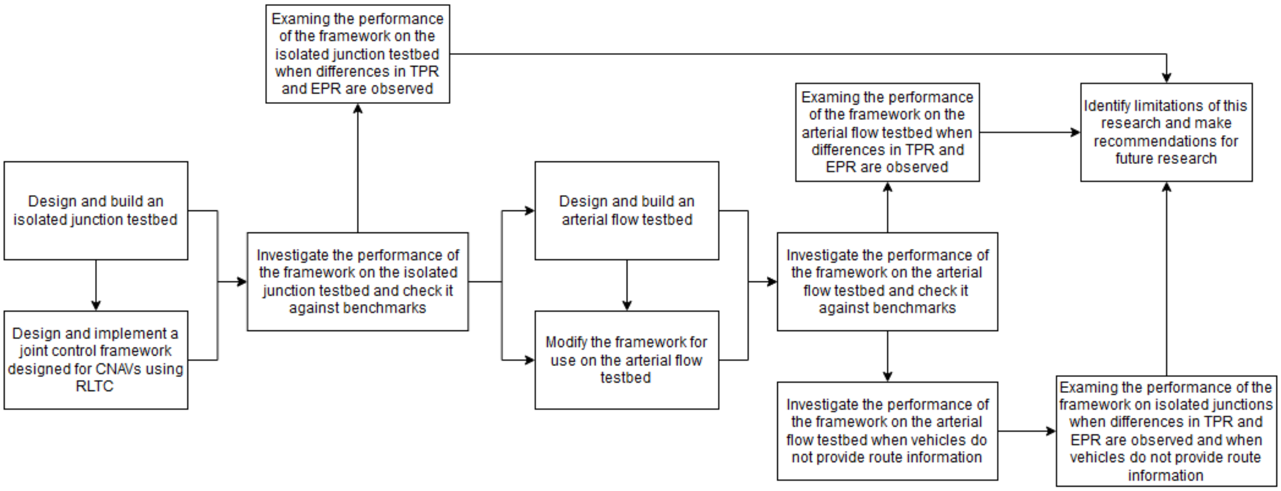
Examing the performance of the framework on the isolated junction testbed when differences in TPR and EPR are observed
Design and build an isolated junction testbed
Design and implement a joint control framework designed for CNAVs using RLTC
Investigate the performance of the framework on the isolated junction testbed and check it against benchmarks
Design and build an arterial flow testbed
Modify the framework for use on the arterial flow testbed
Examing the performance of the framework on the arterial flow testbed when differences in TPR and EPR are observed
Investigate the performance of the framework on the arterial flow testbed and check it against benchmarks
Investigate the performance of the framework on the arterial flow testbed when vehicles do not provide route information
Identify limitations of this research and make recommendations for future research
Examing the performance of the framework on isolated junctions when differences in TPR and EPR are observed and when vehicles do not provide route information

Figure 12: pre-requisite chart, detailing the tasks to be completed and the order in which they will be undertaken.

4 Combining RLTC and GLOSA on Isolated Junction

As stated in the methodology, the next required step, in the process of addressing the identified gaps in knowledge and achieving the stated aims and objectives of this thesis, is to get a better understanding of how a RLTC based joint control framework designed for non-autonomous vehicles will behave on isolated junctions which can only be done by obtaining experimental results which demonstrate the performance of such a joint control framework on an isolated junction.

Getting that experimental data requires: the finding or building of an isolated junction testbed, the design and construction of a joint control framework for non-autonomous vehicles that uses reinforcement learning traffic control, the implementation of GLOSA on the test bed, and the choosing of benchmarks to allow the performance of the joint control framework to be validated.

Therefore this chapter will be split into four sections.

- Testbed design which will include discussion of field tests versus simulations, microsimulation packages, isolated junction layout, and traffic scenarios,
- RLTC implementation which will include discussion of: RL algorithms, state and action spaces, reward functions, neural network architecture, and hyperparameter tuning,
- GLOSA implementation,
- Traffic Control Benchmarks.

4.1 Testbed

In this section, an isolated junction testbed will be selected and described so that it can be used as part of experiments designed to produce experimental data describing the performance and behaviour of joint control frameworks on isolated junctions.

During the selection of the isolated junction testbed, there are three things to consider. Firstly, it must be decided whether to use field tests or simulations and if simulations are used it must be decided what software package is used. Secondly, the junction layout to be used during the experiments. And thirdly, the levels of traffic to be used during the experiments.

4.1.1 Microscopic Traffic Simulators

In this section, the approaches that could have been used to perform this research are discussed. This will begin with a justification for performing the experiments in microscopic simulations rather than field tests, mesoscopic simulations, or macroscopic simulations before moving to a discussion of several microscopic traffic simulators and the identification of one that best meets the needs of these experiments.

The main advantage of a field test is that the results of the test will be closer to the real-world effects than those results achieved through simulation testing, which relies on models of the environment that may make simplifying or incorrect assumptions which could lead to results that are not transferable to the real world. However, field tests involving RTCs and GLOSA are limited in size by the cost of equipment and the need to ensure the health and safety of road users during their interaction with the technology.

By comparison, simulations are cheap to run and can be run multiple times with different variables and scenarios to achieve results in a wider range of situations. While simulations cannot generate results that are as accurate as field tests, the results will be close to the results of a field test so long as the assumptions of any models used are reasonable.

Due to the limitations of field trials and the advantages of using simulations, simulation was selected as the most appropriate method to perform the experiments.

There are three types of simulation that can be done to evaluate road network performance, namely microscopic, mesoscopic, and macroscopic. Microscopic simulations consider the behaviour of individual vehicles in the road network, while macroscopic simulations consider the properties of the road network as a whole or in zones. Mesoscopic simulations achieve a level of detail between microscopic and macroscopic simulations by considering small groups of vehicles whose behaviour is assumed to be homogeneous. Mesoscopic simulations may, for example, report results on a per-lane basis.

Microscopic simulations provide the most detailed results but are also the most computationally intensive to run, while macroscopic simulations provide the least detailed results but are also the least computationally intensive to run. Mesoscopic simulations are the middle ground on both metrics, providing more detailed results than macroscopic simulations and at a lower computational cost than microscopic simulations.

As GLOSA interfaces with each simulated vehicle individually, achieving the goals set out previously requires the use of microscopic simulations where the dynamics of each vehicle are considered. Furthermore, the availability of hardware resources meant the increased computational cost of microscopic simulations could be mitigated.

There are many microsimulation packages available for modelling traffic networks. To ensure the results are dependable, comparable to other work in the field and have both required and modern features, certain criteria were placed on the choice of microsimulation software packages used in this research. Firstly, to ensure that the software has all the features that would be expected in a modern microsimulation software package, it is required to be actively developed. Secondly, the software must have been recently and widely used in reviewed and published RLTC papers and GLOSA papers. Thirdly, the software must give the user the ability to control traffic signals and influence the actions of vehicles through an API so that a RLTC system, and if required an external GLOSA, can be implemented. Finally, the software must have accessible documentation.

Aimsun[212], SUMO[214], and VISSIM[216] were all identified as suitable software packages, for this stage of the research, which met these criteria and were suitable. TSIS-CORSIM, MATSim and Paramics Discovery were also considered but were judged not to have met the second criteria due to a lack of published research relating to GLOSA over the last five years.

As Aimsun[212], SUMO[214], and VISSIM[216] were all suitable the final decision was made for secondary reasons. Aimsun 8.2.1 was selected for these experiments as a software license and learning resources were readily available.

4.1.2 Junction Layout

With simulation software selected, the next task is to design and build an isolated junction which must meet two criteria. Firstly, the junction must be typical of real-world signalised junctions. And secondly, the approach roads of the junction need to have sufficient length such that they do not reduce the amount of time vehicles have to react the future signal plans.

The selected junction was a replica of the signal-controlled crossroads typical of urban roads for left-hand drive vehicles used in “A Reinforcement Learning Approach for Intelligent Traffic Signal Control at Urban Intersections” by Mengyu Guo et al. [228] which had been modified to have approach roads of sufficient length. This was selected as it met the criteria and would allow for the joint control framework to be benchmarked against the RLTC system demonstrated by Mengyu Guo et al. [228]. A full description of the junction follows.

Pedestrian movements are not being considered as they could be performed alongside vehicle movements. Each entrance has three lanes: the rightmost lane functions as a right turn, the middle lane functions as a forward lane and the leftmost lane functions as a left turn and forward lane. All exits have two lanes. The speed limit is set to 13.42m/s (30mph), which is the typical speed limit for urban areas in the UK. The roads leading to and from the junction are 150m long so that the RLTC has at least 10 seconds warning of any vehicle that approaches the junction, which is the length of time a phase will last in the framework (see section 4.2.3). This was seen as sufficiently short, to allow for flexible traffic control decisions, and sufficiently long, to allow for significant future signal plans for the activation of GLOSA. While with a greater abundance or future signal plans it would be more optimal to have the approach roads be closer to 300m in length, in this case the approach roads need only have sufficient length such that they do not reduce the amount of time vehicles have to react the future signal plans and increasing the length further would therefore produce no additional benefit. The layout of the junction allows for the signal controller to have access to eight phases. The first four phases give green time to all lanes on a single approach. The next two phases give green time to opposite approaches, excluding the right turn lane. The final two phases give green time to opposing right-turn lanes.

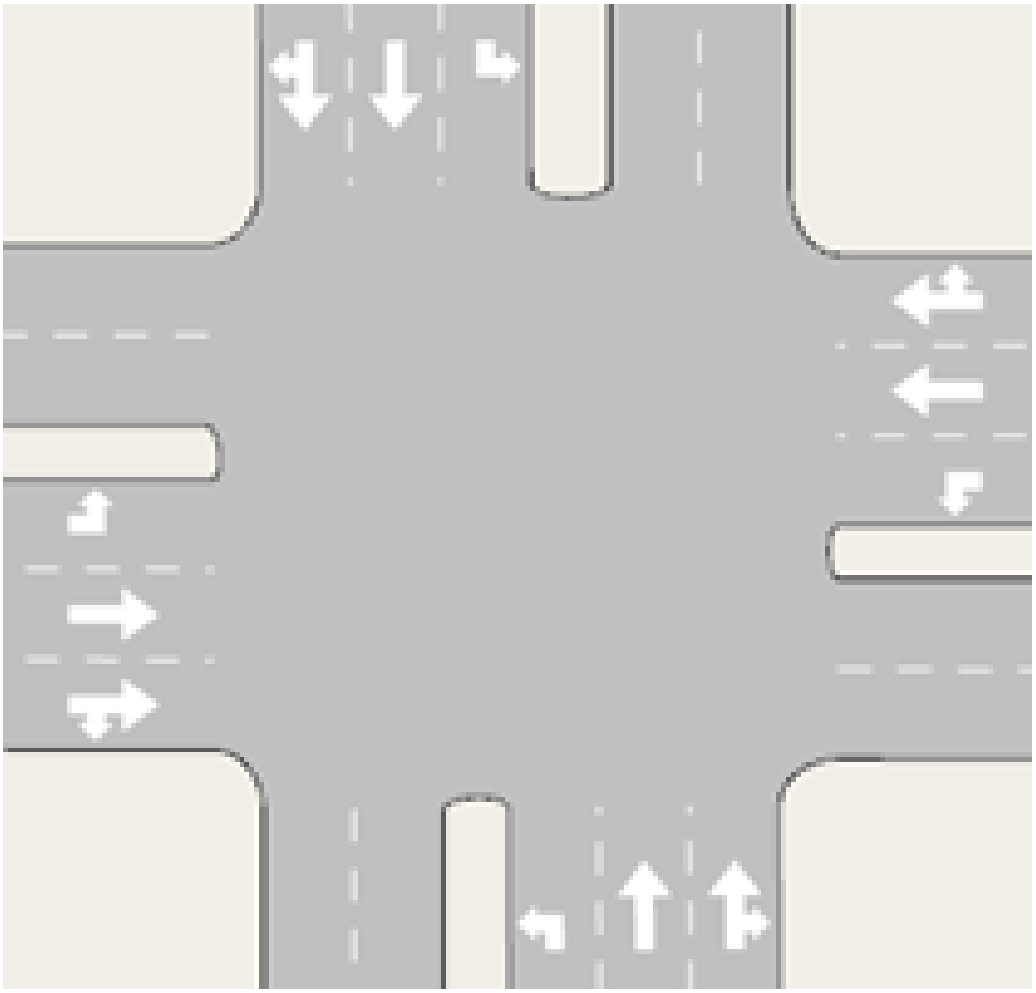

Figure 13: Isolated Junction Layout.

4.1.3 Traffic Scenarios

Two traffic scenarios have been selected for use in the testbed, with a group of agents being trained for each. The first matches traffic flows taken from Mengyu Guo et al. [228] and was selected because it allows for a direct comparison of results between the joint control frameworks used here, and the RLTC system designed by Mengyu Guo et al. [228]. See Table 2 for a demand matrix.

	West	East	North	South
West		360	180	36
East	360		36	180
North	36	90		180
South	90	36	180	

Table 2: Demand matrix expressed in vehicles per hour, 55% scenario.

It was calculated that the saturation of this was approximately 55%, using a combination of Kimber et al.'s formula for the saturation flow of a traffic stream [218], Webster's formulas for the optimal cycle length and equisaturation [220], and the following formula for degree of saturation,

$$\rho = \max_{i=0,1,\ldots,n} \frac{cq_i}{g_i s_i}, \tag{22}$$

where c is the cycle time calculated using Webster's formula for optimal cycle length, q_i is the rate of arrival of vehicles of the i^{th} traffic stream, s_i is the saturation flow of the i^{th} traffic stream calculated using Kimber et al.'s formula, and g_i is the effective green time of the i^{th} traffic stream calculated using Webster's formula for equisaturation. While this is a relatively lower traffic flow, it was noted in the review that, with GLOSA enabled, higher traffic densities led to decreases in GLOSA performance as other vehicles travelling towards the junction and vehicles queuing at the junction tend to block GLOSA-equipped vehicles from following the optimal path. Therefore, it made sense to begin testing at lower traffic flows.

The second scenario was created by multiplying the traffic flows in the 55% scenario such that equation 22 gave a saturation of 70%. As with the first, the second set of traffic flows was also a lower traffic flow of 70% as higher traffic densities have been shown to lead to decreases in GLOSA performance. See Table 3 for a demand matrix.

	West	East	North	South
West		594	297	59.4
East	594		59.4	297
North	59.4	149		297
South	149	59.4	297	

Table 3: Demand matrix expressed in vehicles per hour, 70% scenario.

Setting up the traffic density scenarios such that a given agent only ever sees one of them, instead of training and evaluating the same agent across many scenarios, is a limitation of the research as in the real world, RTCs would see all manner of scenarios. However, doing this should not affect the results that are achieved, only reduce the complexity of and the time needed for training. Furthermore, it should be possible for future work to include multiple scenarios in the training of a single agent to create an agent that can handle multiple scenarios.

Vehicles other than cars (Emergency Vehicles, Public Transport, HGVs, etc.) have been excluded from the simulation, as their introduction would introduce more variables and greater uncertainty in the car following model and GLOSA implementation. This is especially true for HGVs, Emergency Vehicles, and Public Transport, which can have massive effects on the behaviours of other vehicles around them and therefore could heavily distort the results. This in turn could increase the difficulty of producing a working framework. Pedestrians have also been excluded as the phases allow them to navigate the junction without impact during normal operation.

4.2 RLTC Implementation

In this section, a RLTC system will be designed and described so that it can be used as part of experiments designed to produce experimental data describing the performance and behaviour of joint control frameworks on isolated junctions.

During the design of the RLTC system, there are six things which must be considered. Firstly, a reinforcement learning algorithm must be selected. Secondly, a state space must be designed. Thirdly, an action space must be designed. Fourthly, a reward function must be designed. Fifthly, a neural network must be designed. Finally, hyperparameters must be selected.

4.2.1 Reinforcement Learning Algorithm

When deciding which algorithm to use for this research, several criteria were decided upon. Firstly, the algorithm must support discrete action spaces and high-dimensionality state spaces. Secondly, the algorithm must be actively developed by researchers in the field of reinforcement learning and used in existing RLTC research. Two types of algorithms were identified that met these criteria: Q-learning algorithms and Policy Optimisation algorithms.

The most successful of the Q-learning algorithms is DQN [195], which was developed by Deepmind in 2014. This algorithm explores the task and keeps track of the actions it has taken, the states it has visited and the rewards it has received. It then uses this information to learn Q values or action values using one or more neural networks. The earliest implementations of DQN were plagued by stability issues and required a great amount of fine-tuning to achieve satisfactory results. However, since then, it has received near-constant attention from researchers, leading to the creation of multiple variants of the algorithm, including Dueling-DQN and Double-DQN, which have improved performance and stability.

The most popular Policy Optimisation algorithms are (Synchronous) Advantage Actor-Critic Implementation (A2C), Asynchronous Advantage Actor Critic (A3C) [196] and Proximal Policy Optimisation (PPO) [197]. These algorithms are Actor Critic algorithms that work by learning both the optimal policy and the value of the current policy separately. This is done with an actor function that takes the current state as input and returns a probability distribution over the available actions and a critic function that takes the current state as well as the selected action and returns the expected value of that action. The critic is updated based on experienced events, and the actor is updated based on estimates of value provided by the critic.

From these two, DQN was selected for these experiments as it is currently the most widespread algorithm in RLTC research.

4.2.2 State Space

The main requirement of the state space for this application is that it must convey enough information to the reinforcement learning agent to allow it to take account of how its actions effect the GLOSA instructions of approaching vehicles. In particular, it is important that the agent is aware of how many vehicles are likely to reach each stop line or the back of each queue over the course of its next action. Also, any queues would stop vehicles from being able to follow GLOSA suggestions, it is important that the state space include details about how many and where vehicles are queueing.

Other criteria for the state space were that it was preferable to use metrics that would be easy to obtain directly or estimate accurately, and that between all metrics they must convey enough information to the reinforcement learning agent to allow it to learn efficient traffic control.

With this in mind, the state space was designed to be fifty-four values arranged into a vector. The first twelve values are equal to the number of slow-moving vehicles (those with speeds less than 4.47 m/s or 10 mph) in each lane. The second twelve are equal to the number of vehicles that are within 94m (the distance travelled in 7 seconds at the road's speed limit) of the junction in each lane. The third twelve are equal to the number of vehicles that are within 134m (the distance travelled in 10 seconds at the road's speed limit) of the junction in each lane. The fourth twelve are equal to the number of vehicles in each lane. The final eight values describe which phase was previously active.

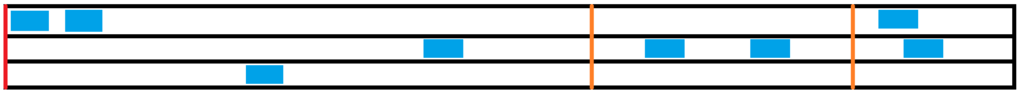

Figure 14: In this diagram three lanes approaching the stop line of the junction (red) are depicted. Also marked are the 94m and 134m lines (orange). Each lane is described by four values. Firstly, the number of slow-moving vehicles (2,0,0 top to bottom). Then the number of vehicles that are within 94m (2,1,1), and 134m (2,3,1) of the stop line. Then finally the number of vehicles in the lane, total (3,4,1).

If phase one was active last, then the first of the eight values is set to 1. If phase two was active last, then the second of the eight values is set to 1, and so on. The remaining seven values are set to 0.

The first four sets of twelve values allow this state space to meet the first criteria, both conveying which approaches are blocked by queues, and how many vehicles could be served by the junction and GLOSA on upcoming phases. However, as issue discovered in early versions of this framework was phase switching. As the state space did not contain information about which phases had come before, the agent could not identify when it was changing phase. This meant that it would almost always switch phases to serve the largest queue and waste a lot of time with the signals being in transitional phases. This behaviour invariably leads to the junction backing up. Therefore, the previous phase information was added so that the state space met the criteria set out above.

4.2.3 Action Space

In order for the RLTC system to be applicable to this research, it must allow enough future signal plans for GLOSA activation. Therefore, the agent selects a new action (phase) whenever the previous phase ends. If the active phase is chosen again, the phase is extended by 10 seconds. If it chooses a phase that is different to the previous phase, 3 seconds of yellow time are triggered on all signals turning from green to red, after which the newly chosen phase begins and lasts for 10 seconds. This allows for up to one phase of future signal plans to be used for GLOSA.

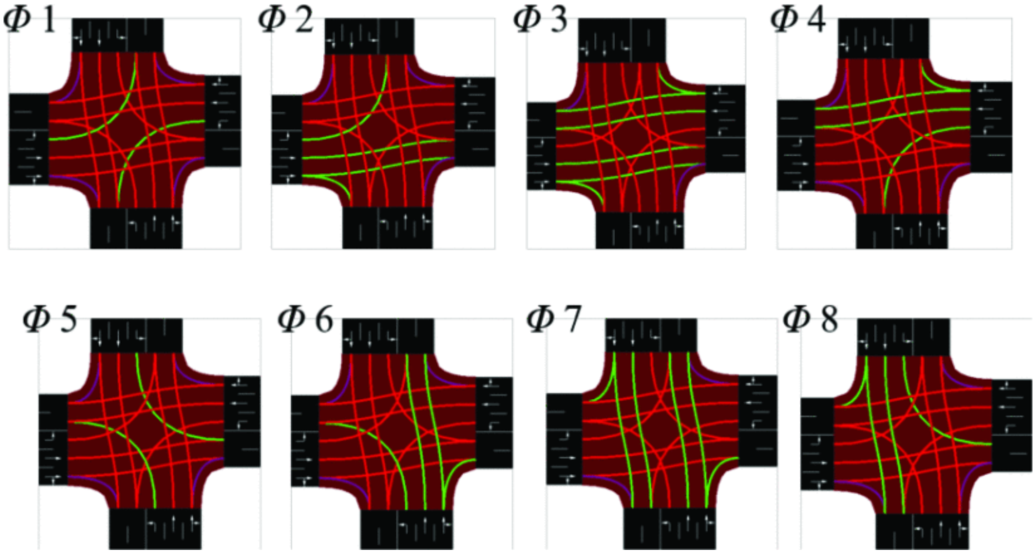

Figure 15: The eight available signal phases [228].

4.2.4 Reward Function

For the application of this research, the reward function must encourage both efficient traffic control and actions which take best advantage of GLOSA. Also to ensure training stability, the reward function should contain multiple metrics, to reduce the chance of the trained agent finding an exploit and acting in a way that maximises the reward but does not achieve efficient traffic control or take advantage of GLOSA.

In order to achieve efficient traffic control, the number of Queuing vehicles, and the number of Vehicles served were originally considered. However, the former encouraged a frequent phase change behaviour which kept vehicles in queues moving.

This movement was of course slow as the vehicles were just repositioning themselves in the queue. But it was fast enough for the vehicles not to count as queuing. This improved the agents score but made counting of queuing vehicles less accurate and created inefficient start-stop driving patterns. Therefore, it was replaced with the number of vehicles approaching the junction.

In order to encourage beneficial use of GLOSA, a reward is assigned for getting vehicles through the junction without them having to stop.

Finally, in early tests a situation frequently arose where the agent would permanently close one or more routes, discovering a local maximum in the reward function. To address this, a penalty was assigned if forty cars were stopped in a single lane, and in that case, all vehicles in the lane were deleted.

Accordingly, the final reward function is the sum of many metrics and is described by the following equation,

$$R_t = 10C + 5D - 7200P - \sum_{i=1}^{12} q_i,$$

where: C is the number of vehicles that have entered the junction since the previous action was selected, D is the number of those C vehicles which did not have to stop between spawning and entering the junction, P is the number of penalties triggered, and each qi is the number of vehicles in the i-th approaching lane. All coefficients were selected by manual iterative process using by-hand methods to achieve a balance of priorities for the agent in this situation, but these could be configured for any real implementation. The coefficient for P was selected to be much larger than the others so the penalties would not be seen as an acceptable price for clearing a large number of vehicles from the simulation.

4.2.5 Network Architecture

The networks used are fully connected feed-forward neural networks. Both have four layers: one input layer, two hidden layers and one output layer. These layers have 54, 64, 32 and 8 nodes, respectively.

After every training step, the parameters of the main network are optimised using the Adam (Adaptive Moment Estimation) method [**?**] to minimise the loss function, which is chosen to be the Mean Square Error loss function. The target network is updated to match the first network after every episode. The Leaky ReLU activation function is used at the Input and hidden layers. All these details were selected by manual iterative process using by-hand methods to avoid both over-learning, which would render the AI unable to extrapolate correctly to unseen scenarios, and the creation of agents that had achieved no learning.

4.2.6 Hyperparameters

The following values were selected by manual iterative process using by-hand methods. The replay memory has a size of 10,000. The minibatch size is set to 128. The discount factor, gamma, is set to 0.999. Epsilon is equal to 0.9 at the start of training and decreases exponentially with time until it reaches its minimum of 0.01 halfway through training. The learning rates varied and were typically set between 0.01 and 0.001 depending on the scenario and the level of GLOSA penetration during training. All these details were selected by manual iterative process using by-hand methods to avoid both over-learning, which would render the AI unable to extrapolate correctly to unseen scenarios, and the creation of agents that had achieved no learning.

4.3 GLOSA Implementation

For this implementation, GLOSA equipped vehicles are assumed to be drivers who have GLOSA-equipped vehicles and are obeying the speed advisory messages they receive. All other vehicles and drivers are assumed to either ignore GLOSA or not have it installed.

This is a simplification of real GLOSA equipped vehicles as many studies have found that drivers can not follow speed advisories perfectly and because it is likely that not all drivers of equipped vehicles will follow speed advisories. However, factoring in these behaviours would introduce a large amount of uncertainty into the results, and therefore it was decided that such behaviours would be ruled out of scope for this thesis.

GLOSA equipped vehicles are modelled using the maximum desired speed attribute, in aimsun, of the vehicles which is set to a value that would allow them to reach the lights while they are green. Because of the limited action space that only provides a phase worth of future signal plans, only vehicles on a currently green approach or will be turning green shortly will receive instruction.

The maximum desired speeds are calculated according to a simple GLOSA implementation and are only dependent on the distance to the junction and the time until green [**?**]. This algorithm was chosen because of its low resource usage, which allows for faster model training times. However, this comes at the cost of real-world performance.

Suppose a vehicle is travelling toward a light that is currently green. In that case, its maximum desired speed will be set so that it will reach the junction at the earliest possible time, but no earlier than 2 seconds after the light turns green. The reason vehicles are often instructed to reach the junction 2 seconds after the light has turned green is safety. Telling a vehicle to arrive as the light turned green would mean telling it to approach a red light and hope that it turned green as (or before) they crossed the stop line, which it might not if there is a technical failure or if the speed advisories were not followed closely enough.

If it cannot get to the intersection during the green time, it will be instructed to smoothly slow down to a minimum speed equal to half the speed limit. This speed was chosen because it is slow enough to allow the vehicles to make smoother stops should the light ahead of them not turn green before they reach the stop line and increase the chance that the light ahead of them will turn green while the vehicle is approaching, reducing the number of stops and the total stopping/queuing delay. Slower speeds were avoided as it was noted in the literature review that slow-moving GLOSA-equipped vehicles tended to cause drivers to unequipped vehicles to get frustrated.

If a vehicle has stopped within 20m of the junction, GLOSA is no longer applied, and the vehicle's maximum desired speed will be reset to its original value.

4.4 Fixed Time Benchmark

In addition to comparing the results to those achieved by Mengyu Guo et al. [228], a fixed time benchmark will also be used as this will allow a more detailed comparison of performance can be made. As this test bed is an isolated junction, the formula for saturation flow by Kimber et al. [218], Webster's formula for equisaturation [220], and Equation 22 for the degree of saturation, can be used to find an optimal fixed time signal plan for each scenario (see Table 4).

Table 4: Fixed time signal plans obtained using Webster's method.

Phase	1	3	5	7
55% Scenario	9.79	10.64	5.00	6.18
70% Scenario	16.86	18.33	8.429	10.65

5 Isolated Junction Experiments

In the previous chapter an isolated junction testbed was created, a joint control framework for non-autonomous vehicles that uses reinforcement learning traffic control was designed and implemented, GLOSA was incorporated, and fixed time benchmarks were identified. However, in order to begin addressing the aims and objectives of this thesis and gain a better understanding of how a RLTC based joint control framework designed for non-autonomous vehicles will behave on isolated junctions, experimental results must be acquired which demonstrate the performance of a joint control framework designed for non-autonomous vehicles on an isolated junction.

To do this, some experimental design work must be undertaken. As stated in the methodology, the experimental data required pertains to the performance of the joint control framework on isolated junctions, and more specifically demonstrating that the joint control framework out performs benchmark systems and quantifying how the joint control framework will perform if the penetration rates of GLOSA at the site of deployment are different from those in the training scenario. The experiments must also include the training of multiple agents. Therefore, two experiments will be needed.

The first experiment of this chapter must be to train a number of RLTC agents at a range of GLOSA penetration rates (called training penetration rates or TPR). Training an agent with a TPR of 0% is equivalent to training a RLTC system without GLOSA. During preparations for running this experiment, it was found that training with agents TPRs of 0%, 10%, 20%, 40%, 60%, 80%, 100% in each traffic scenario for 200 episodes each, in which the agents control the signalised junction for 30 simulation minutes, was sufficient to produce trained agents that met or exceeded the performance of the fixed time benchmarks and therefore could be used in the second experiment.

It should be noted that training took many attempts while parameters and hyperparameters were tuned, and during this process a vast majority of trained agents were rejected visually as their decision making was clearly flawed, exhibiting behaviour like repeatedly serving the same approach, or failing to serve one or more approaches.

If an agent is not failed visually, it will then be evaluated with the evaluation penetration rates (EPR) set equal to the TPR and compared to the benchmark. This will take place over 50 episodes, each 30 simulation minute long, and the speeds, stopping times, and junction entry speeds of all vehicles will be monitored, which will be sufficient to validate the performance of the joint control framework against the benchmark systems, and provide the experimental data required to address the first gap in knowledge for isolated junction scenarios.

If an agent completes evaluation but, despite passing the visual check, achieved inferior performance compared to the benchmarks in this evaluation, further parameters and hyperparameters was conducted.

The second experiment will evaluate the agents, which were trained and passed evaluation in the first experiment, in scenarios with EPRs that are different from their TPRs on the isolated junction testbed. Evaluation will last for fifty episodes, each thirty simulation minute long, and the speeds, stopping times, and junction entry speeds of all vehicles will be monitored which will be sufficient to provide the results required to address the second gap in knowledge for isolated junction scenarios.

5.1 Experiment 1 - 55% scenario

When looking at the experimental data, the first important results to check were those that would validate the performance of the joint control framework against the benchmark systems. This was done by looking at the performance of the 0% TPR policy which outperformed the fixed time traffic control benchmarks in some metrics when GLOSA is disabled. Compared to the fixed time signal plan defined in section 4.4, the average queue length is reduced by 11%, and the average speed of entry into the junction is increased by 9%. Compared to the framework by Mengyu Guo et al. [228], the average queue length is reduced by 28%.

The 100% TPR agent's performance was also examined, and it was found that it manages a 25% reduction in waiting times compared to the fixed time signal plan, as well as a 38% reduction in queue lengths and 19% increase in average junction entry speed. Compared to the framework by Mengyu Guo et al. [228], it reduces stopping time by 5% and stops by 50%.

Overall, all the final agents in this scenario, regardless of TPR, outperformed both benchmarks in terms of stopping time and queue length or number of stops, and all agents outperform the fixed time benchmark in terms of average speed of entry into the junction. These results validate that the potential performance of the joint control framework exceeds the benchmark systems in this traffic scenario. Furthermore, these results were achieved with the simulations running faster than real time on a single desktop computer with a consumer CPU and GPU, meaning that computation is not a barrier to the real-world deployment of this framework.

The second important result to check was the difference in performance between TPRs of 0% and 100%. It was found that the agent trained at a 100% TPR decreased the average vehicle stopping time by 30% and increased the average junction entry speed by 9% compared to the 0% TPR agent. This implies that the joint control framework was successful as the included GLOSA had a positive effect on overall traffic flow despite its use in combination with an adaptive traffic control system in this traffic scenario.

This result was also found to hold for agents with TPRs at 40% or 60% and above (or with greater adoption of GLOSA). Compared to the 0% TPR agent, average vehicle stopping time decreased for all agents with TPRs of 40% and higher (see figure 16), while junction entry speed increased for all agents with TPRs of 60% and higher (see figure 17). This would tally with sources uncovered in the literature review which suggested that the effectiveness of GLOSA was not significant until a critical mass of GLOSA equipped vehicles was reached.

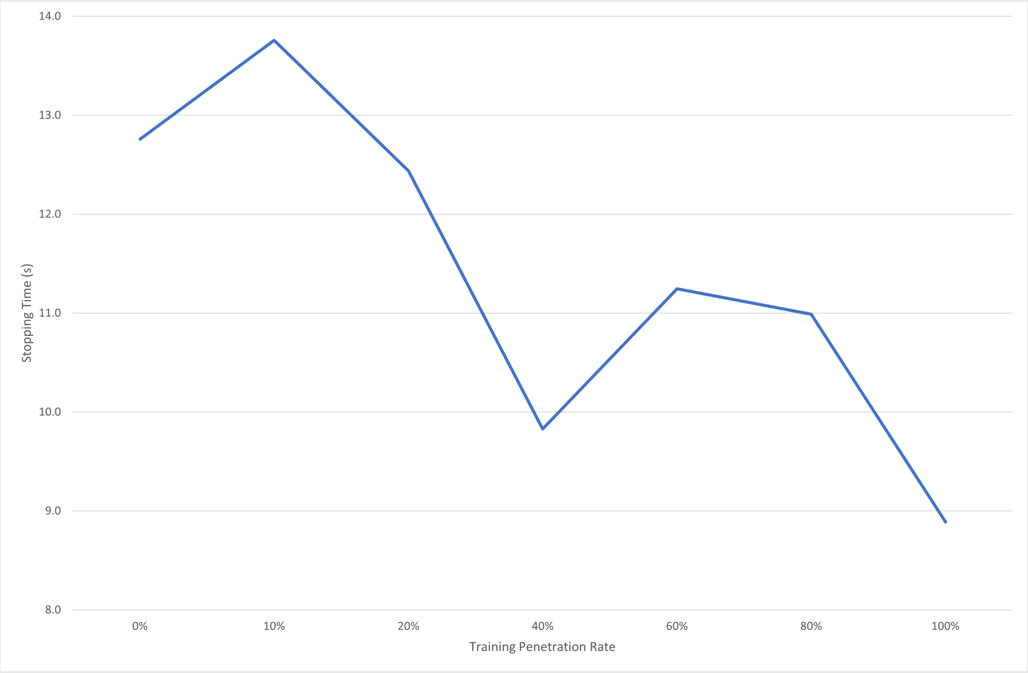

Figure 16: Performance on the metric of average vehicle waiting time in experiments by TPR at 55% traffic density

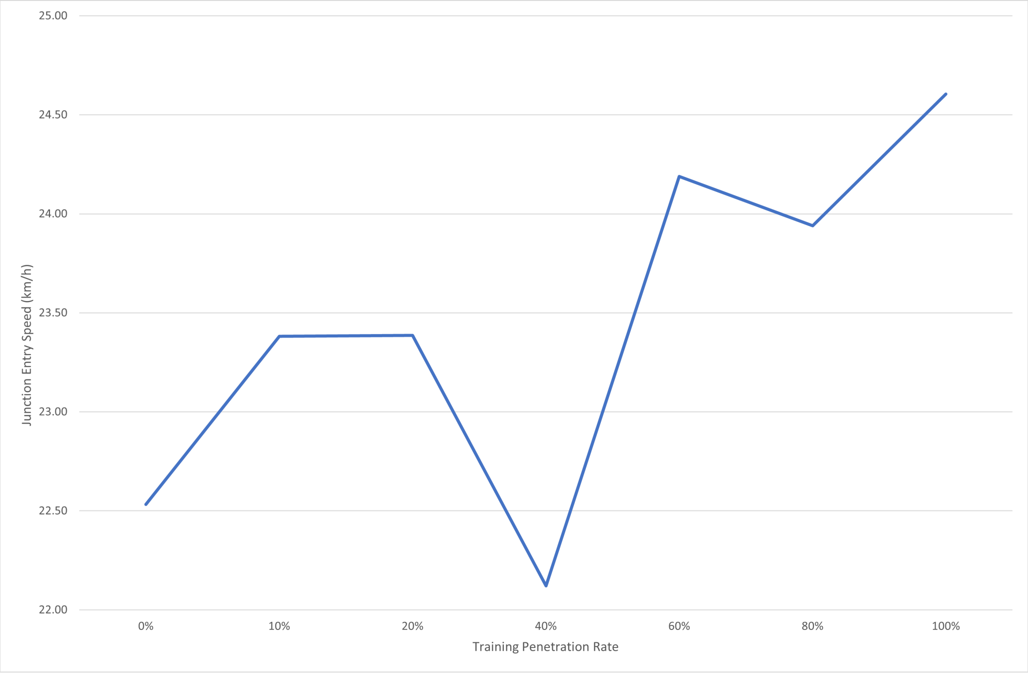

Figure 17: Performance on the metric of average junction entry speed in experiments by TPR at 55% traffic density

These results clearly demonstrate that GLOSA was able to have a positive effect on driving smoothness, with drivers able to slow down smoother when approaching red lights, eliminating stops. Also, thanks to advanced warning of green phases from the GLOSA system, equipped vehicles were able to speed up earlier and clear the junction faster, improving the junction's efficiency.

However, it should also be noted that the average speeds appeared to decrease in agents with TPRs of 60% and higher, with the largest reduction seen being around 6% as the TPR was increased (see figure 18). Perhaps this can be address with further design tweaks and/or a greater amount of training.

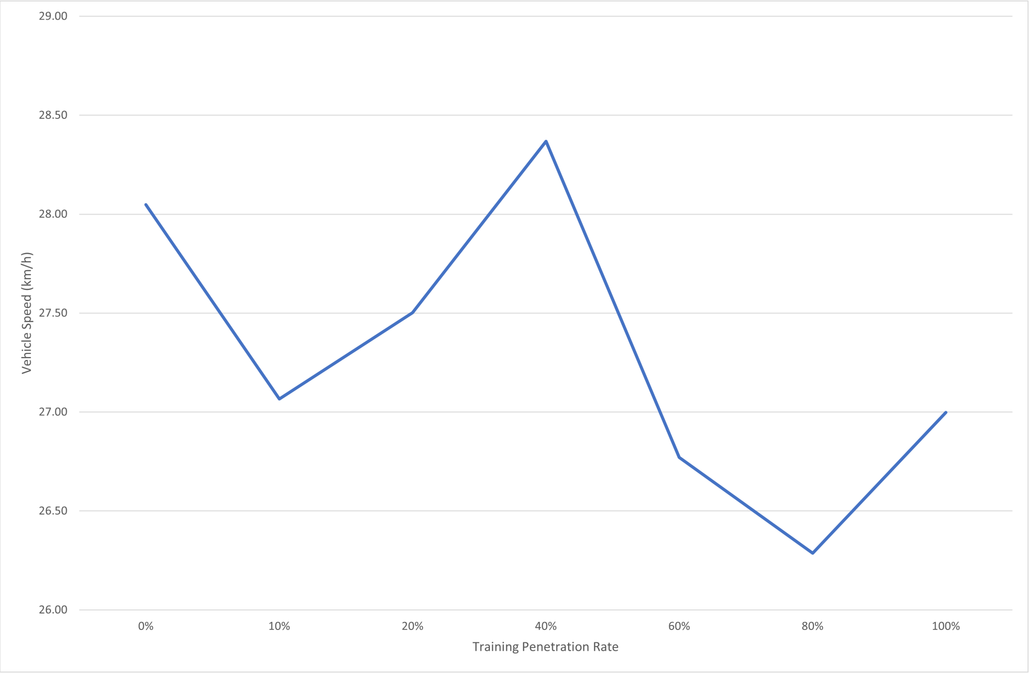

Figure 18: Performance on the metric of average vehicle speeds in experiments by TPR at 55% traffic density

One final result of interest from this first experiment was the performance of the 40% TPR agent which is something of an outlier having recorded far better vehicle speeds than the other agents, better stopping times than the 60% and 80% agents, but the worst junction entry speeds of any agent. This result is the extreme example of the randomness of training these agents and demonstrates a limitation of the test methodology used here.

While fixed random seeds ensure that each agent sees the same two hundred episodes, in terms of vehicle spawning times, during training and ensure that the random decisions taken by the agent are the same for all agents, each agent does not experience the same two hundred training episodes. There are two reasons for this. Firstly, agents with different TPRs will see different vehicle behaviours as more or less vehicles follow GLOSA instructions. Secondly, a small change in parameters or hyperparameters will quickly cause agents to make differing choices in the episodes as they update their neural networks, leading episodes that would otherwise match, to reach a wide array of different situations and conclusions, which only increases the gulf as different agents see, remember and learn from differing experiences.

A simple solution to this would be to train more agents, however the training of these agents and the tuning of the hyperparameters is extremely time consuming by hand. This will be especially true in an arterial environment, where several agents must be trained cooperatively.

The best remedial option for this going forward, is the adoption of hyperparameter optimisation to automate this process, in the experiments in phase 2, allowing training to be performed much faster, with far less manual input, and therefore allowing for more agents to be trained ensuring the agents selected for evaluation will not only be better trained, but also will all be more similar in terms of quality.

Overall, the results in this scenario demonstrate that a RLTC system based joint control framework can have benefits when applied to Connected Non-Autonomous Vehicles on isolated junctions.
A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

5.2 Experiment 1 - 70% scenario

In the 70% traffic density scenario, the policies became less effective as the TPR was increased. As the TPRs increased, the stopping time tended to increase (see figure 19), and the average speeds tended to decrease (see figure 20). The only positive was the average junction entry speed (see figure 21), which did appear to increase as the TPR increased.

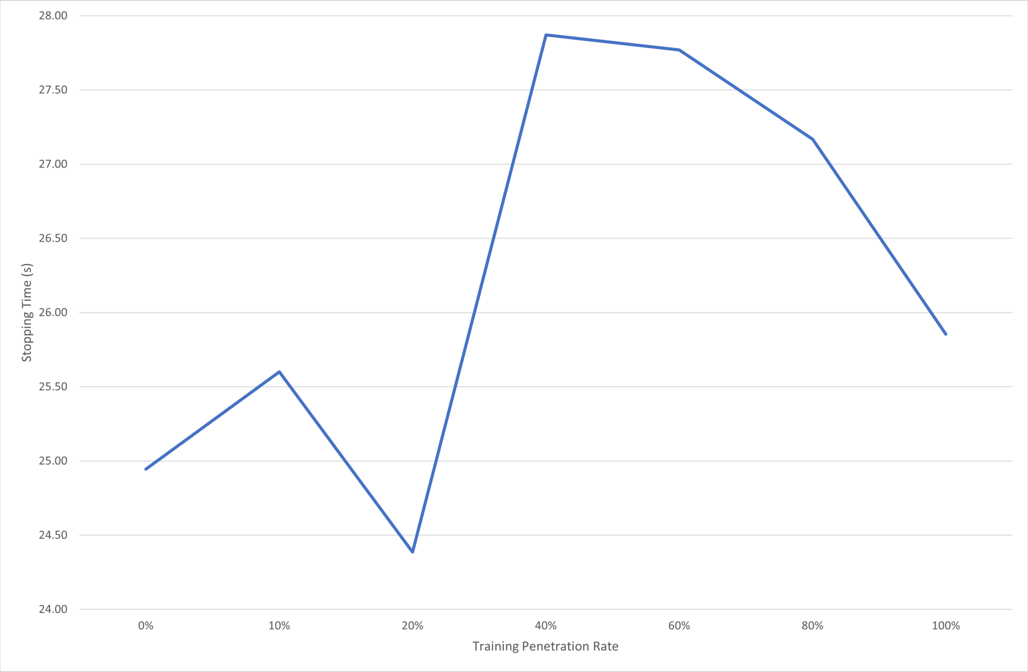

Figure 19: Performance on the metric of average vehicle waiting time in experiments by TPR at 70% traffic density

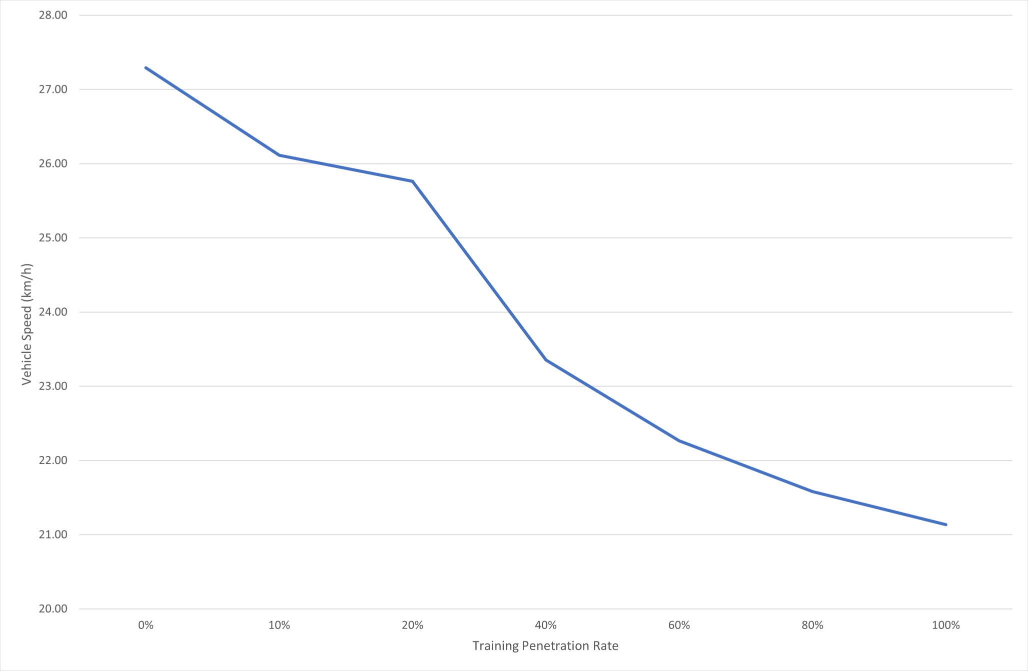

Figure 20: Performance on the metric of average vehicle speeds in experiments by TPR at 70% traffic density

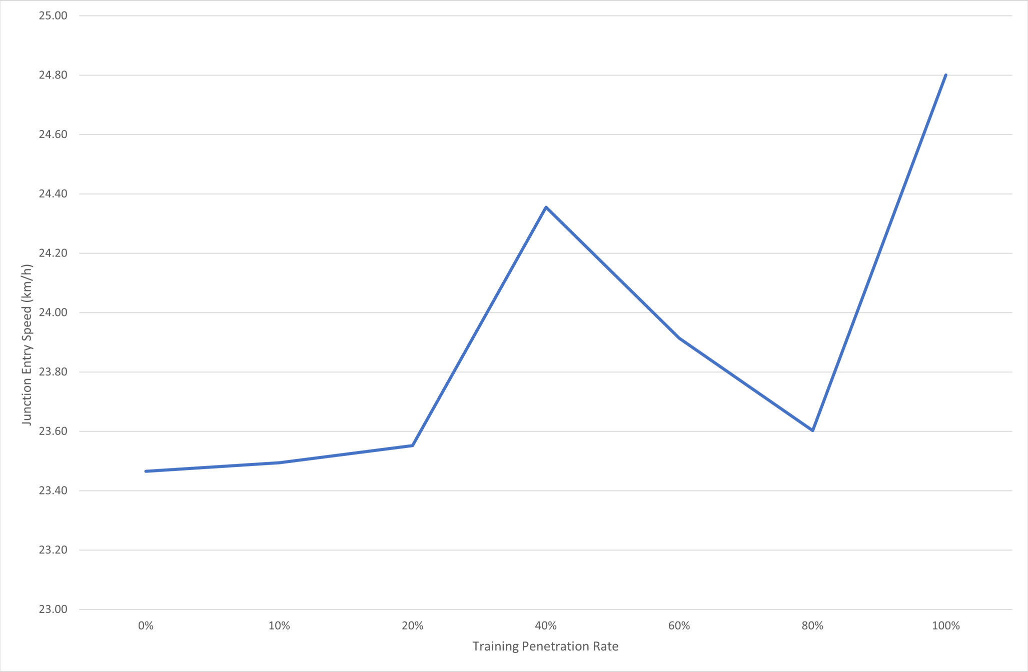

Figure 21: Performance on the metric of average junction entry speed in experiments by TPR at 70% traffic density

This was caused by GLOSA being less effective at higher traffic densities because of the increased frequency of queues forming at the junction that block approaching equipped vehicles from following the GLOSA instructions. This queue formation is exacerbated, regardless of the GLOSA penetration rate, because the limited time horizons available and the limited activation distance placed on the system were insufficient to allow equipped vehicles to pass the junction without stopping. Most equipped vehicles could not be provided with useful future signal plans and were instead simply told to approach the junction at a lower speed. However, it is possible that with the inclusion of hyperparameter optimisation in training, alongside further tweaks, that a working policy could be found for this scenario in future experiments. Until then though, this framework is clearly only workable in free flow conditions.

5.3 Experiment 2 - 55% scenario

In the 55% traffic density scenario, as the EPR increased, the TPR 0% policy performed worse on the metrics of stopping time and vehicle speed, while there was little change in junction entry speed (see figures 22, 23, and 24). The 10% TPR policy was affected in an analogous way to the 0% TPR policy. The stopping time increased, and the vehicle speeds decreased as the EPR increased, although these changes were more gradual. However, in contrast to the 0% TPR policy, the junction entry speed increased with the 10% TPR policy as the EPR was increased. The 20% TPR policy was the first policy to show decreased stopping time as the EPR increased. It also achieved increased junction entry speeds at higher EPRs. However, vehicle speeds still decreased as the EPR was increased. All agents with TPRs above 20% behaved in a comparable way.

The first interesting result was that the agents with TPRs of 10% or less were unable to manage GLOSA equipped vehicles. The cause was that the agents made decisions expecting vehicles to not be equipped with GLOSA and therefore often chose to prioritise existing queues over approaching vehicles, as it could not trust that approaching vehicles were GLOSA equipped and would clear the junction without stopping but it could guarantee that the queue would be cleared. It is possible that this issue could be address with a different state space or reward function, which allowed the system to better determine a vehicles likelihood to clear the junction without stopping and earn a greater reward when one did.

The second interesting result is that agents with TPRs at or above 20% are not negatively impacted by the introduction of more GLOSA equipped vehicles but are negatively impacted by a reduction in GLOSA equipped vehicles. The latter part was expected, as less GLOSA equipped vehicles should mean less benefits of GLOSA are seen. However, the former part was very unexpected as it was hypothesised that a deviation in driver behaviour would always be bad for the agent. Nevertheless, the simple explanation is that at these higher TPRs, it made sense for the agent to gamble that a vehicle was GLOSA equipped and that it would clear the junction without stopping. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues.

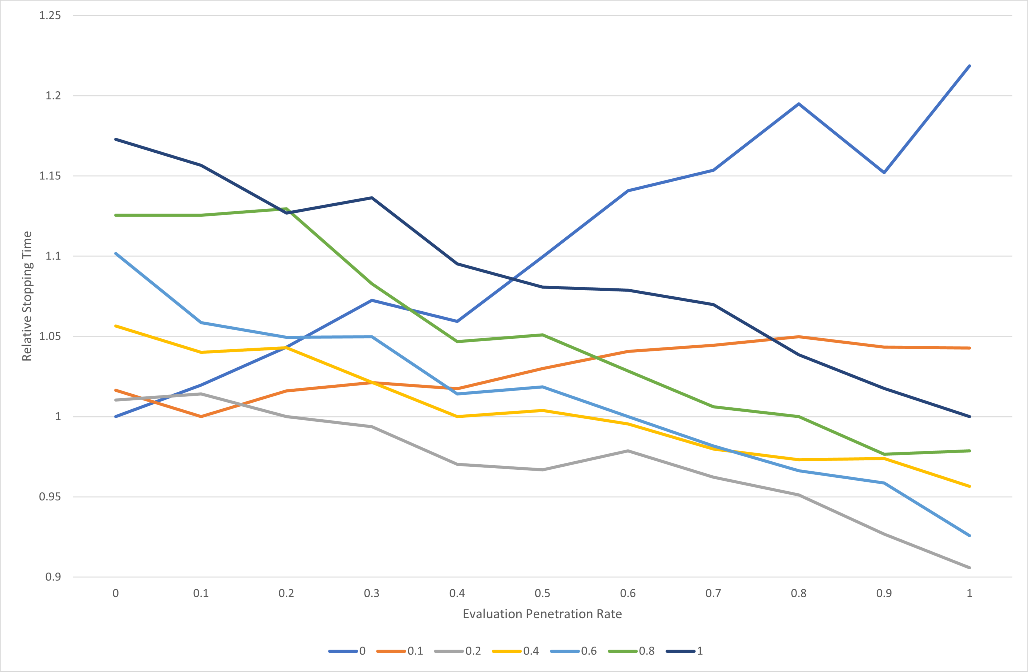

Figure 22: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 55% traffic density

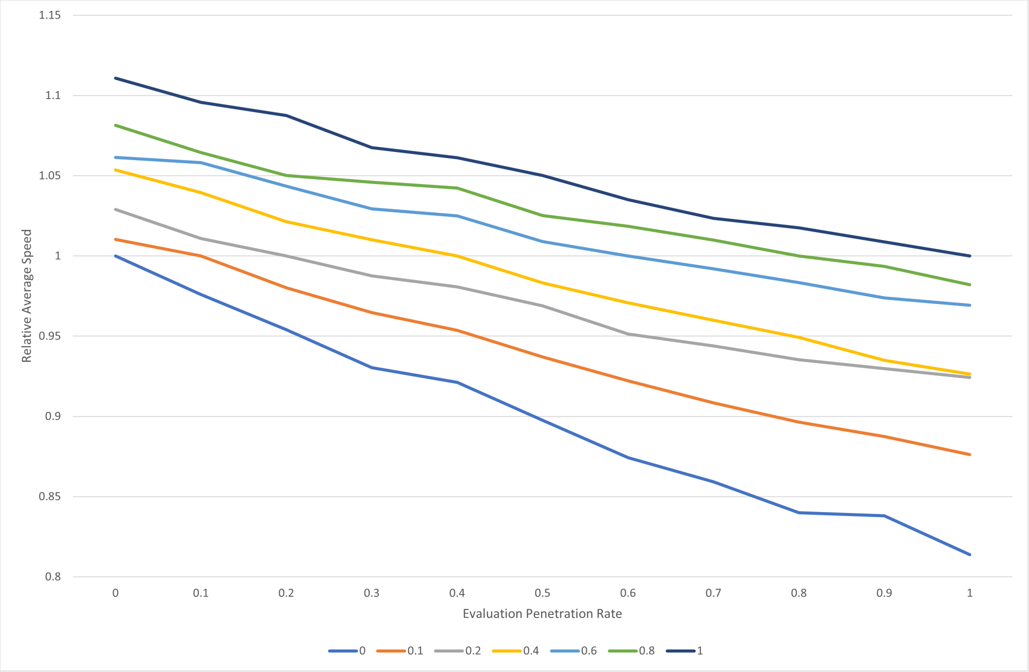

Figure 23: Relative performance on the metric of average vehicle speeds in experiments by EPR at 55% traffic density

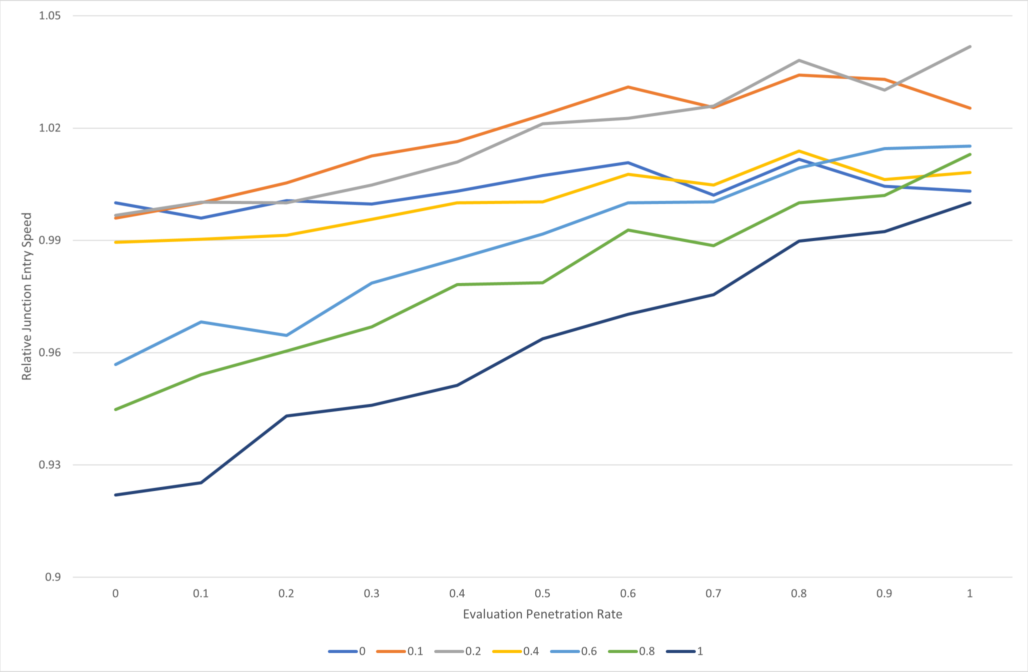

Figure 24: Relative performance on the metric of average junction entry speed in experiments by EPR at 55% traffic density

Overall, the results for the 55% traffic density scenario suggest that divergent values for the TPR and EPR, combined with a low TPR, can lead to deficient performance for the joint control framework. Therefore, the benefits of having more equipped vehicles can be outweighed by the drawbacks of the RLTC system not expecting the change in driving behaviour. However, past a certain TPR there are only benefits to increasing the EPR.

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

5.4 Experiment 2 - 70% scenario

Despite the failure of the framework in the 70% scenario, it was decided to continue testing into the second experiment to identify if any of the policies improved at any EPRs. However, as EPRs were increased, most metric deteriorated at all TPRs, especially average speeds (see figure 26). However, some lower TPR agents did manage to increase junction entry speeds and decrease stopping time when the EPR was increased (see figures 25 and 27). Unfortunately, this behaviour is currently unexplained. Regardless though, these results imply a failure of the framework to train agents capable of acting as joint control systems in this scenario.

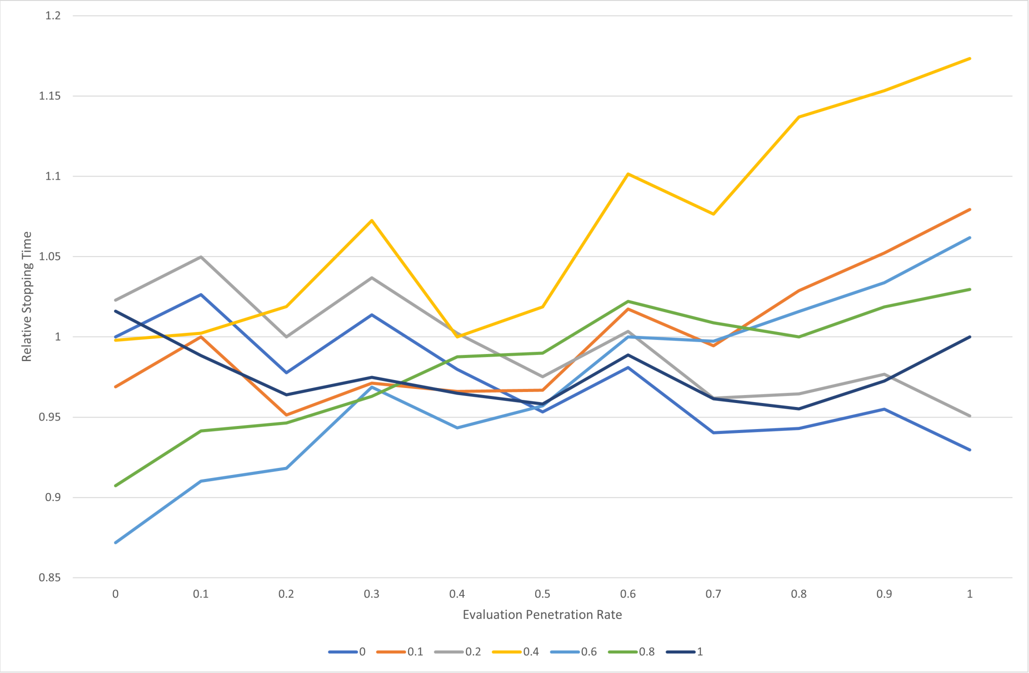

Figure 25: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 70% traffic density

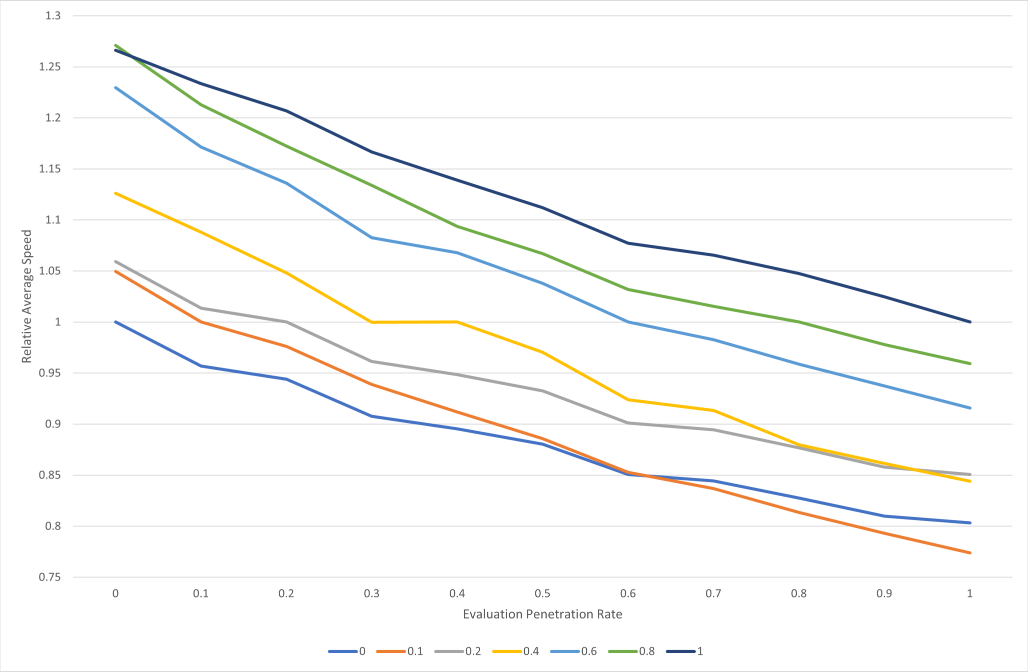

Figure 26: Relative performance on the metric of average vehicle speeds in experiments by EPR at 70% traffic density

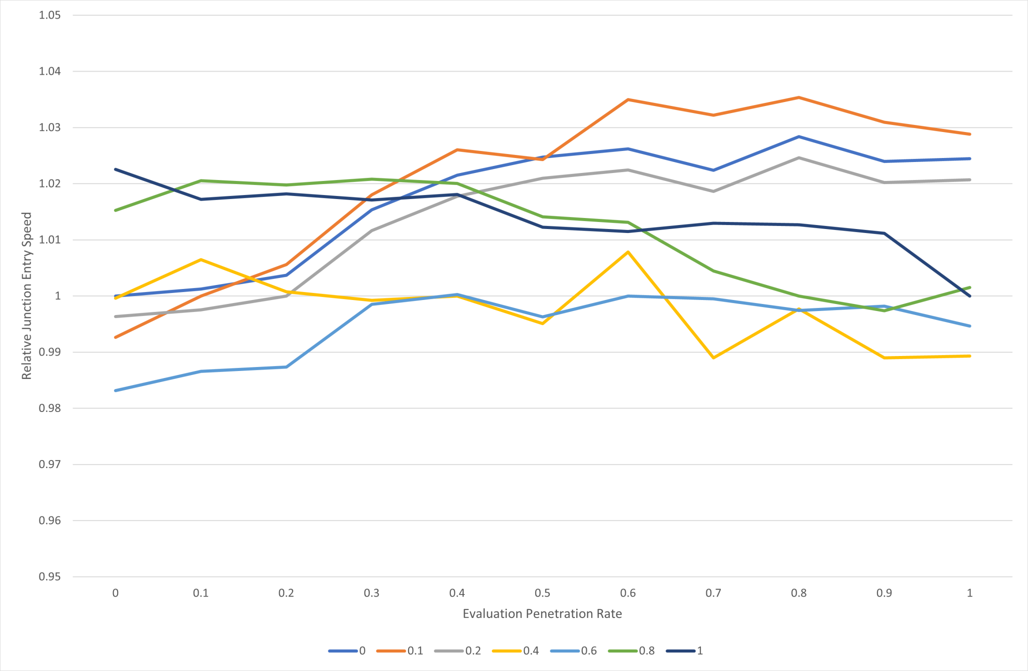

Figure 27: Relative performance on the metric of average junction entry speed in experiments by EPR at 70% traffic density

5.5 Experiment 1 and 2 - Discussion

From these experiments, there are two main results. First, there are scenarios where this framework can be successful although it is limited to lower traffic densities. Secondly, changes in EPRs can lead to weaker performance, with low TPR agents performing worse at higher TPRs and other agents performing worse at lower TPRs.

In learning this, the goals of phase 1 have been completed. However, there are some short comings of this framework and experiment design which will need addressing in phase 2. Steps must be taken to speed up, increase the stability of and automate training, so that more agents can be trained and better hyperparameters found. This should include the implementation of hyperparameter optimisation, as well as other steps to improve training stability and to reduce the computational load of training. Also, many of the add-ons to DQNs that have been previously reviewed could be implemented to improve stability of training. Also, finding a way to increase the amount of future signal plans could make GLOSA more effective at avoiding queues, which might make the framework more effective at higher traffic densities.

6 Combining RLTC and GLOSA on Arterial Flows

Now that a better understanding of how a RLTC based joint control framework designed for non-autonomous vehicles will behave on isolated junctions has been obtained, the next required step in the process of addressing the identified gaps in knowledge and achieving the stated aims and objectives of this thesis is to gain a better understanding of how a RLTC based joint control framework designed for non-autonomous vehicles will behave on arterial flows. To gain this understanding, experimental results which demonstrate the performance of such a joint control framework on an arterial flow must be obtained.

Getting that experimental data requires: the finding or building of an arterial flow testbed, the design and construction of a joint control framework for non-autonomous vehicles that uses reinforcement learning traffic control and is applicable to arterial flows, the implementation of GLOSA on the test bed, and the choosing of benchmarks to allow the performance of the joint control framework to be validated. Also, the recommendations from the previous chapter, like the use of hyperparameter optimisation and finding a way to increase the amount of future signal plans, need implementing.

Therefore this chapter will be split into four sections.

- Testbed design which will include discussion of microsimulation packages, isolated junction layout, and traffic scenarios,
- RLTC system implementation which will include discussion of: RL algorithms, state and action spaces, reward functions, neural network architecture, and hyperparameter optimisation,
- GLOSA implementation,
- Traffic Control Benchmarks.

6.1 Testbed

In this section, the design of the arterial flow testbed will be described. This description will be split into four parts: the choice of microscopic traffic simulator, the junction layout, the traffic flows, and the GLOSA implementation.

6.1.1 Microscopic Traffic Simulators

After the previous experiments, it was recommended that ways be found to speed up training, so that better agents could be trained. This recommendation is made even more pressing by the fact that these new experiments require far more time to complete than the isolated junction experiments for two reasons. Firstly, the simulation testbed is larger and must simulate more vehicles at any given time. Secondly, multiple agents must be trained together, increasing the complexity of training and the number of attempts needed to find hyperparameters for which the agents will train.

Therefore, the priorities when selecting a microscopic traffic simulator for these experiments are different to those in the isolated junction experiments. As Aimsun, SUMO, and VISSIM met the previous requirements they were considered, with speed now being the most important secondary factor, instead of availability and quality of learning resources.
Aimsun's[212] scripting interface does not have a command to start a simulation, which limits the opportunities to automate testing and requires the simulation to be running before the script can interface with it. It also limits parallel simulations per machine and per license, which places a limit on the number of RLTC agents that can be trained simultaneously.

While VISSIM[216] does allow simulations to be started from the script, allowing for easy automation of repeated experiments, it has a slightly restrictive parallelisation policy that limits users to running at most four simulations per license at any given time.

As SUMO[214] is free and open-source software, it can be run in parallel without limits. During testing, it was also found to complete simulations faster than the other software packages considered here. Furthermore, SUMO also offers the option to import road networks from the OpenStreetMap databases.

After re-investigating each microsimulation package, SUMO 1.15.0 was selected for these experiments because of its speed and flexible approach to parallelisation.

6.1.2 Arterial Flow Layout

With the simulation package selected, it remains to design an arterial testbed. As the performance of GLOSA is highly dependent on the network geometry, specifically the distance between junctions, and as it was important to validate the RLTC system for many types of junctions, it was decided that the testbed should fulfil two requirements. Firstly, it should have between 10 to 15 signalised junctions, which should have varying geometry. Secondly, these junctions should not be evenly spaced. These will help to validate the applicability of this research to more types of junctions and network geometries.

The testbed that was created was a one-to-one scale simulation that closely approximates a 3-mile stretch of the A3024 in Southampton, England, between the Windhover Roundabout and its junction with the A334 as this met the criteria. The eleven unevenly spaced signal-controlled junctions that are included in the test bed are modelled after the A3024's junctions with: Botley Rd, Warburton Rd, Orpen Rd, Gavan St, Kathleen Rd, Hinkler Rd, North East Rd, Upper Deacon Rd and Deacon Rd, Ruby Rd and Bath Rd, White's Rd, and Bitterne Rd East (A334).

Two un-signalled junctions are modelled at Sedgewick Rd and at Bursledon Rd, between the junctions with White's Rd and Bitterne Rd East, are also modelled. Minor turnings from driveways and shops or smaller junctions not listed above have not been implemented. Openstreetmap.org was used to provide the initial layout, which was then altered so the number of lanes at each junction and on each link matched the real-world layout and the available signal phases matched the real-world junctions. For example, the junction between the A3024, Upper Deacon Rd, and Deacon Rd operates as a staggered crossroads with four phases which each serve a different approach.

All modelled roads except the A3024 will be collectively referred to as the side roads. To allow the application of the framework to this arterial, all incoming lanes from side roads have each been extended to at least 200m, while other incoming lanes have been extended to be at least 350m. This is to allow for approaching GLOSA equipped vehicles to had adequate reaction time. Outgoing lanes at the entry/exit points have also been extended. The speed limit is 40mph along the entire simulated arterial.

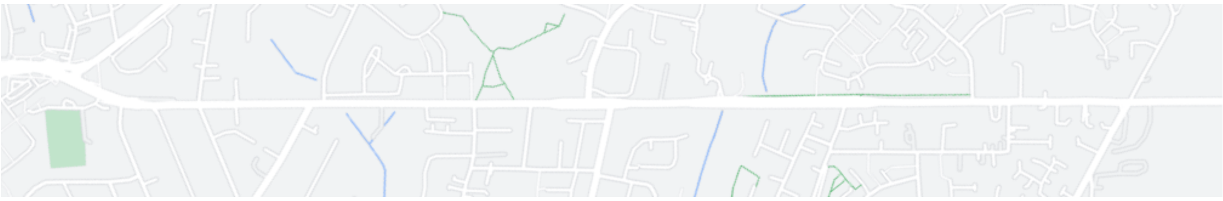

Figure 28: The A3024 in Southampton between Bitterne Rd East and Botley Road.

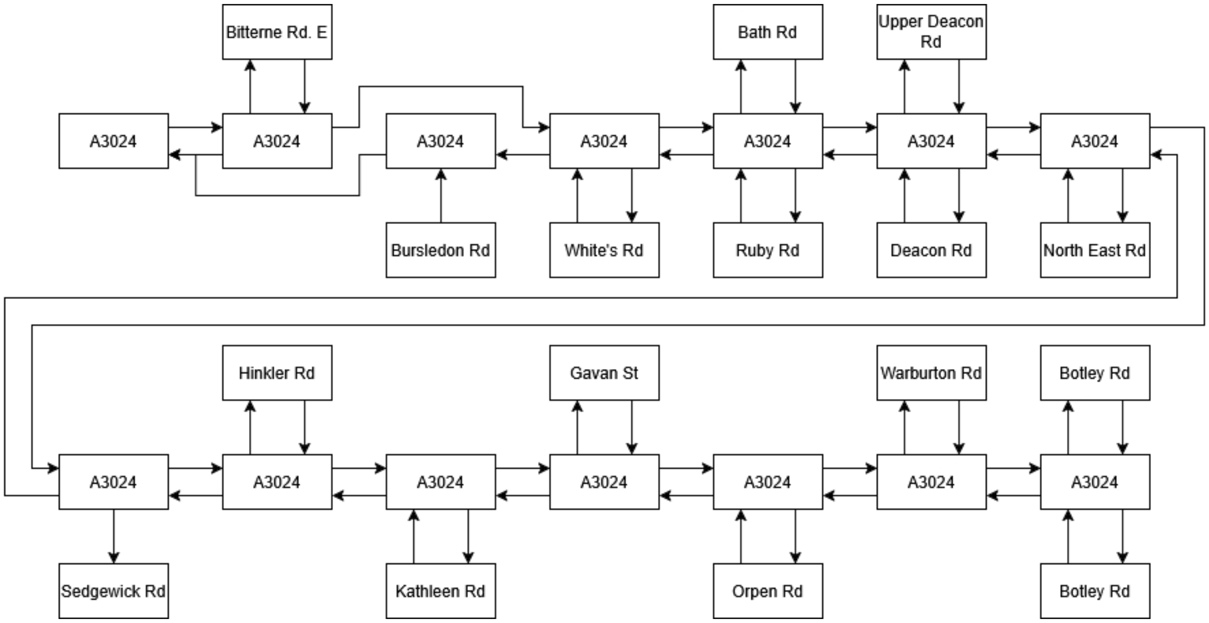

Figure 29: An abstract diagram of available routes on the simulated A3024.

6.1.3 Traffic Flow

With the testbed layout selected, it remains to design a traffic flow to use for the experiments. Once again it was decided that, as GLOSA are more effective at lower traffic densities, they should remain the focus of this research.

Three scenarios were designed using LinSig[222] for the arterial flow experiments. More details on how this was done can be found in the LinSig 3.2 User Guide [226]. All scenarios assume that at each junction, a fixed percentage of vehicles approaching a junction will turn onto each of the side roads while the remaining vehicles continue on or navigate onto the main arterial. Vehicles joining the arterial from side roads turn onto the A3024 Eastbound and A3024 westbound in equal measure. Botley Rd, Warburton Rd, Orpen Rd, Gavan St, Kathleen Rd, Hinkler Rd, North East Rd, Upper Deacon Rd and Deacon Rd, Ruby Rd and Bath Rd, White's Rd, and Bitterne Rd. East (A334) all have equal inflows and outflows.

The two busiest junctions on the arterial are with Bitterne Rd. East (A334) and Botley Rd, the former directly connecting two main roads and providing access to central Southampton via the Northam River bridge, and the latter providing fast access to the A3025 and central Southampton via the Itchen bridge. The other junctions are residential streets providing access to one or more cul-de-sacs. Sedgewick Rd is a one-way street providing access to the same residential area as Kathleen Rd and North East Rd. As the latter two roads provide faster access in most cases, the volume of traffic turning onto Sedgewick Rd should be lower.

To reflect this, the arterial has been modelled with the following assumptions. At the Bitterne Rd East (A334) junction, 25% of vehicles approaching from the A3024 (Eastbound) navigate onto Bitterne Rd East (A334). At the Botley Rd junction, 10% of vehicles approaching from the A3024 or Botley Rd (NE) leave the network via Botley Rd (SW). At the Sedgewick Rd junction, 2.5% of approaching vehicles turn onto Sedgewick Rd. Lastly, on all remaining junctions, 5% of approaching vehicles will navigate onto each available side road.

In scenario one, the arterial flow has a Practical Reserve Capacity (PRC) of 81.8%, which is equivalent to the arterial operating capacity of 55%. In scenario two, the arterial flow has a PRC of 53.8%, equivalent to operating at 65% capacity. In scenario three, the arterial flow has a PRC of 33.3%, equivalent to operating at 75% capacity.

6.2 RLTC Implementation

With the testbed and scenarios designed, the next task is the implementation of a RLTC system. Since the testbed has been changed from an isolated junction to an arterial flow, some changes must be made to the previous framework's state and action spaces. However, other changes were also recommended in the previous section to address issues discovered in the previous experiments. For example, the previous framework proved unstable during training, with the DQN method prone to rapid catastrophic un-learning. Therefore, modifications will need to be implemented to address this, and other performance issues.

This is especially important as this new testbed is an arterial flow with varied junction types which will require multiple agents (One agent per junction) to control all the signalised junctions along the arterial, greatly increasing the burden of training. To lessen this burden and speed up training times, some changes must be made to the original framework before it can be deployed. These modifications include the implementation of a number of new methods: Double DQN, Prioritised Experience Replay, Dueling Network, Noisy Networks, Categorical DQN, and N-Step Learning. Also, a hyperparameter optimisation algorithm is required to automate the selection of hyperparameters, allowing for more attempts at training and a higher training success rate, allowing for the creation of better agents.

This section will be organised as follows: hyperparameter optimisation, State Space, Action Space, Reward Function, Network Architecture, Other Details.

6.2.1 Hyperparameter Optimisation

As concluded in the previous chapter, gaining experimental results which demonstrate the performance of such a joint control framework on an arterial flow is going to require that training be accelerated as much as possible to generate more agents of higher quality. However, before such an algorithm can be implemented, a specific algorithm must be chosen.

All hyperparameter optimisation algorithms aim to achieve the same thing, finding the values of hyperparameters, x, that minimise the loss of the agent over a testing set, but there are many different approaches to this problem. When deciding which would be best for this research, the following algorithms were examined: Grid Search, Random Search and Bayesian Optimisation.

Several algorithms were considered: including Grid Search, Random Search and Bayesian Optimisation.

The main concern when comparing these algorithms was the speed at which they could find near-optimal hyperparameters. Initially, the ease with which the tuning could be parallelised was a concern. However, it was then concluded that it not actually advantageous to parallelise the tuning within a given scenario. This was because two or more groups of agents could be trained independently of each other if each group were being trained for different scenarios.

Therefore, the main concern became choosing an algorithm that would minimise the number of iterations required to find the hyperparameter values that would minimise the loss. As Grid Search and Random Search are unable to narrow down the search space, Bayesian Optimisation was the clear preferred option.

6.2.1.1 Bayesian Optimisation Implementation

Bayesian Optimisation was performed, to determine the best learning rate, using the skopt (known as “scikit-optimize”) python package with expected improvement used as the acquisition function, and the noise parameter set to 10^{-10} through a manual iterative process using by-hand methods.

6.2.2 State Space

The state space for each agent/junction needed for this implementation has all the same criteria as the previous implementation, and it is therefore like the previous framework. However, a few modifications are needed to match the new scenario. The main required change is that the agent needs to know if vehicles have cleared the previous junction on their route, as otherwise it would be exceedingly difficult for the agent to determine which vehicles can and cannot clear the junction in the next phase.

Overall, the updated state space is built in two parts: one containing traffic information, and one containing phase information. The first contains five values per group of approaching lanes. The first value for each group represents the number of slow-moving approaching vehicles (travelling at less than 4.47m/s or 10mph) that do not need to clear a previous signalised junction. The second and third values for each group are the number of approaching vehicles within 304m and 179m (17 and 10 seconds of travel at 40mph) of the junction. The fourth value for each group is the number of vehicles approaching the junction that do not need to clear a previous signalised junction. The fifth value for each group is the longest current stationary time. An example of how these numbers are calculated can be seen in figure 14

The first four values should allow the agent to identify congestion and learn to clear it, with the fourth value specifically allowing an agent to consider the effects of its neighbours. The last value should be helpful in maintaining fairness on the junction and avoiding the behaviour of repeatedly selecting phases on the busiest approaches while ignoring side roads for extended periods of time.

A group of approaching lanes is, in this work, defined as a set of lanes that always share the same signal. On the isolated junction testbed, there are eight groups of approaching lanes, two on each approach, with the first combining the middle and leftmost lanes and the second being the rightmost lane.

The phase information is a vector with one entry for each available phase. The element relating to the last active phase is set equal to 1, and the rest are set to 0. This information is included to help the agent understand the difference between a lane that is quiet and, therefore, has no queuing vehicles and a busy lane that currently has no slow-moving vehicles because it received green time in the previous phase. This helps prevent the agent from wasting enormous amounts of time with frequent phase changes.

While this state space is adequate for acquiring some of the required experimental data, it needs modifications if it is to be used in a scenario where vehicle routes are not known. Accordingly, if vehicles are not providing their routes, the state space will be modified so that the second and third values so that they only count vehicles that do not need to clear a previous signalised junction.

6.2.3 Action Space

As in previous experiments, the action space for each agent/junction consists of the available signal phases at that junction with the agent selecting a new action (phase) whenever the previous phase ends. If the active phase is chosen again, the phase is extended by 10 seconds. However, some alterations have been made to equip the framework for the new scenario and to learn lessons from the previous experiments.

To ensure the junctions are clear before a new phase starts, a 5-second phase transition begins if it chooses a phase that is different to the previous phase. It includes 2 seconds of amber time, one second of red time, and 2 seconds of red plus amber time, after which the newly chosen phase begins. However, the red phase of the transition period is extended to 3 seconds on the larger Bitterne Road East and the Deacon Road junctions.
After this transition, the newly chosen phase lasts for 7 seconds if it serves traffic on a side street, or 20 seconds if it serves traffic on a main road. This change means that GLOSA equipped vehicles on the main road get a much greater degree of future signal plans.

To avoid the training of agents that achieve a local minimum by only show green time to the main road, a reflex has been added where the agents will be forced to give green phases to an approach if a vehicle has been stationary on that approach for more than 120 seconds. This should partially remove the local minimum and lead agents to show fairer junction control. This length of time had to be set on the upper end of the acceptable amount of time for a vehicle to wait at a junction, so the agent is as much as possible only interrupted when it is attempting to only show green time to the main road and is not interrupted during normal operation. Also, while this mechanism did need to somewhat reflect the amount of time any driver should be willing to wait, this was not a strong requirement as the mechanism was not intended to be relied upon by the agent, which should allocate green time to vehicles long before the mechanism triggers.

6.2.4 Reward Function

The reward function consists of three parts: the passing reward, p, which is awarded to the agent for each vehicle that clears the junction; the delay penalty, d, which is awarded for each slow-moving vehicle (less than $0.1ms^{-1}$) every timestep; and the non-stop reward, n, which is awarded to the agent for each vehicle that clears the junction without having to queue.

$$R_t = p + 10^{-4}n - 10^{-3}d \tag{23}$$

The passing reward and delay penalty incentivises the agent to keep the junction clear and keep queue lengths to a minimum, which is a behaviour that is both generally good for a traffic control system and also good for GLOSA, which will be unable to provide accurate and optimal speed advisories to approaching vehicles if queues get too large. Finally, the non-stop reward attempts to incentivise behaviours that take greater advantage of the implemented GLOSA system. All coefficients were selected by manual iterative process using by-hand methods to achieve a balance of priorities for the agent in this situation, but these could be reconfigured for any real implementation.

6.2.5 Network Architecture

Both the target and policy networks share the same architecture. All layers of the neural networks are fully connected feed-forward layers. The input layer has a size equal to the state space and is connected to the hidden layer. The hidden layer has 128 nodes and uses Leaky ReLU as its activation function. The feature layer is fully connected to both the hidden advantage layer and the hidden value layers, which both have 128 nodes. The hidden layers are fully connected to their respective output layers. The output value layer has N_{atoms} nodes, while the output advantage layer has $|\mathcal{A}| \times N_{atoms}$ nodes. All these details were selected by manual iterative process using by-hand methods to avoid both over-learning, which would render the AI unable to extrapolate correctly to unseen scenarios, and the creation of agents that had achieved no learning. All these could be reconfigured for any real implementation.

6.2.6 Other Details

Table 5: Hyperparameters for Arterial Flow Framework

Learning Rate	$[10^{-4}, 10^{-1}]$
Memory Size	$10,000$
Batch Size	128
Timesteps between updates of the Target Network	100
Discount Factor	0.9
PER hyperparameter α	0.5
PER hyperparameter β	0.6
PER hyperparameter ϵ_{prior}	10^{-6}
v_{min}	-10
v_{max}	100
N_{atoms}	51
n_{steps}	3

All these details were selected by manual iterative process using by-hand methods to avoid both over-learning, which would render the AI unable to extrapolate correctly to unseen scenarios, and the creation of agents that had achieved no learning.

6.3 GLOSA Implementation

With the testbed and updated frameworks built, it remains to implement a GLOSA. For these experiments, the implementation of GLOSA included with SUMO 1.15.0 is used with all settings left at their defaults. Further details of the SUMO's GLOSA implementation can be found in SUMO's documentation [224].

6.4 Fixed Time Benchmark

At this point, the testbed and updated framework has been built and GLOSA implemented, however as with the experiments performed on the isolated junction testbed, a benchmark to compare the performance of the framework against is required. To ensure that this benchmark allows for a meaningful comparison, the software used for the benchmark must meet certain criteria. Firstly, the software must be designed to operate or optimise signal plans for arterial flow with both signalised junctions and non-signalised junctions. Secondly, the software must have accessible documentation. Thirdly, the software must be used in industry. LinSig V3.2.44.1 met these criteria and was chosen due to availability.

Using LinSig, a fixed time signal plan was created for the arterial with a 90-second cycle time. LinSig was also used for the calibration of the traffic scenarios and PRC calculations. More details on how this was done can be found in the LinSig 3.2 User Guide [226].

7 Arterial Flow Experiments

In the previous chapter an arterial flow testbed was created, an updated joint control framework for non-autonomous vehicles that uses reinforcement learning traffic control was designed and implemented, GLOSA was incorporated, and a fixed time benchmark was identified. However, in order to continue addressing the aims and objectives of this thesis and gain a better understanding of how a RLTC based joint control framework designed for non-autonomous vehicles will behave on an arterial flow, experimental results must be acquired which demonstrate the performance of a joint control framework designed for non-autonomous vehicles on an arterial flow.

To do this, some experimental design work must be undertaken. As stated in methodology, the experimental data required pertains to the performance of the joint control framework on arterial flows, and more specifically demonstrating that the joint control framework out performs benchmark systems and quantifying how the joint control framework will perform if the penetration rates of GLOSA at the site of deployment are different from those in the training scenario or if the joint control framework is not made aware of the routes vehicles are taking. The experiments must also include the training of multiple teams of agents. Therefore, four experiments, numbered 3 to 6, will be needed.

In experiment three, a number of groups of RLTC agents at a range of TPRs will be trained. As before agents were trained with TPRs of 0%, 10%, 20%, 40%, 60%, 80%, 100% in each traffic scenario (55%, 65%, 75% saturation) for 200 episodes each, in which the agents control the signalised junction for thirty simulation minutes.

However, an extra 3-minute period, where the junctions on the arterial operate on the fixed time control scheme, is introduced at the beginning of each episode. This allows vehicles to be spread throughout the network when training starts in each episode, and therefore, agents do not repeatedly experience extended periods of low traffic flow at the beginning of each episode.

Also, to counter instances of rapid un-learning, agents being trained are routinely evaluated and are saved if they are showing improvements or reverted if not.

This was sufficient to produce trained agents that met or exceeded the performance of the fixed time benchmarks could be used in the fourth experiment.

For each junction, TPR, and scenario, multiple agents were trained with Bayesian Optimisation, which was used to select the best-performing learning rates which would produce the agents with the lowest loss. Agents were trained separately in an environment where the other junctions controllers follow the fixed time benchmark plan.

After training, the best performing agents are evaluated, with EPR equal to TPR, over 100 episodes, each 30 simulation minutes long, and the speeds, number of stops, and waiting time of all vehicles will be monitored, which will be sufficient to validate the performance of the joint control framework against the benchmark system, and provide the experimental data required to address the first gap in knowledge for arterial flow scenarios.

Experiment four will evaluate the agents, which were trained and passed evaluation in experiment three, in scenarios with EPRs that are different from their TPRs on the arterial flow testbed. Evaluation will last for fifty episodes, each thirty simulation minutes long, and the speeds, stopping times, and junction entry speeds of all vehicles will again be monitored which will be sufficient to provide the results required to address the second gap in knowledge for arterial flow scenarios.

However, experimental data is also needed to assess how a joint control framework that uses reinforcement learning traffic control will perform if vehicles do not provide their routes to the joint control framework in advance. Therefore, two further experiments, numbered 5 and 6, will be completed simultaneously.

In experiment five, a number of groups of cooperative RLTC agents will be trained on the arterial network testbed at a range of TPRs without vehicles providing any information about their intended route. Again agents will be trained at TPRs of 0%, 10%, 20%, 40%, 60%, 80%, 100% in each traffic scenario (55%, 65%, 75% saturation) for 200 episodes each, in which the agents control the signalised junction for 30 simulation minutes.

However, an extra 3-minute period, where the junctions on the arterial operate on the fixed time control scheme, is introduced at the beginning of each episode. This allows vehicles to be spread throughout the network when training starts in each episode, and therefore, agents do not repeatedly experience extended periods of low traffic flow at the beginning of each episode.

Again, to counter instances of rapid un-learning, agents being trained are routinely evaluated and are saved if they are showing improvements or reverted if not.

This was sufficient to produce trained agents that met or exceeded the performance of the fixed time benchmarks and could be used in the sixth experiment. Again, this will be done with Bayesian Optimisation for each junction, TPR, and scenario.

Then, After training the best performing agents will be evaluated on an arterial network testbed with the EPR set equal to the TPR and compared to a fixed time benchmark. This step will provide the results required to address the third gap in knowledge.

Experiment six will evaluate the agents, which were trained and passed evaluation in experiment five, in scenarios with EPRs that are different from their TPRs on the arterial flow testbed. Evaluation will last for 50 episodes, each thirty simulation minutes long, and the speeds, stopping times, and junction entry speeds of all vehicles will be monitored which will be sufficient to provide the results required to address the second gap in knowledge for arterial flow scenarios.

7.1 Experiment 3 - 55% Scenario

Firstly, compared to the fixed time benchmark in the 55% traffic density scenario, the framework with the TPR and EPR set to 0%, decreased average vehicle waiting times and average journey stops by 65% and 33% and increased average vehicle speeds by 32%. This validates the performance of the joint control framework is above that of the benchmark system.

Also, agents that were trained at higher TPRs were found to perform better than those trained at lower TPRs, with the 100% TPR agent decreasing average vehicle waiting times and average journey stops by 11% and 3% compared to the 0% TPR agent. Average speeds also increased by 12%. This suggests that GLOSA was effective in this scenario. However, while the increase in vehicle speeds was quite linear (see figures 30), waiting times and journey stops did not improve until TPRs of 80% and 40% were reached, respectively (see figures 31 and 32). Some of this can be explained by GLOSA being less effective because unequipped vehicles would be sometimes blocking equipped vehicles. However, that alone does not explain why, for TPRs of 20% and below, waiting times and journey stops increased by as much as 6% and 1.2% compared to the 0% TPR agent.

Instead, a better explanation is that these agents were unable to reliably determine the best action to take because they did not know if any given vehicle was equipped. This led agents with TPRs between 10% and 60% to make a lot of guesses about how vehicles might behave, leading to an increase in vehicles failing to clear the junction without stopping. This was also the case with the 80% TPR agents, but by that point the positive effects of GLOSA fully outweighed this effect.

Overall, GLOSA was clearly effective within the framework in this scenario at higher TPRs, but only somewhat effective at lower (non-zero) TPRs. Furthermore, these results were achieved with the simulations running faster than real time on a single desktop computer with a consumer CPU and GPU, meaning that computation is not a barrier to the real-world deployment of this framework.

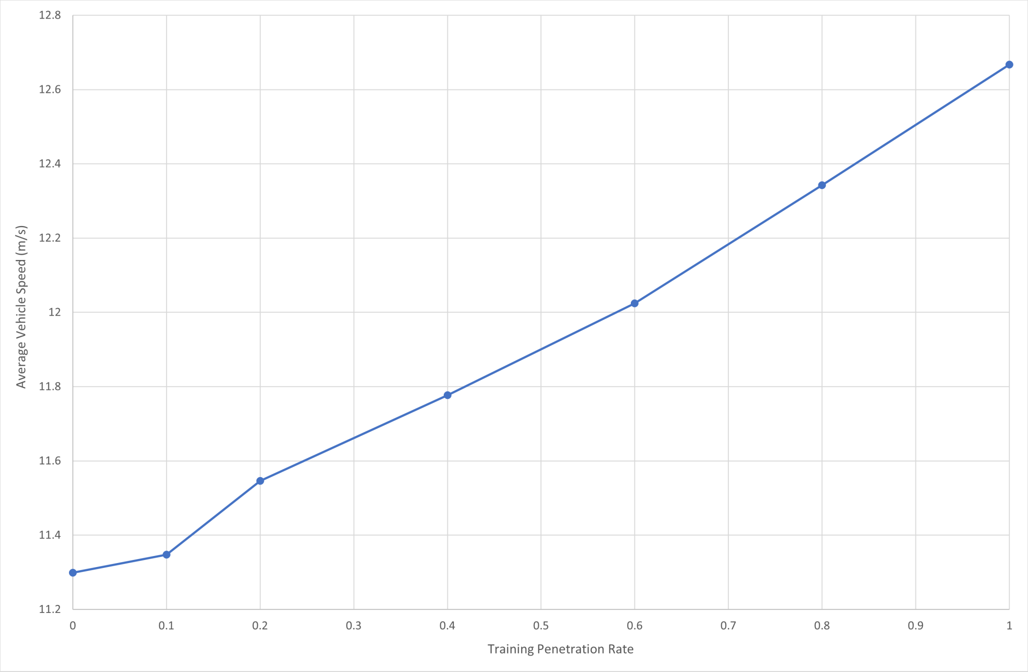

Figure 30: Average vehicle speeds in experiments by TPR at 55% traffic density

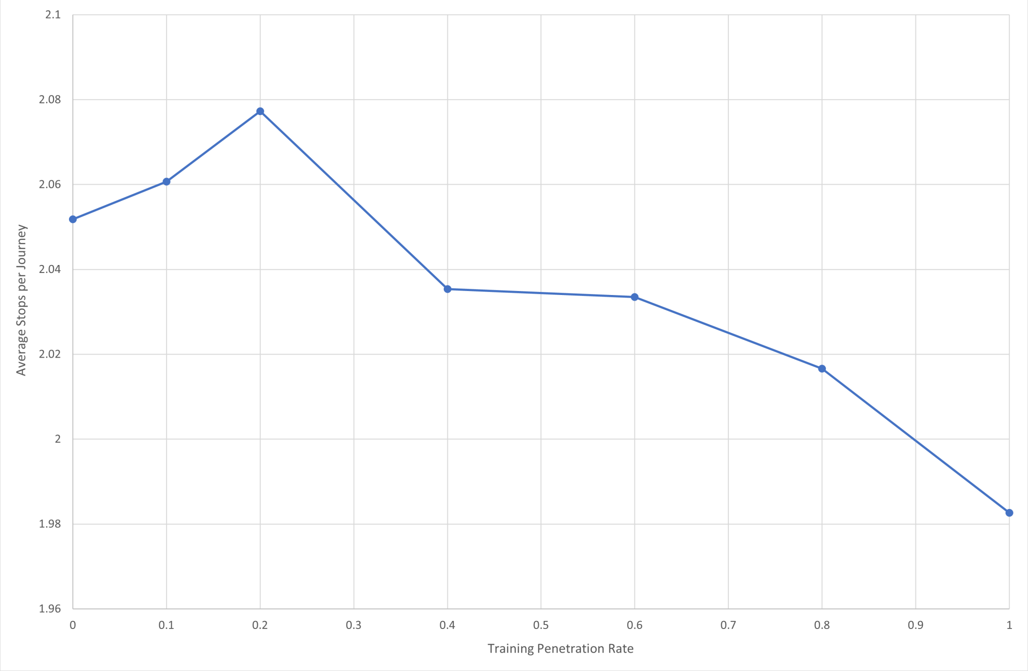

Figure 31: Average journey stops in experiments by TPR at 55% traffic density

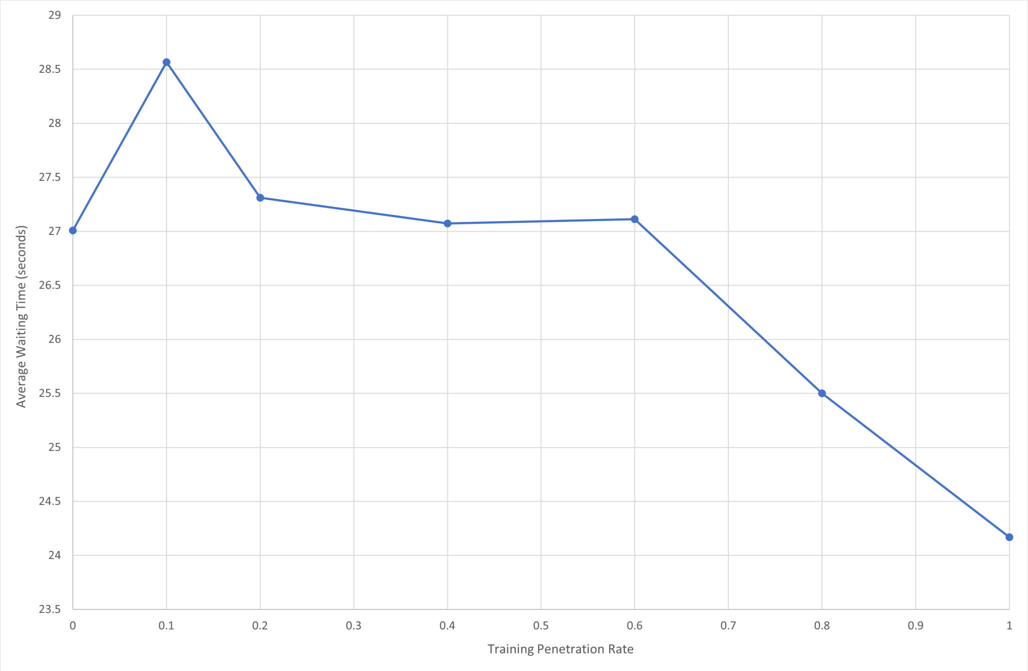

Figure 32: Average vehicle waiting time in experiments by TPR at 55% traffic density

A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

7.2 Experiment 3 - 65% Scenario

Here again, the results validated the performance of the joint control framework is above that of the benchmark system. Compared to the fixed time benchmark, in the 65% traffic density scenario, the framework, with the TPR and EPR set to 0%, decreased average vehicle waiting times and average journey stops by 60% and 26% and increased average vehicle speeds by 33%.

Also, as before agents that were trained at higher TPRs were found to perform better than those trained at lower TPRs, with the 100% TPR agent decreasing average vehicle waiting times and average journey stops by 20% and 5% compared to the 0% TPR agent. Average speeds also increased by 14%. This again validates that GLOSA was clearly effective within the framework at higher TPRs.

However, the same issue with the agents being unable to reliably determining the best action was available again here although to a lesser degree as in this scenario both the increase in vehicle speeds and decrease in waiting times were quite linear (see figures 33 and 35). Only the journey stops did not improve until TPRs of 60% were reached (see figure 34). For TPRs below 60%, journey stops increased by as much as 0.4% compared to the 0% TPR agent.

Overall, GLOSA was clearly effective within the framework in this scenario at higher TPRs, but only somewhat effective at lower (non-zero) TPRs.

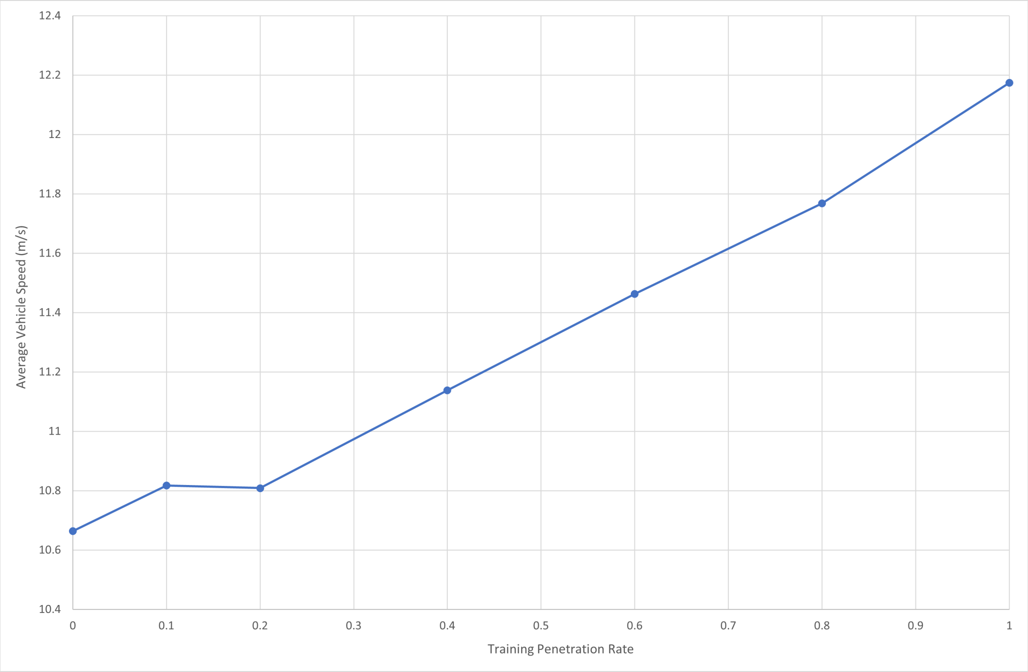

Figure 33: Average vehicle speeds in experiments by TPR at 65% traffic density

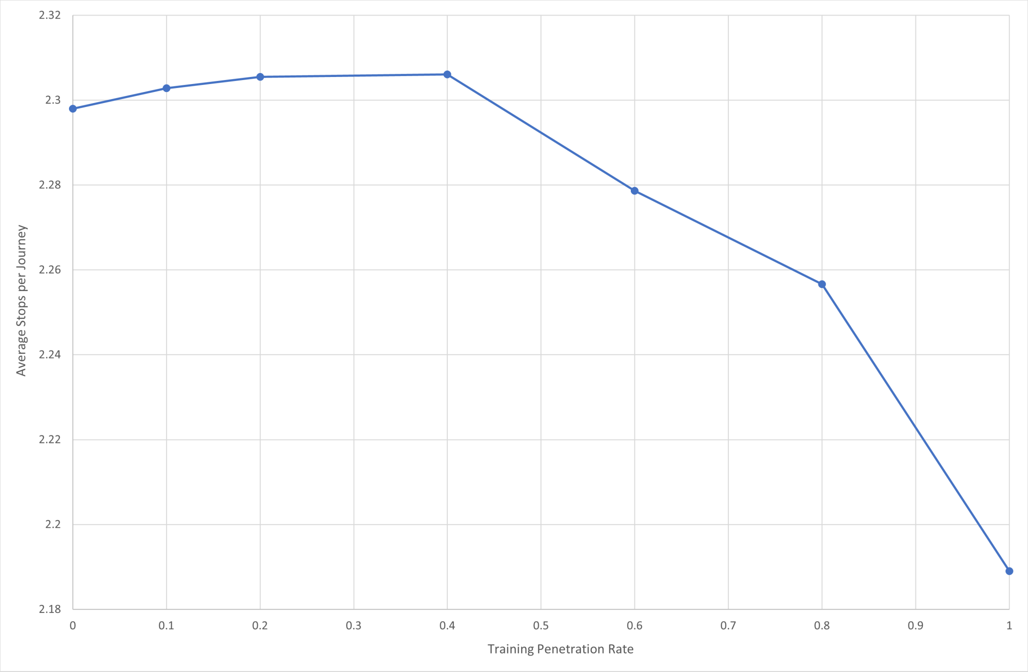

Figure 34: Average journey stops in experiments by TPR at 65% traffic density

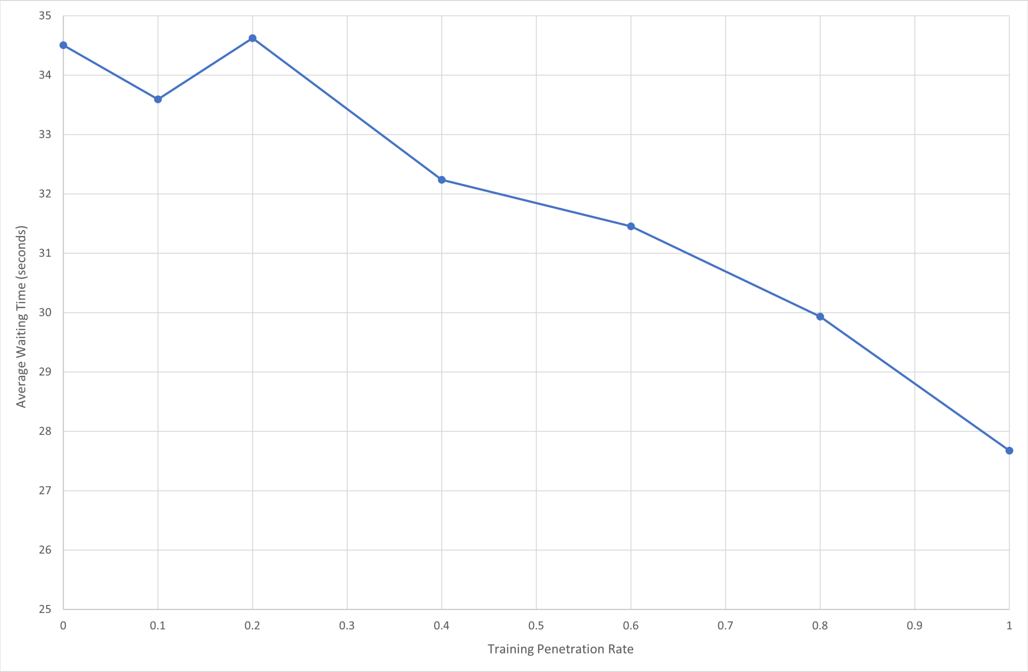

Figure 35: Average vehicle waiting time in experiments by TPR at 65% traffic density

A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

7.3 Experiment 3 - 75% Scenario

Here all the previous trends continued again. Compared to the fixed time benchmark, in the 75% traffic density scenario, the framework, with the TPR and EPR set to 0%, decreased average vehicle waiting times and average journey stops by 54% and 27% and increased average vehicle speeds by 27%. Again, the results validated the performance of the joint control framework is above that of the benchmark system.

Agents that were trained at higher TPRs were found to perform better than those trained at lower TPRs, with the 100% TPR agent decreasing average vehicle waiting times and average journey stops by 20% and 4% compared to the 0% TPR agent. Average speeds also increased by 15% (see figure 36). This again validates that GLOSA was clearly effective within the framework at higher TPRs.

However, the same issue with the agents being unable to reliably determining the best action was available again here although to a lesser degree as in this scenario both waiting times and journey stops didn't meaningfully improve until TPRs of 60% and 40% were reached, respectively (see figure 38 and 37).

Overall, GLOSA was clearly effective within the framework in this scenario at higher TPRs, but only somewhat effective at lower (non-zero) TPRs.

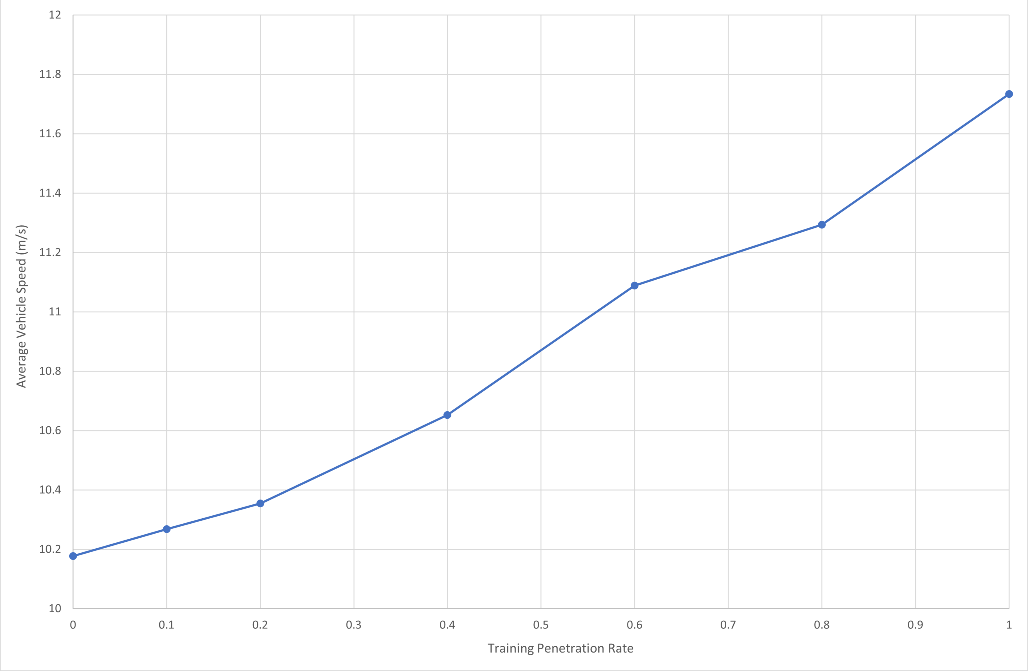

Figure 36: Average vehicle speeds in experiments by TPR at 75% traffic density

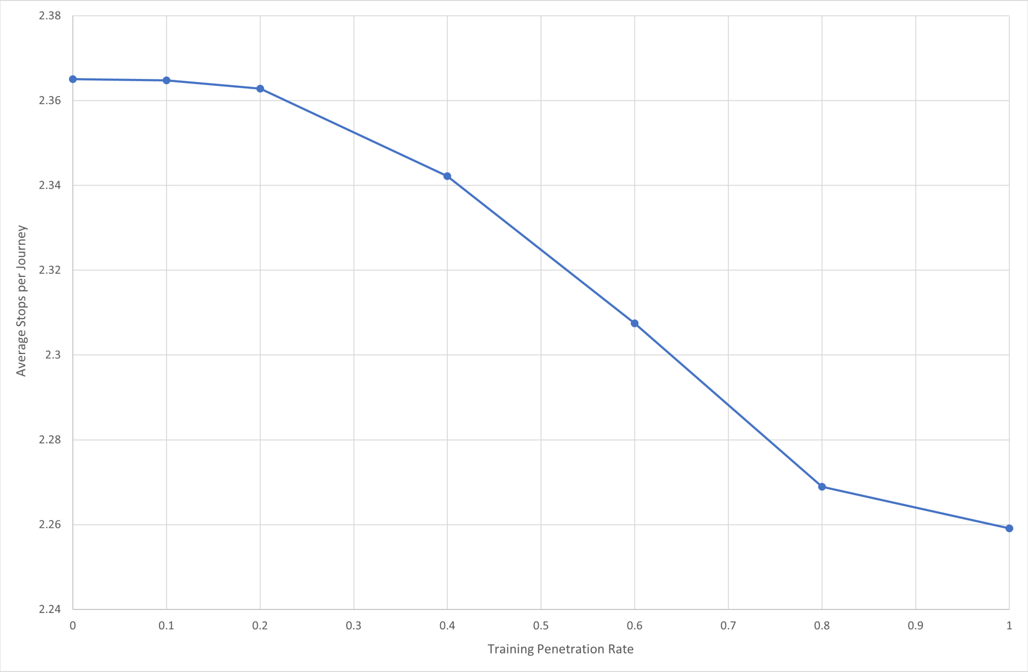

Figure 37: Average journey stops in experiments by TPR at 75% traffic density

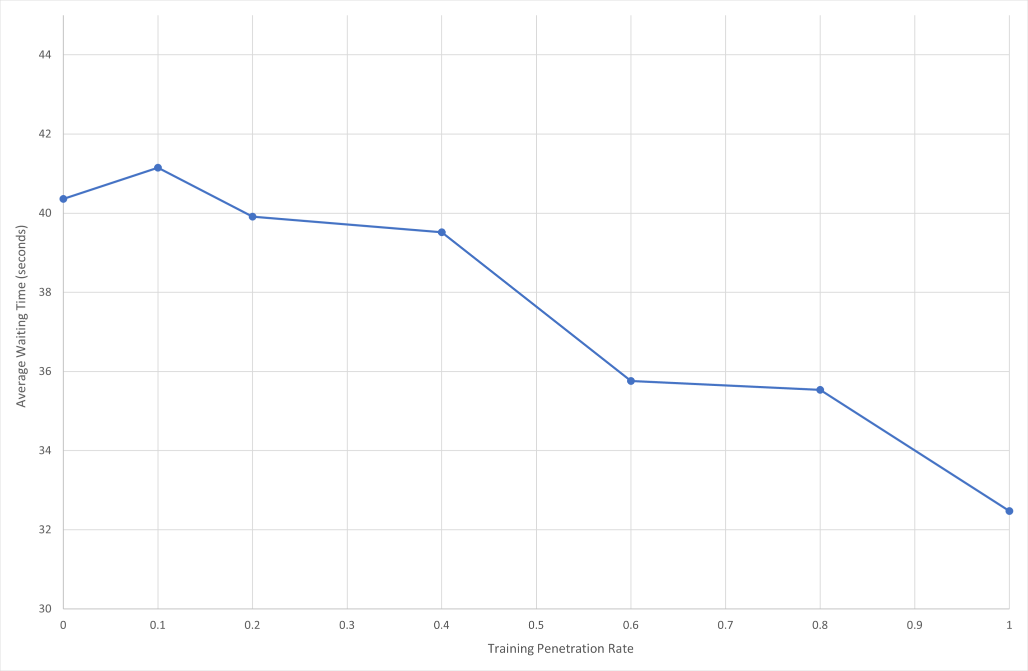

Figure 38: Average vehicle waiting time in experiments by TPR at 75% traffic density

A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

7.4 Experiment 3 - Discussion

Overall, the framework outperformed the fixed time benchmark in all tested scenarios. Also, agents with higher TPRs outperformed those with lower TPRs. However, a small but not 0% TPR sometimes led to an agent that performed worse on the metrics of journey stops and waiting times than the 0% TPR agent because uncertainty about the future behaviour of approaching vehicle forced the framework into less accurate decisions.

The positive results in all scenarios demonstrate that this joint control framework was able to have a positive effect on driving smoothness on the arterial, with drivers able to slow down smoother when approaching red lights, eliminating stops. Also, thanks to advanced warning of green phases from the GLOSA system, equipped vehicles were able to speed up earlier and clear the junction faster, improving the junctions efficiently, even at the higher traffic densities tested here. Overall, these results demonstrate that a RLTC system based joint control framework can have benefits when applied to Connected Non-Autonomous Vehicles (CNAVs) on arterials.

Two factors are thought to have positively impacted the results here. Firstly, the use of Bayesian Optimisation and the other upgrades made to the reinforcement learning optimisation led to better performance from the trained agents. Secondly, while the increased base phase length for phases serving vehicles on the arterial may have reduced the flexibility of the RLTC system, the increase in the availability of future signal plans allowed a much larger number of equipped vehicles to pass junctions without stopping and, therefore, somewhat negated the issues seen with queueing vehicles blocking the junction in previous experiments.

Importantly, this result also validates this framework and approaches like it for a wide range of junction and arterial flow geometries, as all the variables used can always be re-tuned to fit new situations.

7.5 Experiment 4 - 55% Scenario

In the 55% traffic density scenario, on the metric of average journey stops, only the agents trained with a TPR of 40% or higher improved as the EPR increased (see figure 40). The performance of the agents trained at lower TPRs degraded quite badly as EPRs increased on this metric, with increases in stops of up to 11% observed. Just like in the previous framework, the cause of this was that the agents made decisions expecting vehicles to not be equipped with GLOSA and therefore often choose to priorities existing queues over approaching vehicles, as it could not trust that approaching vehicles were GLOSA equipped and would clear the junction without stopping but it could guarantee that the queue would be cleared. It is possible that this issue could be address with a different state space or reward function, which allowed the system to better determine a vehicles likelihood to clear the junction without stopping and earn a greater reward when one did.

For the policies with TPRs at or above 40% there were no negative effects of the introduction of more GLOSA equipped vehicles. The cause of this is that at these higher TPRs, it made sense for the agent to gamble that a vehicle was GLOSA equipped and that it would clear the junction without stopping. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues.

On the other metrics, waiting times and average speeds, all agents performed better if the EPR was increased. This was especially true for agents trained at higher TPRs, which tended to make much larger gains as the EPRs increased. For example, when evaluated with a 100% EPR, the agents trained with 80% and 100% TPR were able to reduce waiting times by 20% and 19% respectively and increase average speeds by 12% compared to when they were evaluated with a 0% EPR. The agents trained with 0% and 10% TPR could only achieve a reduction of 9% in waiting times and an increase of 10% in average speeds (see figures 41 and 39).

This would also be in keeping with the understanding that has so far been built up where higher TPRs lead to policies that were more likely to gamble that approaching vehicles were GLOSA equipped and would therefore clear the junction without stopping or avoid becoming part of the queue for as long as possible. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues.

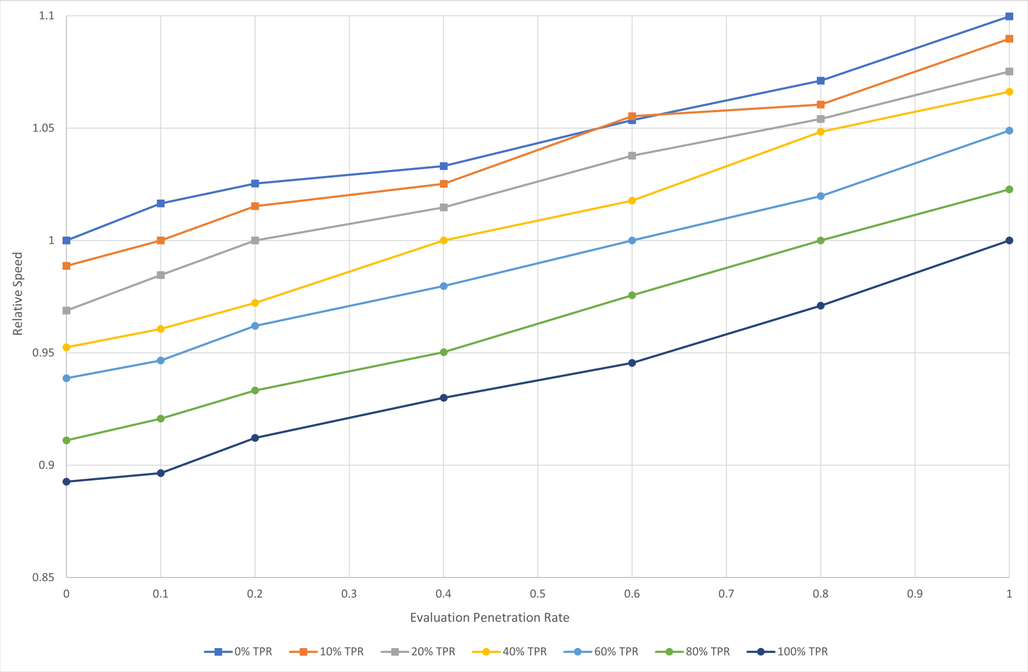

Figure 39: Relative performance on the metric of average vehicle speeds in experiments by EPR at 55% traffic density

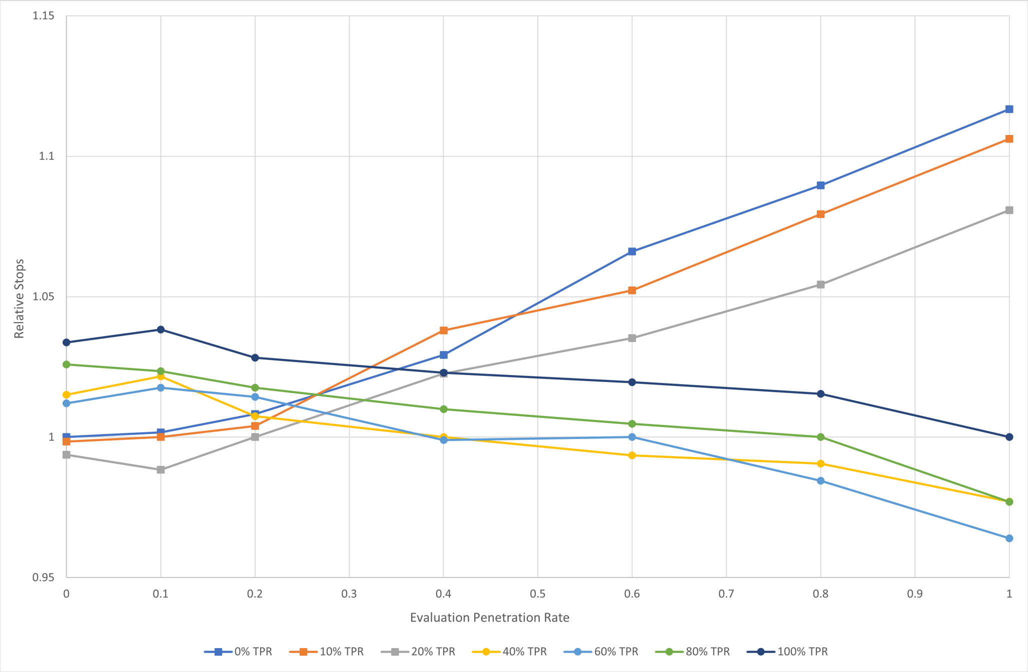

Figure 40: Relative performance on the metric of average journey stops in experiments by EPR at 55% traffic density

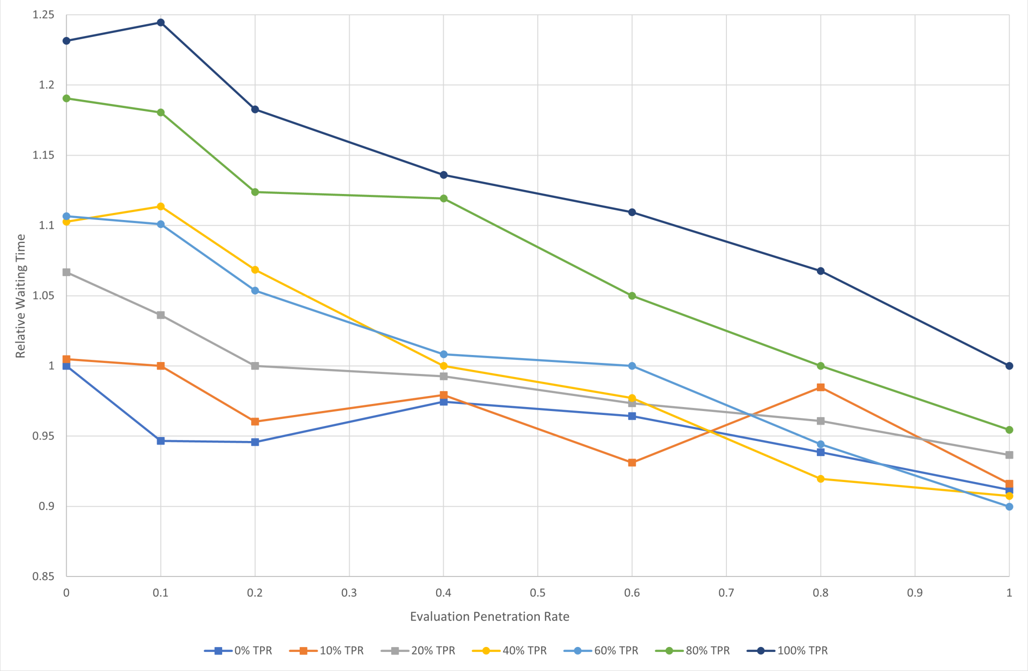

Figure 41: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 55% traffic density

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

7.6 Experiment 4 - 65% Scenario

In the 65% traffic density scenario, all agents performed better on the metrics of waiting times and average speeds if the EPR was increased (see figures 42 and 44). Again, this was especially true for agents trained at higher TPRs, which tended to make much larger gains as the EPRs increased. For example, when evaluated with a 100% EPR, the agents trained with 80% and 100% TPR were able to reduce waiting times by 27% and increase average speeds by 16% compared to when they were evaluated with a 0% EPR. The agents trained with 0% and 10% TPR could only achieve a reduction of 11% and 9% in waiting times and an increase of 11% in average speeds.

On the metric of average journey stops, only the agents trained with a TPR of 60% or higher improved as the EPR increased (see figure 43). The performance of the agents trained at TPRs lower than 40% degraded as EPRs increased on this metric with increases in stops of up to 8% observed. The performance of the agent trained with a TPR of 40%, on this metric, never deviated by more than a percent regardless of the EPR. Most of these results are in keeping with the current understanding that higher TPRs lead to policies more likely to gamble that cars are GLOSA equipped (see section 7.5)..

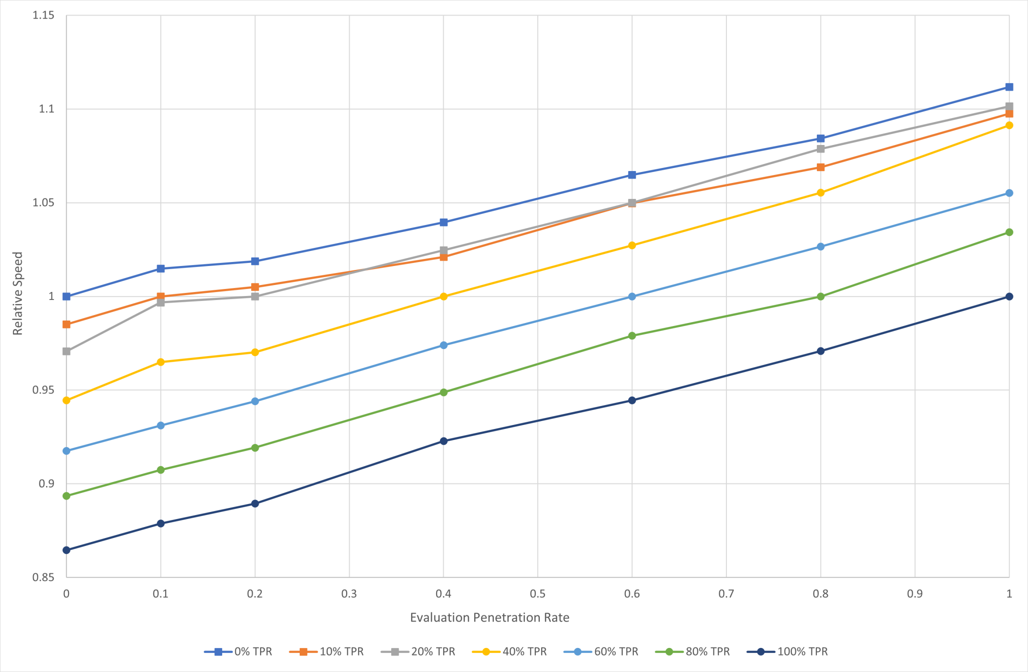

Figure 42: Relative performance on the metric of average vehicle speeds in experiments by EPR at 65% traffic density

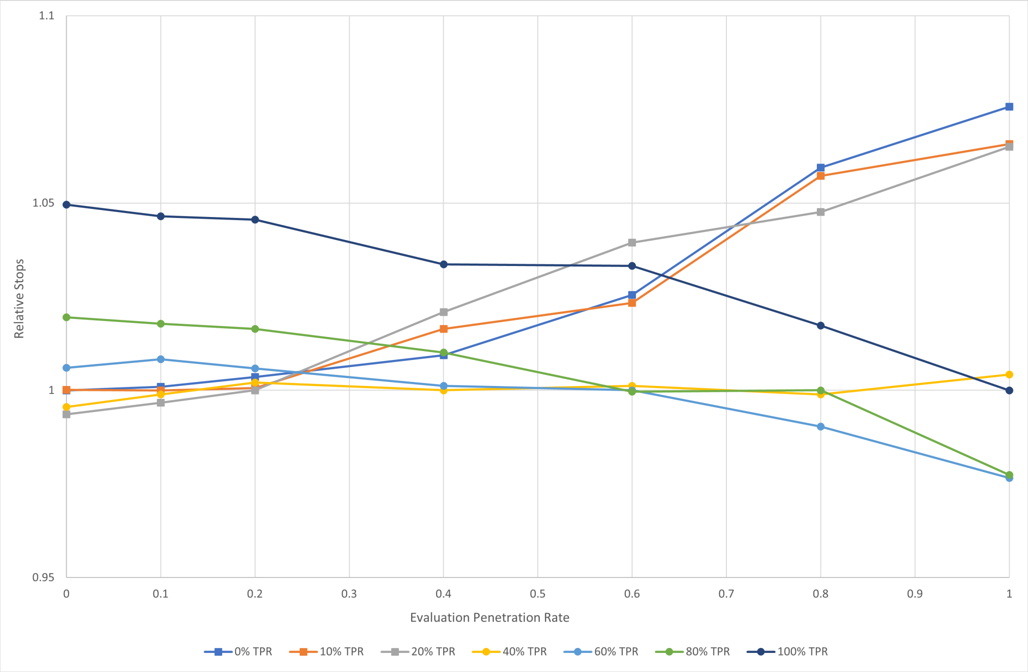

Figure 43: Relative performance on the metric of average journey stops in experiments by EPR at 65% traffic density

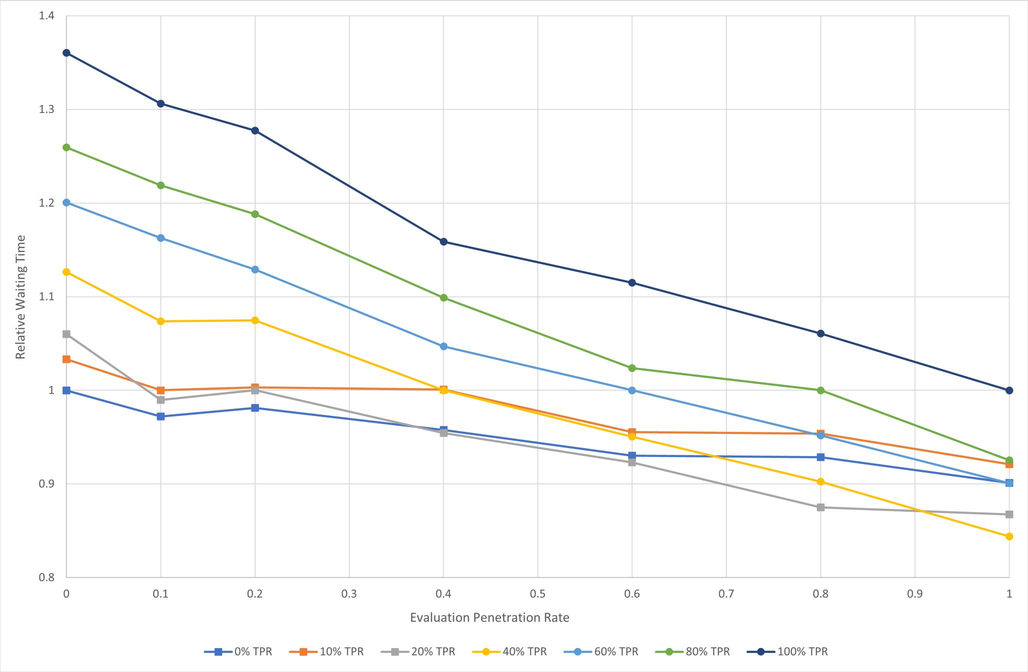

Figure 44: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 65% traffic density

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

7.7 Experiment 4 - 75% Scenario

In the 75% traffic density scenario, all agents performed better on the metrics of waiting times and average speeds if the EPR was increased. Again, this was especially true for agents trained at higher TPRs, which tended to make much larger gains as the EPRs increased. For example, when evaluated with a 100% EPR, the agents trained with 80% and 100% TPR were able to reduce waiting times by 32% and increase average speeds by 21% compared to when they were evaluated with a 0% EPR (see figure 45 and 47). The agents trained with 0% and 10% TPR could only achieve a reduction of 14% in waiting times and an increase of 13% in average speeds.

On the metric of average journey stops, only the agents trained with a TPR of 40% or higher improved as the EPR increased (see figure 46). The performance of the agents trained at TPRs lower than 40% degraded as EPRs increased on this metric with increases in stops of up to 8% observed. The performance of the agent trained with a TPR of 40% on this metric did improve as the EPR increased, but it only achieved a 2% reduction in stops. Again most of these results are in keeping with the current understanding that higher TPRs lead to policies more likely to gamble that cars are GLOSA equipped (see section 7.5).

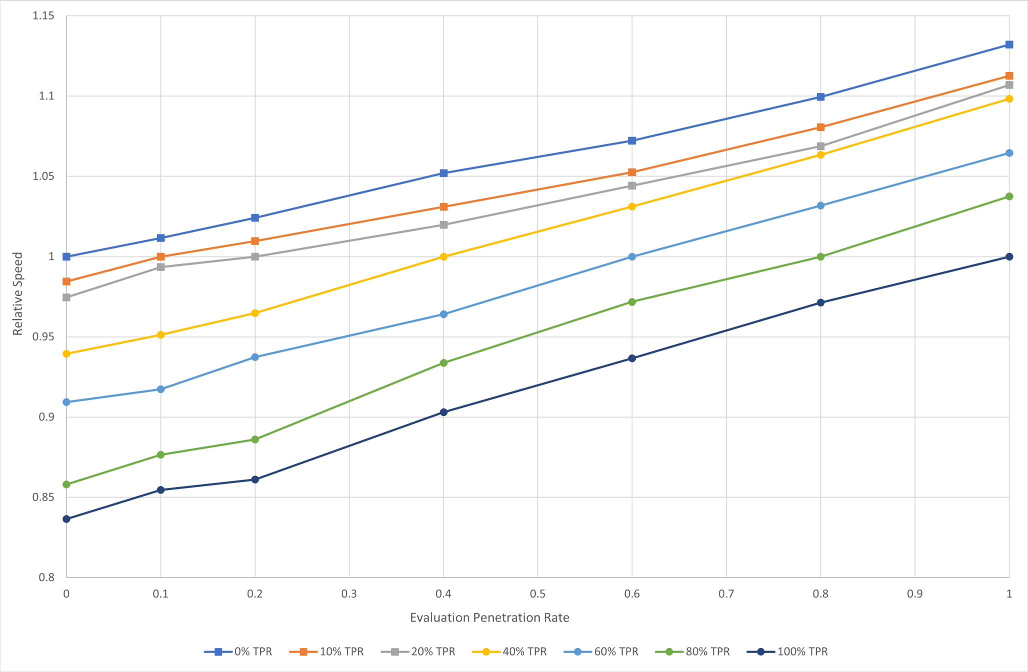

Figure 45: Relative performance on the metric of average vehicle speeds in experiments by EPR at 75% traffic density

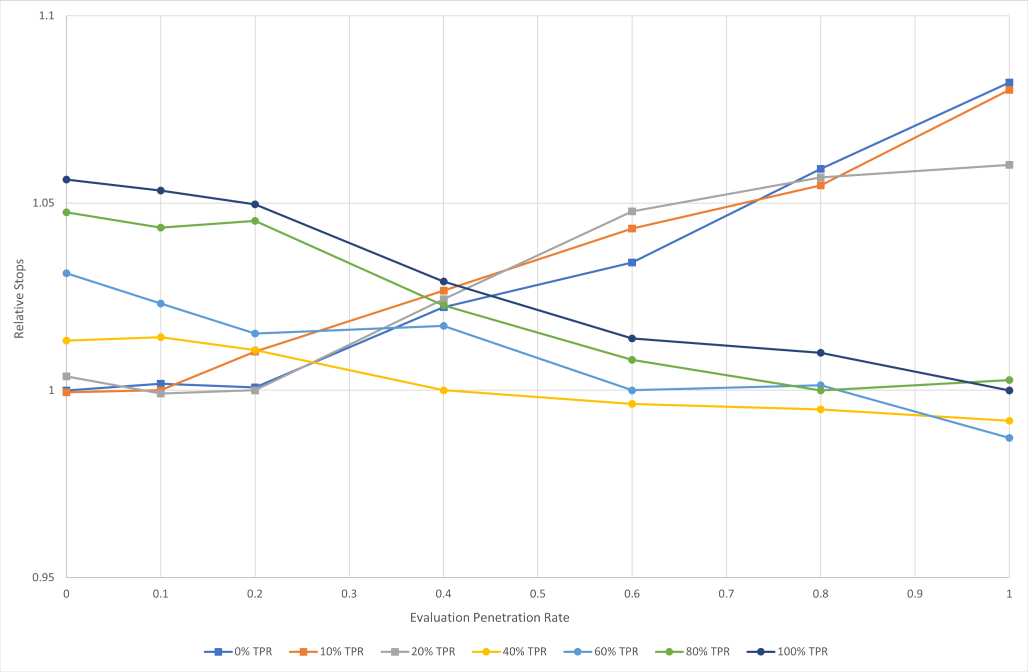

Figure 46: Relative performance on the metric of average journey stops in experiments by EPR at 75% traffic density

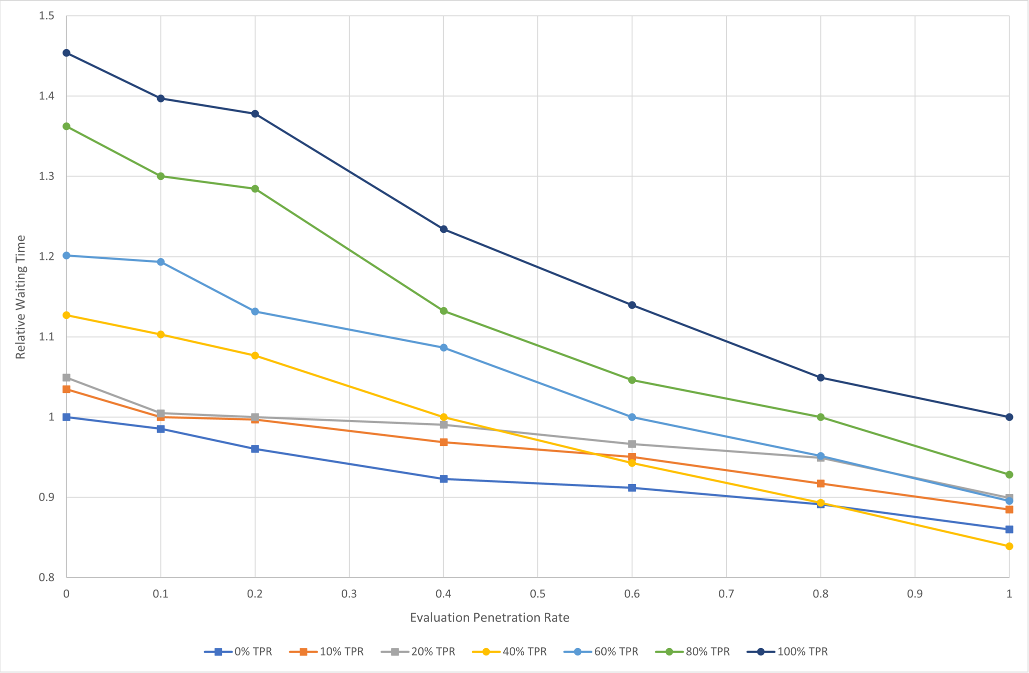

Figure 47: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 75% traffic density

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

7.8 Experiment 4 - Discussion

The results for all scenarios, once again, suggest that divergent values for the TPR and EPR can lead to poorer performance for the joint control framework. As before, this is likely caused by the unexpected change in vehicle behaviour, causing the agents' learned predictions of action value to be incorrect and leading to suboptimal decision-making; therefore, the benefits of having more equipped vehicles are outweighed by the drawback of the RLTC system not expecting the change in driving behaviour.

One difference from the previous results is that the TPR required for higher EPRs to have positive effects appears to have increased from 20% to 40%. That seems to have been caused by the short spacing between the junctions and the increased base phase lengths, which gave the RLTC system less time and fewer chances to react and made the accuracy of its decisions more important.

Overall, while all agents to improve on the metrics of waiting time and average speed as the EPR increased, agents trained with a TPR of 40% or less tended to perform better on the metric of stops at lower EPRs and tended to gain less on the metrics of waiting time and average speed as the EPR increased, while agents trained with a TPR of 60% or more tended to perform better on the metric of stops at higher EPRs and made larger gains on the metrics of waiting time and average speed as the EPR increased.

7.9 Experiment 5 - 55% Scenario

Compared to the fixed time benchmark, in the 55% traffic density scenario, the framework, with the TPR and EPR set to 0%, decreased average vehicle waiting times and average journey stops by 61% and 29% and increased average vehicle speeds by 29%. However, compared to the agents trained with access to route information, wait times and journey stops were increased by up to 16% and 5% respectively, while average speed decreased by up to 3%.

Again, agents that were trained at higher TPRs were found to generally perform better than those trained at lower TPRs, with the 100% TPR agent decreasing average vehicle waiting times and average journey stops by 7% and 5% compared to the 0% TPR agent. Average speeds also increased by 11%. However, while the increase in vehicle speeds and decrease in journey stops was quite linear (see figure 50 and 49), waiting times didn't improve until TPRs of 40% were reached (see figure 50), and most of the improvement seen occurred once TPRs of 80% were reached. In fact, for TPRs of 20% and below, waiting times increased by as much as 3% compared to the 0% TPR agent. As was concluded in section 7.1, this is explained by a mix of GLOSA being less effective because unequipped vehicles would be sometimes blocking equipped vehicles, and the agents being unable to reliably determine the best action to take because they did not know if any given vehicle was equipped. This led agents with TPRs between 10% and 60% to make a lot of guesses about how vehicles behave, leading to an increase in vehicles failing to clear the junction without stopping. This was also the case with the 80% TPR agents, but by that point the positive effects of GLOSA fully outweighed this effect.

Overall, GLOSA was clearly effective within the framework in this scenario at higher TPRs, but only somewhat effective at lower (non-zero) TPRs.

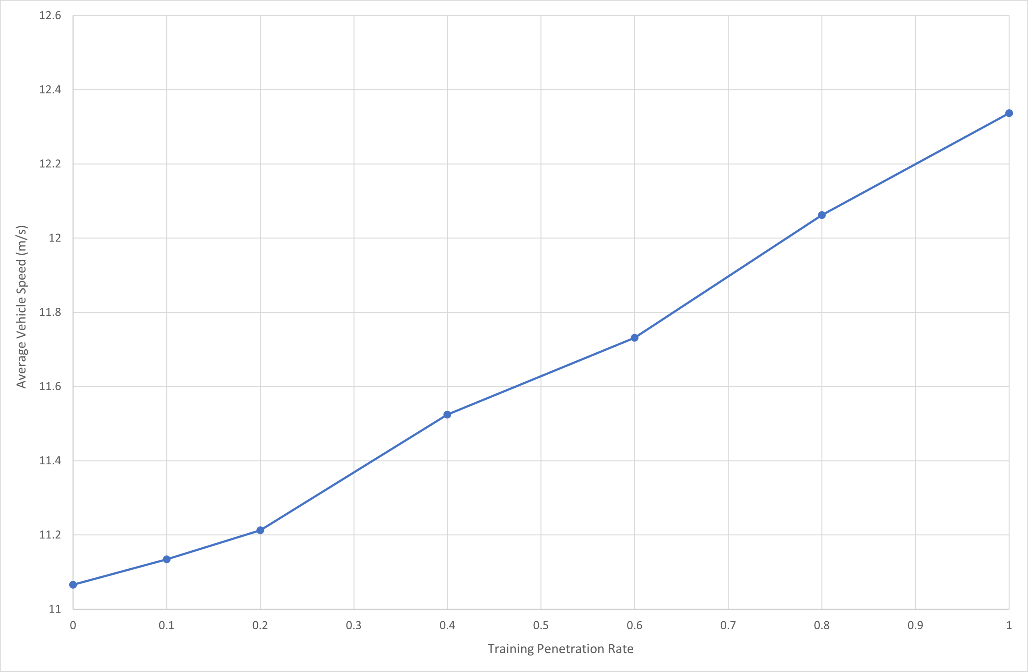

Figure 48: Average vehicle speeds in experiments by TPR at 55% traffic density with unknown route information

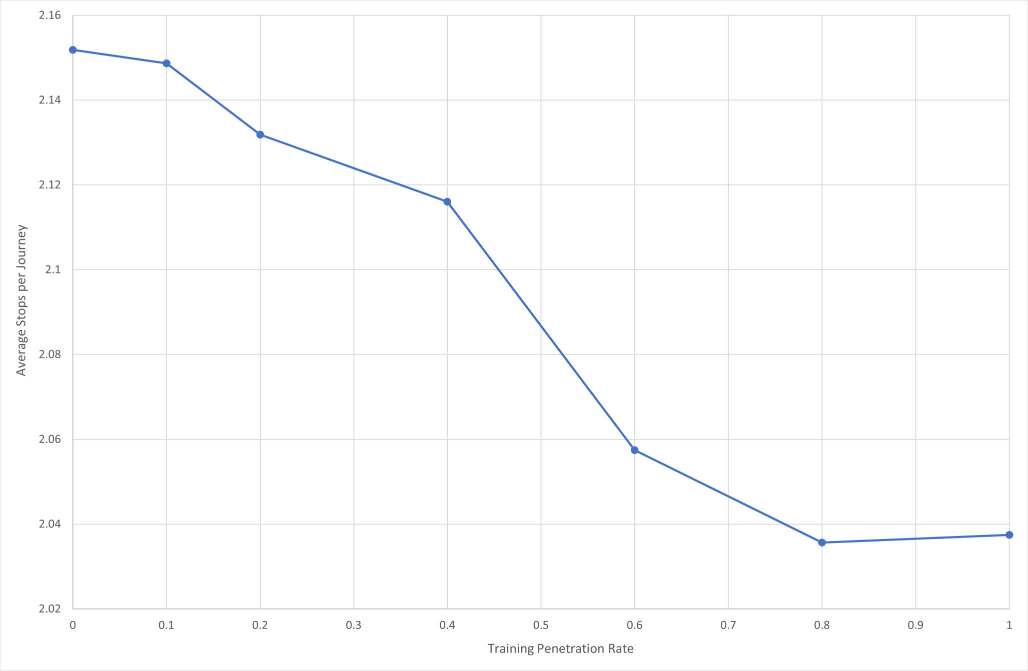

Figure 49: Average journey stops in experiments by TPR at 55% traffic density with unknown route information

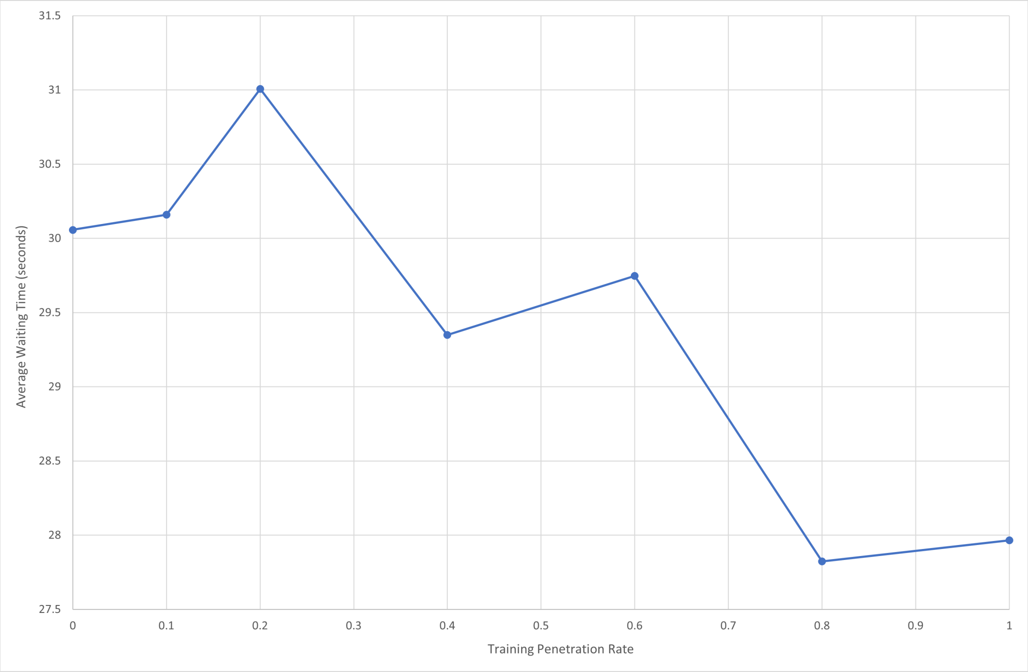

Figure 50: Average vehicle waiting time in experiments by TPR at 55% traffic density with unknown route information

A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

7.10 Experiment 5 - 65% Scenario

Compared to the fixed time benchmark, in the 65% traffic density scenario, the framework, with the TPR and EPR set to 0%, decreased average vehicle waiting times and average journey stops by 57% and 22% and increased average vehicle speeds by 31%. However, compared to the agents trained with access to route information, wait times and journey stops were increased by up to 17% and 5% respectively, while average speed decreased by up to 4%.

Agents that were trained at higher TPRs were found to perform better than those trained at lower TPRs, with the 100% TPR agent decreasing average vehicle waiting times and average journey stops by 12% and 7% compared to the 0% TPR agent. Average speeds also increased by 13%. In this scenario, the increase in vehicle speeds was quite linear (see figure 51). However, the agent trained with 10% TPR underperformed on this metric. Waiting times and journey stops did not decrease reliably until TPRs of 60% were reached (see figure 53 and 52).

These results are in keeping with the current understanding that, at lower TPRs, GLOSA is less effective, and the agents are unable to reliably determine the future behaviour of the vehicles (see section 7.1). Overall, though, GLOSA was clearly effective within the framework in this scenario at higher TPRs, but only somewhat effective at lower (non-zero) TPRs.

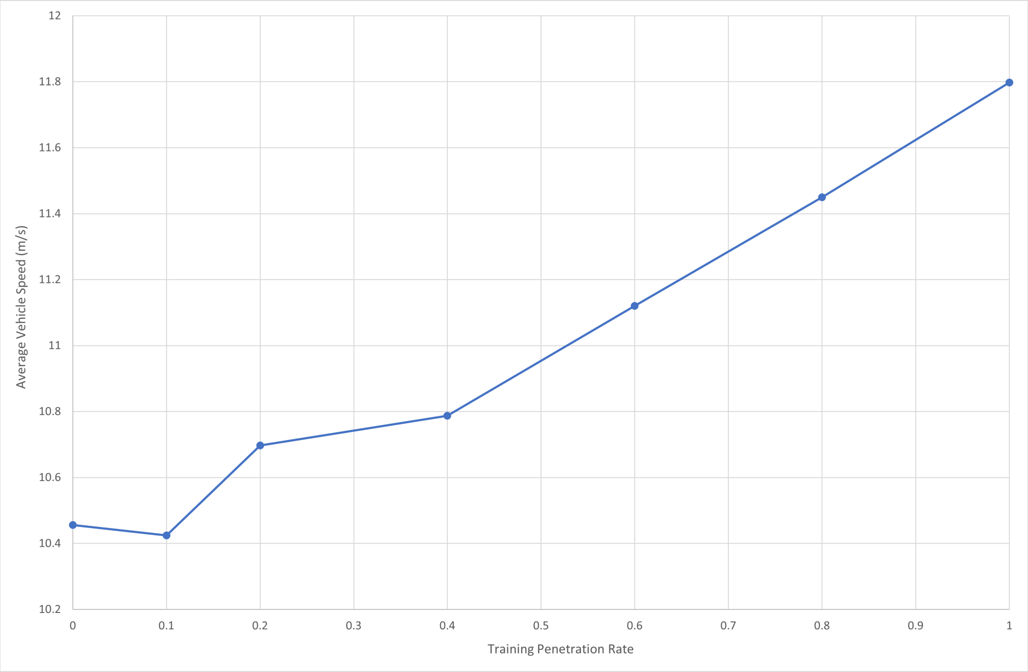

Figure 51: Average vehicle speeds in experiments by TPR at 65% traffic density with unknown route information

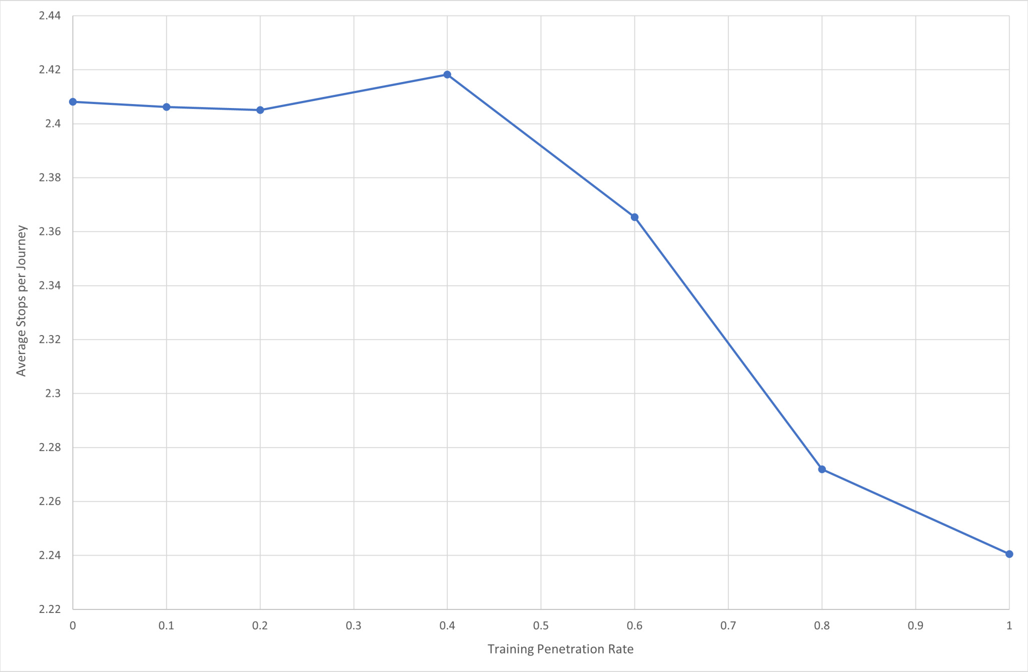

Figure 52: Average journey stops in experiments by TPR at 65% traffic density with unknown route information

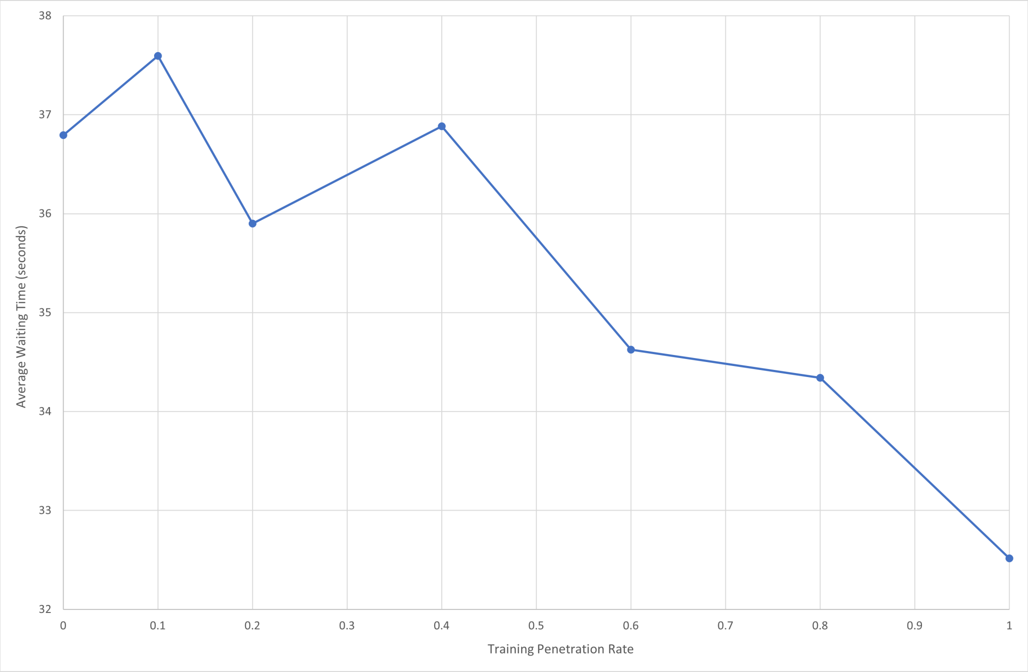

Figure 53: Average vehicle waiting time in experiments by TPR at 65% traffic density with unknown route information

A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

7.11 Experiment 5 - 75% Scenario

Compared to the fixed time benchmark in the 75% traffic density scenario, the framework, with the TPR and EPR set to 0%, decreased average vehicle waiting times and average journey stops by 50% and 24% and increased average vehicle speeds by 23%. However, compared to the agents trained with access to route information, wait times and journey stops were increased by up to 17% and 5% respectively, while average speed decreased by up to 4%.

Agents that were trained at higher TPRs were found to perform better than those trained at lower TPRs, with the 100% TPR agent decreasing average vehicle waiting times and average journey stops by 19% and 9% compared to the 0% TPR agent (see figure 56 and 55). Average speeds also increased by 16% 54s. In this scenario, the agent trained with TPRs of 20% or lower showed only mild improvement or no improvement in all metrics over agents trained with a TPR of 0%. After this point, improvements were more forth coming.

These results are in keeping with the current understanding that, at lower TPRs, GLOSA is being less effective, and the agents are unable to reliably determine the future behaviour of the vehicles (see section 7.1). Overall, though, GLOSA was clearly effective within the framework in this scenario at higher TPRs, but only somewhat effective at lower (non-zero) TPRs.

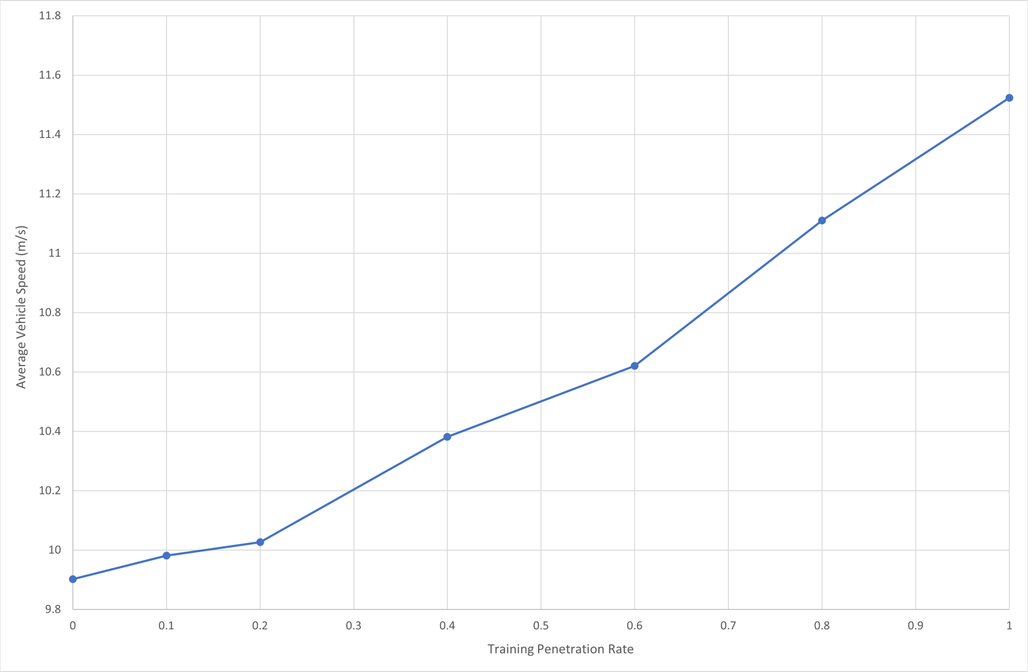

Figure 54: Average vehicle speeds in experiments by TPR at 75% traffic density with unknown route information

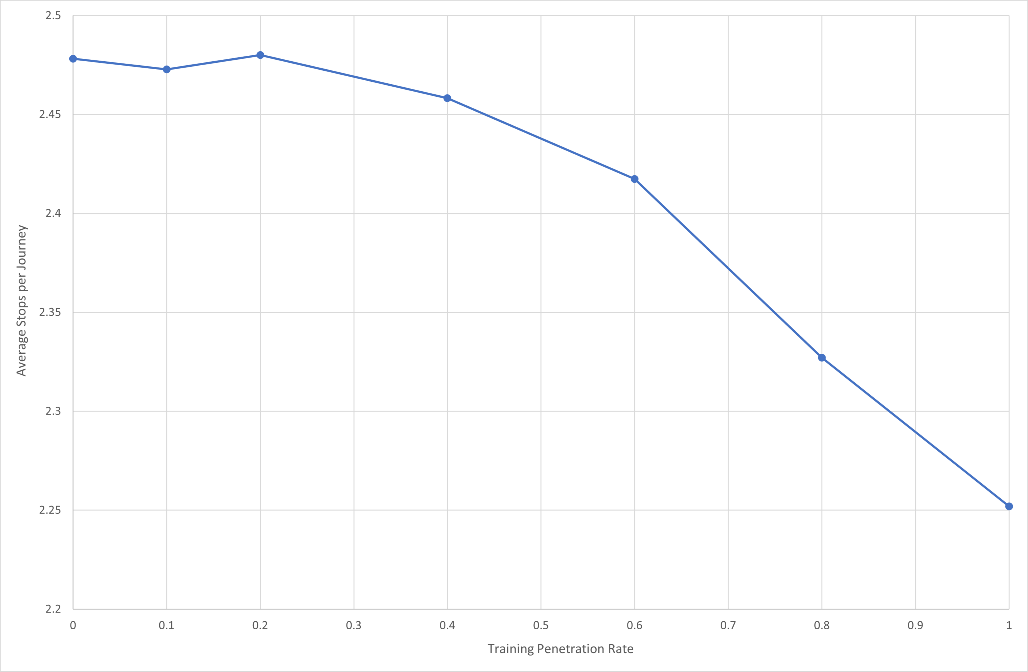

Figure 55: Average journey stops in experiments by TPR at 75% traffic density with unknown route information

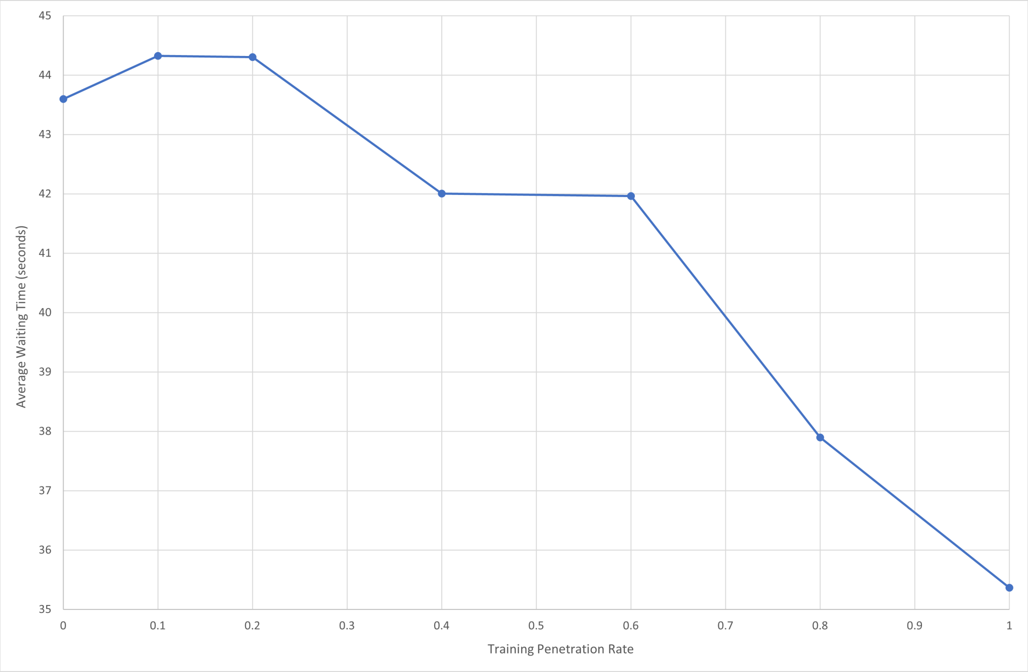

Figure 56: Average vehicle waiting time in experiments by TPR at 75% traffic density with unknown route information

A one-way ANOVA test was performed on these results for each metric, and it was found that the p values were all $\ll 1\%$ and therefore the changes in TPR did have statistically significant effect on the performance of the framework on all metrics. The test statistics can be found in Table 7.

7.12 Experiment 5 - Discussion

Overall, the framework outperformed the fixed time benchmark in all tested scenarios but was unable to perform as well as agents trained with route information, with wait times and journey stops increasing by up to 17% and 5%, respectively, while average speed decreased by up to 4%. As before, agents with higher TPRs outperformed those with Lower TPRs, and a small but not 0% TPR sometimes led to an agent that performed worse on some or all metrics. Typically, the TPR needed to exceed 40% or 60% for a meaningful improvement to present itself.

Results clearly indicate that at lower TRPs, GLOSA is less effective because unequipped vehicles often block equipped vehicles, and the agents being unable to reliably determine the best action to take because they did not know if any given vehicle was equipped. This led agents with TPRs between 10% and 60% to make a lot of guesses about how vehicles behave, leading to an increase in vehicles failing to clear the junction without stopping. This was also the case with the 80% TPR agents, but by that point the positive effects of GLOSA fully outweighed this effect.

The reduction in performance, compared to the framework when route information was available, was the expected outcome. Removing the route information partially blinds the RLTC system and makes it less able to model the effects of its actions. However, as before, the positive results in all scenarios demonstrate that this joint control framework was able to have a positive effect on driving smoothness on the arterial without route information, with drivers able to slow down smoother when approaching red lights, eliminating stops. Also, thanks to advanced warning of green phases from the GLOSA system, equipped vehicles were able to speed up earlier and clear the junction faster, improving the junctions efficiently. Overall, these results demonstrate that a RLTC system based joint control framework can have benefits when applied to Connected Non-Autonomous Vehicle (CNAVs) on arterials without route information.

7.13 Experiment 6 - 55% Scenario

In the 55% traffic density scenario, all agents performed better on the metrics of average speeds, and most agents performed better on the metrics of wait times if the EPR was increased. This was especially true for agents trained at higher TPRs, which tended to make much larger gains as the EPRs increased. For example, when evaluated with a 100% EPR, the agents trained with 60% and 80% TPR were able to reduce waiting times by 22% and 25% respectively and increase average speeds by 15% compared to when they were evaluated with a 0% EPR. this validates that the framework can in many cases manage increases in the EPR in this scenario.

However, the agents trained with 0% and 10% TPR could only achieve an increase of 9% in average speeds. These agents also struggled to reliably reduce wait times, with there never being a decrease larger than 5% and 2% respectively and cases where 0.3% and 0.4% increases were observed. On the metric of average journey stops, only the agents trained with a TPR of 40% or higher improved as the EPR increased. The performance of the agents trained at lower TPRs degraded quite badly as EPRs increased on this metric, with increases in stops of up to 10% observed.

This would be in keeping with the understanding that has so far been built up where higher TPRs lead to policies that were more likely to gamble that approaching vehicles were GLOSA equipped and would therefore clear the junction without stopping or avoid becoming part of the queue for as long as possible. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues (see section 7.5).

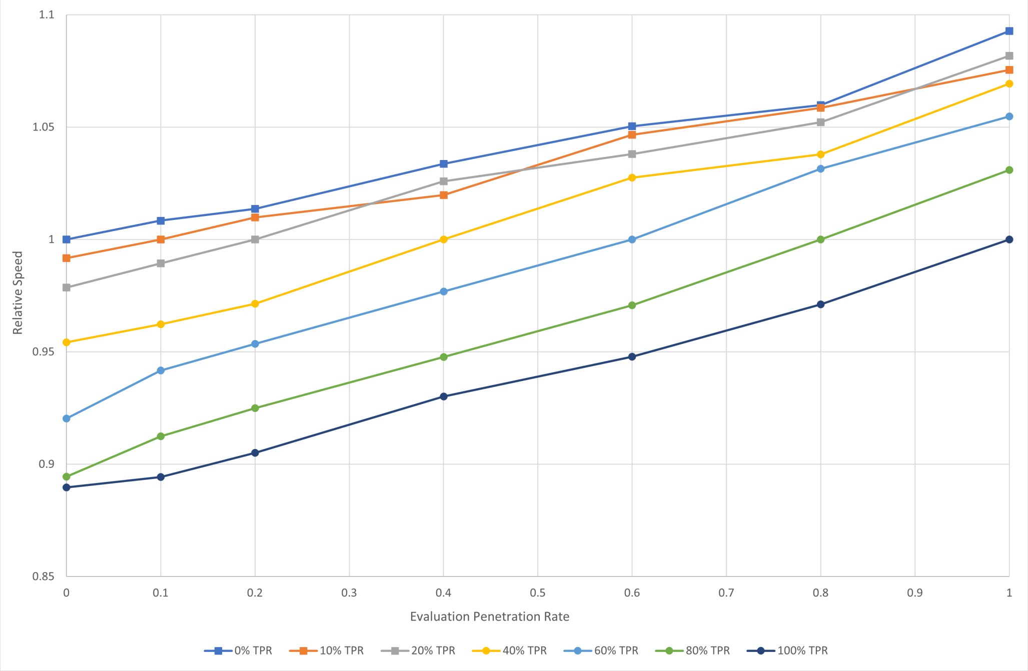

Figure 57: Relative performance on the metric of average vehicle speeds in experiments by EPR at 55% traffic density with unknown route information

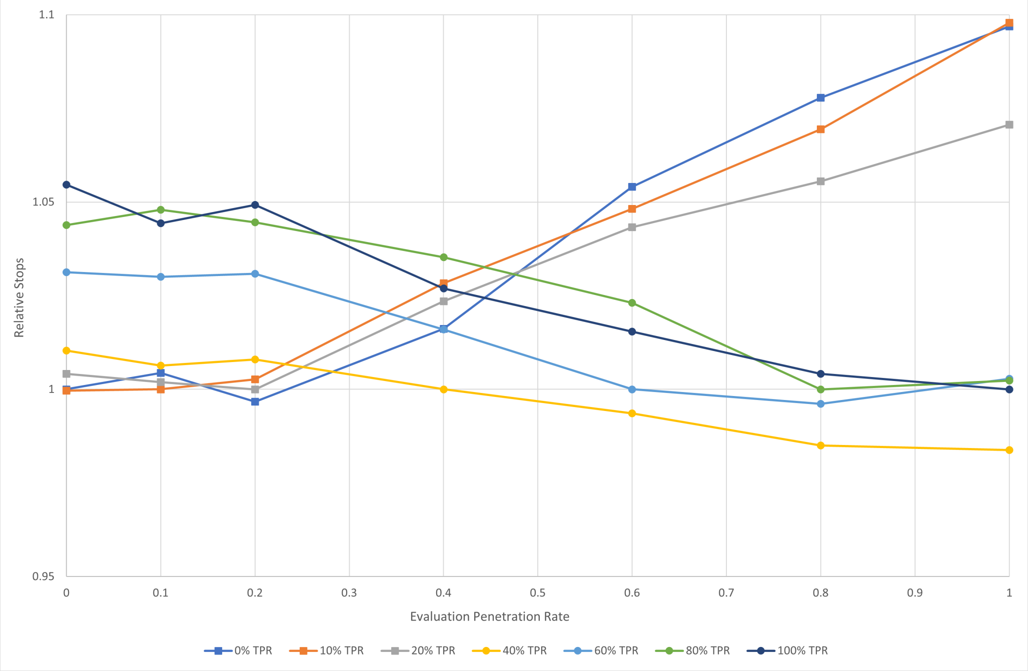

Figure 58: Relative performance on the metric of average journey stops in experiments by EPR at 55% traffic density with unknown route information

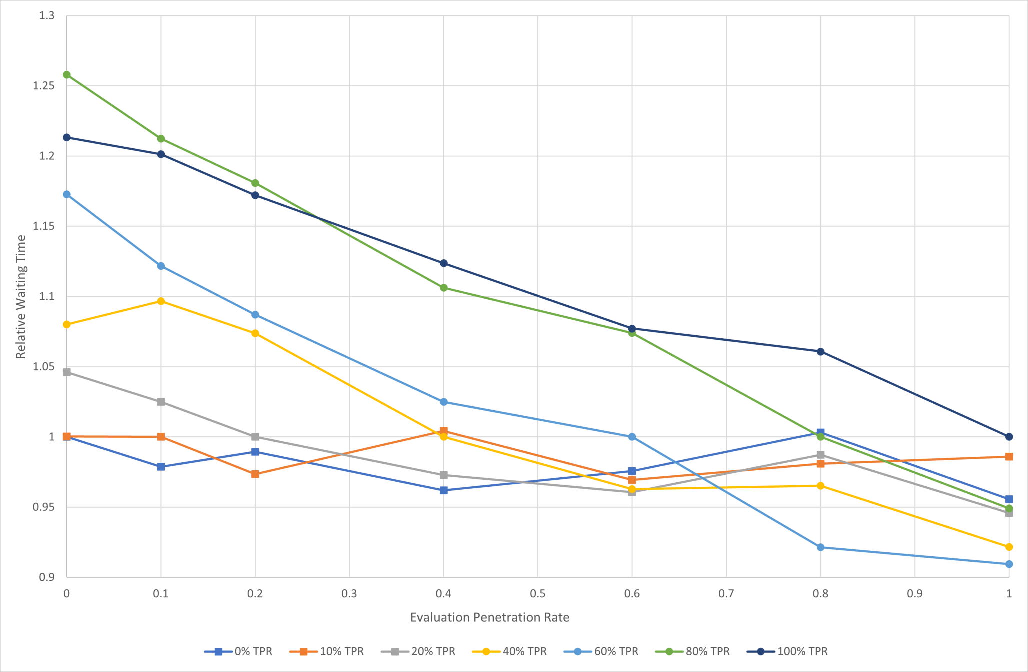

Figure 59: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 55% traffic density with unknown route information

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

7.14 Experiment 6 - 65% Scenario

In the 65% traffic density scenario, all agents performed better on the metrics of waiting times and average speeds if the EPR was increased. Again, this was especially true for agents trained at higher TPRs, which tended to make much larger gains as the EPRs increased. For example, when evaluated with a 100% EPR, the agents trained with 80% and 100% TPR were able to reduce waiting times by 24% and 27% respectively and increases in average speeds by 16% and 17% respectively, compared to when they were evaluated with a 0% EPR. This validates that the framework can in many cases manage increases in the EPR in this scenario.

However, the agents trained with 0% and 10% TPR could only achieve a reduction of 8% and 12% in waiting times and an increase of 11% and 13% in average speeds, and on the metric of average journey stops, only the agents trained with a TPR of 60% or higher improved as the EPR increased. Also, the performance of the agents trained at TPRs lower than 40% degraded as EPRs increased on this metric with increases in stops of up to 7% observed. Interestingly, increasing the EPR from 80% to 100% had by far the largest negative effect on stops of any increase in EPR on these agents. The performance of the agent trained with a TPR of 40%, on this metric, did also tend to degrade but never deviated by more than 2% regardless of the EPR.

This once again is in keeping with the understanding that has so far been built up where higher TPRs lead to policies that were more likely to gamble that approaching vehicles were GLOSA equipped and would therefore clear the junction without stopping or avoid becoming part of the queue for as long as possible. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues (see section 7.5).

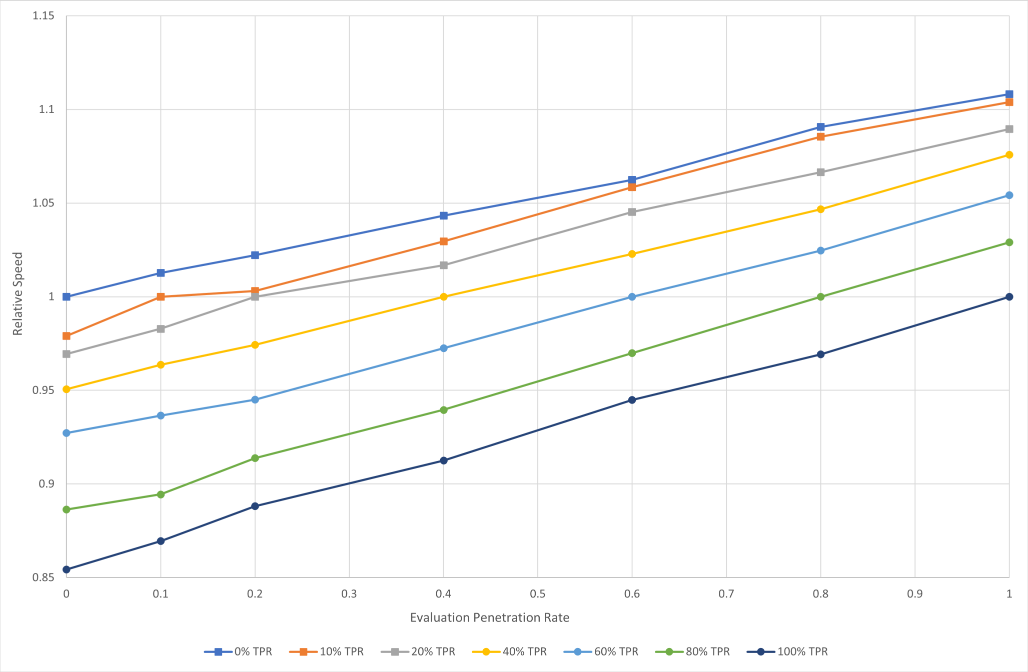

Figure 60: Relative performance on the metric of average vehicle speeds in experiments by EPR at 65% traffic density with unknown route information

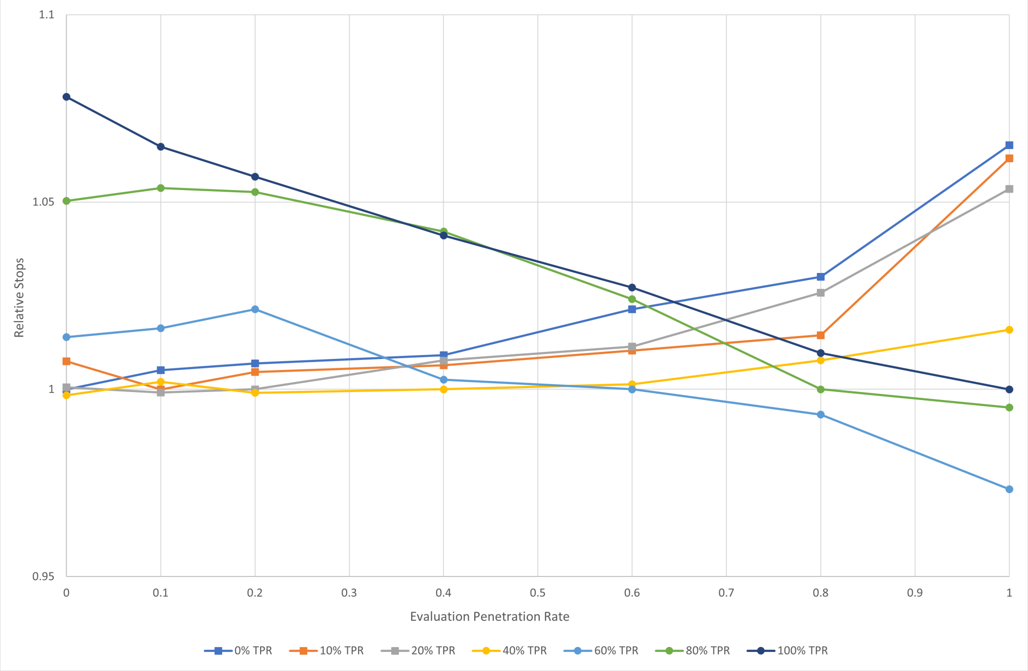

Figure 61: Relative performance on the metric of average journey stops in experiments by EPR at 65% traffic density with unknown route information

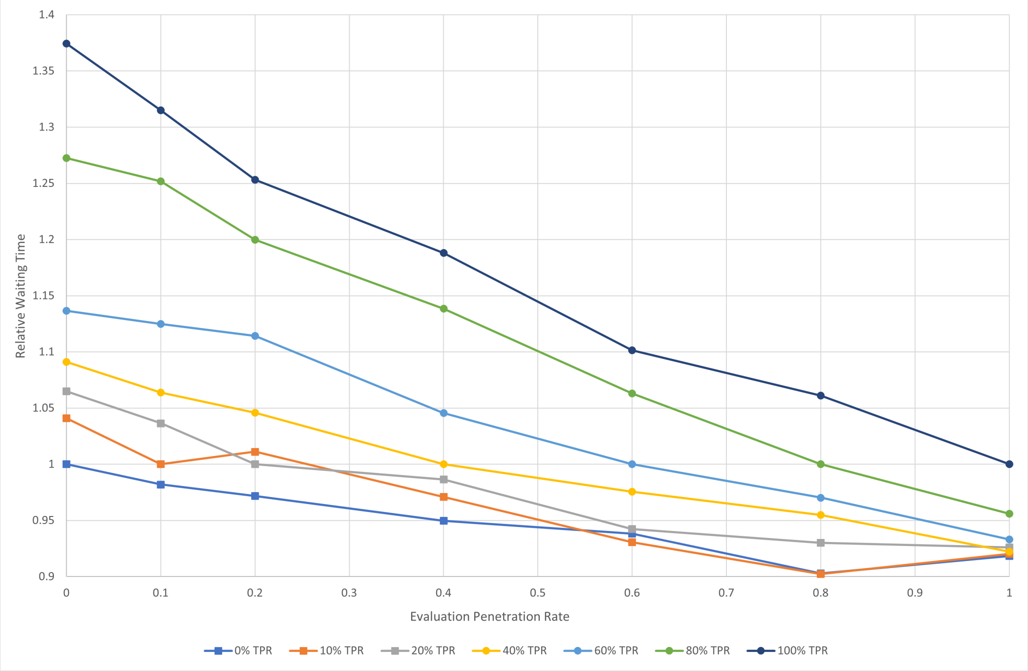

Figure 62: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 65% traffic density with unknown route information

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

7.15 Experiment 6 - 75% Scenario

In the 75% traffic density scenario, all agents performed better on the metrics of waiting times and average speeds if the EPR was increased. Again, this was especially true for agents trained at higher TPRs, which tended to make much larger gains as the EPRs increased. For example, when evaluated with a 100% EPR, the agents trained with 80% and 100% TPR were able to reduce waiting times by 27% and 32% and increase average speeds by 19% and 21% compared to when they were evaluated with a 0% EPR. This validates that the framework can in many cases manage increases in the EPR in this scenario. However, the agents trained with 0% and 10% TPR could only achieve a reduction of 10% and 8% in waiting times and an increase of 13% and 12% in average speeds.

On the metric of average journey stops, only the agents trained with a TPR of 40% or higher improved as the EPR increased. The performance of the agents trained at TPRs lower than 40% degraded as EPRs increased on this metric with increases in stops of up to 9% observed. The performance of the agent trained with a TPR of 40%, on this metric, did tend to improve as the EPR increased, but it only achieved a 2.5% reduction in stops at best.

This once again is in keeping with the understanding that has so far been built up where higher TPRs lead to policies that were more likely to gamble that approaching vehicles were GLOSA equipped and would therefore clear the junction without stopping or avoid becoming part of the queue for as long as possible. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues (see section 7.5).

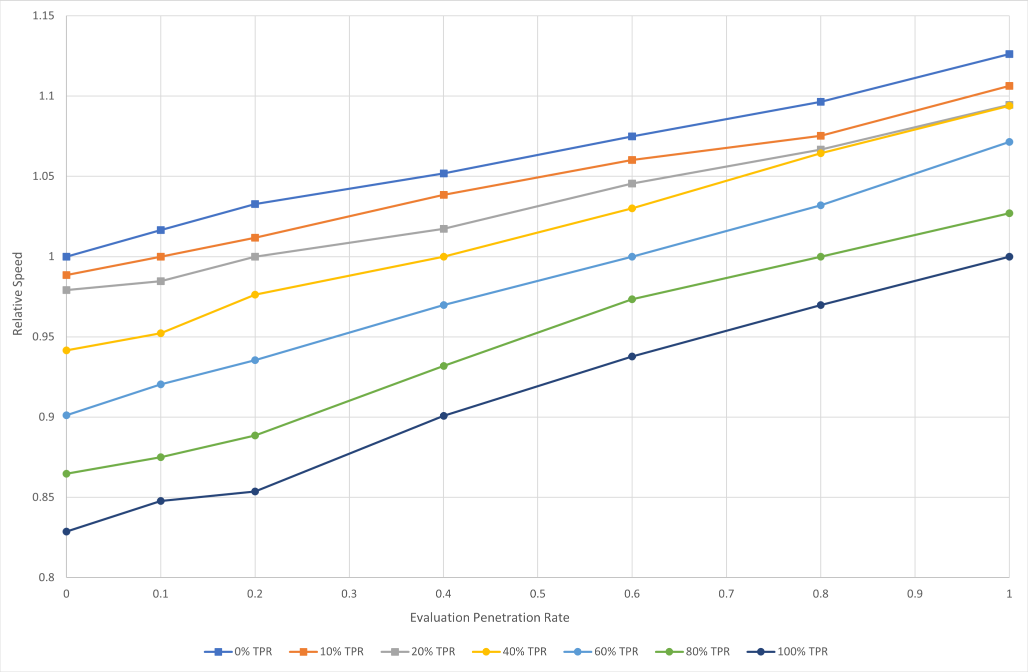

Figure 63: Relative performance on the metric of average vehicle speeds in experiments by EPR at 75% traffic density with unknown route information

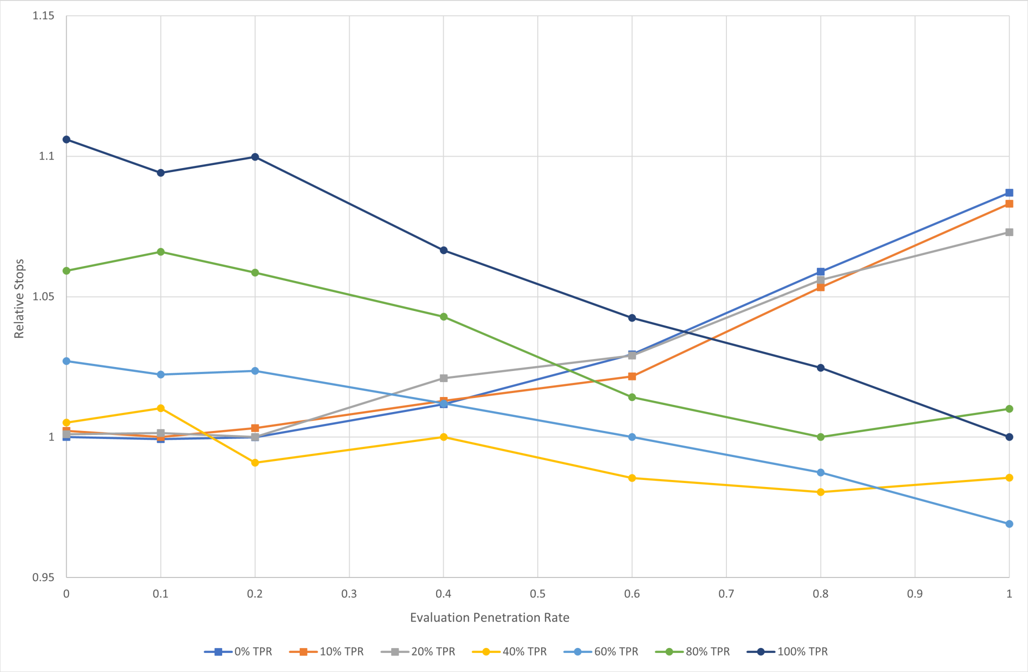

Figure 64: Relative performance on the metric of average journey stops in experiments by EPR at 75% traffic density with unknown route information

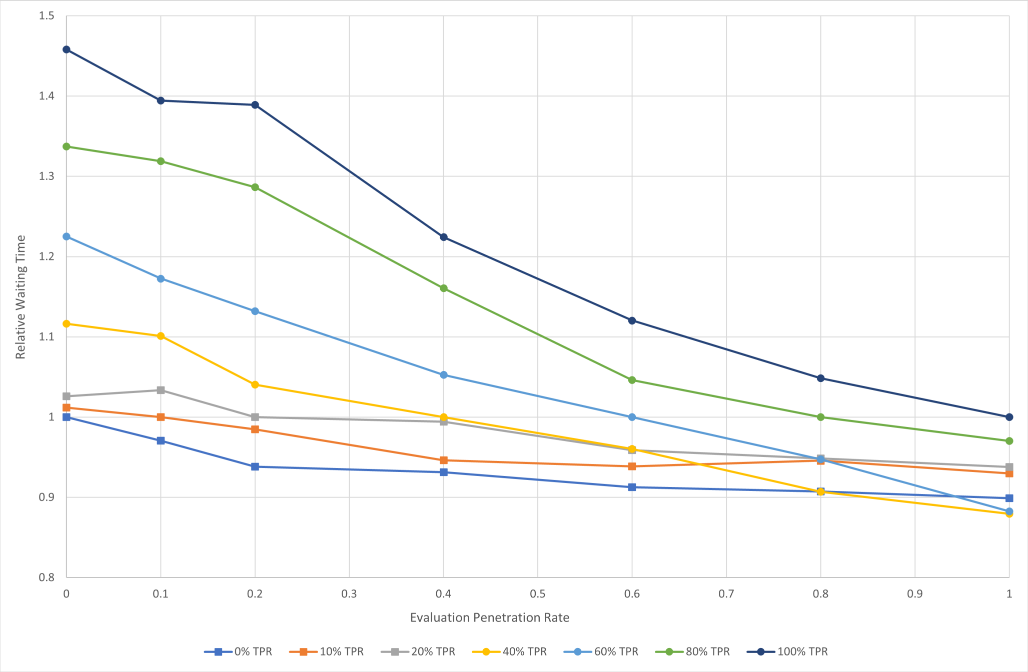

Figure 65: Relative performance on the metric of average vehicle waiting time in experiments by EPR at 75% traffic density with unknown route information

A two-way ANOVA test was performed on these results for each metric and it was found that the p values were all $\ll 1\%$ and therefore: the changes in TPR had a statistically significant effect on the performance observed on all metrics; the changes in EPR had a statistically significant effect on the performance observed on all metrics; and the effect of each factor (TPR and EPR) on performance was significantly dependent on the level of the other factor. The test statistics can be found in Table 7.

7.16 Experiment 6 - Discussion

Overall, while all agents improved on the metrics of waiting time and average speed as the EPR increased, agents trained with a TPR of 20% or less tended to perform better on the metric of stops at lower EPRs, while agents trained with a TPR of 60% or more tended to perform better on the metric of stops at higher EPRs. Also, in the 55% traffic scenario, the agents trained at TPRs lower than 20% showed only minimal improvement on the metric of waiting time.

The results for all scenarios, once again, suggest that divergent values for the TPR and EPR can lead to poorer performance for the joint control framework. As before, this is caused by the unexpected change in vehicle behaviour, causing the agents' learned predictions of action value to be incorrect and leading to suboptimal decision-making. Therefore, the benefits of having more equipped vehicles are outweighed by the drawbacks of the RLTC system not expecting the change in driving behaviour.

Overall, while all agents improved on the metrics of waiting time and average speed as the EPR increased, agents trained with a TPR of 20% or less tended to perform better on the metric of stops at lower EPRs, while agents trained with a TPR of 60% or more tended to perform better on the metric of stops at higher EPRs. At this point, a recommendation can be made that RLTC-equipped joint control frameworks should be retrained when the penetration rate at the site of deployment changes significantly. Another recommended alternative strategy for future research would be to change the state space to include information about how many or which vehicles are GLOSA equipped, so that the RLTC system can adapt to changes in penetration rates and scenarios where the RLTC model is inaccurate can be avoided. It is possible that this could be achieved either by giving the agents the live penetration rate, as one or more scalar values for approaching roads/areas, or by including information encoded in a DTSE style, where specific vehicles can be marked as either equipped or unequipped. However, further research would be required to identify if these measures would be successful.

8 Conclusion

The aim of this thesis is to understand the potential for using reinforcement learning to combine RTCs and GLOSA in CNAV dominated traffic scenarios. To achieve that aim, several objectives were set out.

The first objective of this thesis is to understand the existing approaching and their limitations by fully reviewing the state of the art in GLOSA, RTCs, and existing combinations of the two. To fulfil this objective, a literature review was conducted. It was identified that three strategies had been found that allowed for the creation of joint control frameworks. These strategies have centred on three approaches: predicting future signal plans to activate GLOSA, using dynamic programming or similar methods to schedule the arrival times of vehicles approaching the junction, and combining RLTC with either GLOSA or reinforcement learning controlled vehicles.

The two former approaches were found to be unsuitable for real world deployment, while the latter type of solution had been shown to be effective for CAVs and held the potential to also be effective for connected non-autonomous vehicles. However, it was found that no research existed that constructed a RLTC-based joint control framework for connected non-autonomous vehicles. Also, none of the existing research into RLTC-based joint control frameworks examined the effects of differences between the penetration rates of evaluation and training or of vehicle route information being unavailable. This satisfied the first objective and led to the defining of three gaps in knowledge.

Firstly, it was unknown how a joint control framework that uses reinforcement learning traffic control and is designed for non-autonomous vehicles will perform. Secondly, it is unknown how a joint control framework that uses reinforcement learning traffic control will perform if the penetration rates of GLOSA at the site of deployment are different from those in the training scenario. Thirdly, it is unknown how a joint control framework that uses reinforcement learning traffic control will perform if route information is unavailable.

After this the second objective was completed by designing two framework that incorporates RTCs and GLOSA for CNAV dominated traffic scenarios, one for Isolated junctions and one for Arterials.

The third objective was to understand the potential performance of the framework, by performing a series of experiments. Upon conducting these experiments it was found that there was potential for frameworks to outperformed the benchmark systems in terms of waiting time, queue length, number of stops and average speed for traffic densities up to at least 75%, with

drivers able to slow down smoother when approaching red lights, eliminating stops. Also, thanks to advanced warning of green phases from the GLOSA system, equipped vehicles were able to speed up earlier and clear the junction faster. Furthermore, these results were achieved with the simulations running faster than real time on a single desktop computer with a consumer CPU and GPU, meaning that computation is not a barrier to the real-world deployment of this framework. However, there are other barriers including the requirements in terms of training time and power which are presently vast, with several months of work being required to produce each set of agents.
The results also implied that the joint control framework was successful at including and utilising GLOSA, as it had had a positive effect on overall traffic flow despite its use in combination with an adaptive traffic control system. However, often a critical mass of GLOSA equipped vehicles had to be reached before positive results were achieved, with results becoming negative at low penetration rates. Overall, these results quantified the performance of a RLTC-based joint control frameworks for CNAVs and addressed the first gap in knowledge.

The fourth objective was to understand the impact of operation in imperfect conditions, e.g. differences in TPR and EPR, or route information being unavailable, by performing further experiments. Upon conducting these experiments, the results showed that differences in TPRs and EPRs sometimes resulted in deficient performance.

Often low TPR agents would perform worse or poorly in all metric when the EPR was high because they made decisions with the assumption that vehicles would not be equipped with GLOSA and therefore often choose to prioritise existing queues over approaching vehicles, as it could not trust that approaching vehicles were GLOSA equipped and would clear the junction without stopping but it could guarantee that the queue would be cleared. This indicates that low penetration rates of GLOSA are another barrier to practical deployment.

However, higher TPR agents saw no negative effects of the introduction of more GLOSA equipped vehicles as they were more willing to gamble that a vehicle was GLOSA equipped and that it would clear the junction without stopping. Accordingly, the agents tended to make decisions that would actively benefit GLOSA equipped vehicles if they were there, instead of always defaulting to clearing existing queues.

Also, while there were clear negative effects to route information being withheld from the framework, as they had to rely more on predictions to know how many vehicles would arrive at the junction soon, results were still overall positive in these scenarios. Overall, these results quantified the impact of operation in imperfect conditions, e.g. differences in TPR and EPR, or route information being unavailable, and addressed the second and third gaps in knowledge.

As a whole, these results are positive for GLOSA and take frameworks like this one a step closer to being implementable. However, further research in this area would be required as some challenges, and many questions remain, which must be addressed before that can happen. Also, the value of such research is dependent on the development and adoption timelines for CAVs and GLOSA capable vehicles.

If CAVs are expected to arrive in the near future, the continuation of this research seems unjustifiable as the initial startup costs, both in terms of future research required to address the frameworks limitations and barriers to deployment such as the adoption of GLOSA capable vehicles, would likely outweigh the benefits gained from this technology before the widespread adoption of CAVs. However, if CAVs, although long promised, are in fact not expected in the foreseeable future, there is clear benefits to further research in the development of joint control frameworks for CNAVs, like the ones presented here, and GLOSA more generally.

8.1 Limitations and Future Work

The last objective was to understand the limitations of this research and the direction future research should take. Over the course of conducting this research has a few simplifying assumptions have been made. Firstly, as real-world tests were not possible, this work has relied on simulations and this has introduced some error as the car following models, GLOSA equipped driver behaviour models, and traffic scenarios do not perfectly reflect the real world.

This however is presently unachievable, as this work contained other simplifications that must be addressed. In particular, the lack of pedestrians, public transport, HGVs, and other types of vehicles. This work excluded them as they would have introduced more variables to elements like the car following models, and complexity to the framework, which could have caused distortion of the results or a failure of the framework. However, with the lessons learned in this thesis, concerning how best to implement a joint control framework for CNAVs, it should now be possible for other researcher to explore the effects of including pedestrians, public transport, HGVs, and other types of vehicles.

One important open question is how pedestrian or public transport priority could be implemented. It is possible that these features could be achieved with modifications to the state and reward functions, to allow the agents to identify and manage such road users. However, this requires further research.

Also, the literature review identified that HEVs benefit differently from GLOSA implementations than petrol or diesel vehicles due to differences in their energy/fuel consumption models. As HEVs are increasing taking up greater and greater market shares, joint control frameworks designed for today's traffic flows may need to be retrained or redesigned to account for further changes in traffic makeups.

It would also make sense for future work to create agents who are trained for a broad range of traffic scenarios, as this research only trains agents in a single traffic density scenario each. This change should not have a profound effect on performance in any given scenario. But such work would be required before a real-world test or implementation.

A final limitation is the use of only lower traffic densities. However, the work in this thesis has contributed to making it possible to apply similar frameworks to these higher density scenarios.

Aside from these limitations, there also exists several situations and solutions that have not been tried. This work has not been extended to connected networks, which would be a crucial step to the implementation of this system in many urban environments.

Another recommended alternative strategy for future research would be to try changing the state space to include information about how many or which vehicles are GLOSA equipped, so that the RLTC system can adapt to changes in penetration rates and scenarios where the RLTC model is inaccurate can be avoided. It is possible that this could be achieved either by giving the agents the live penetration rate, as one or more scalar values for approaching roads/areas, or by including information encoded in a DTSE style, where specific vehicles can be marked as either equipped or unequipped. Such an upgrade could solve the issue of deficient performance for divergent TPRs and EPRs.

On a similar front, many RLTC systems are beginning to use raw pixel snapshots (live camera feeds) as their state spaces. This is a solution which offers a wealth of data to those RLTC systems, at a fraction of the cost of other detector systems. However, while this would appear beneficial for standard RLTC systems, questions remain over the use of raw pixel snapshots data in joint control frameworks as it is possible that a large number of cameras would be needed to properly cover the approach roads to sufficient distances for GLOSA. If many cameras were needed, the complexity of training could be greatly increased.

Finally, the largest performance gains to be made would appear to be related to increasing the length of future signal plans, to activate GLOSA for more vehicles. While further increasing phase length is not recommended, as this would simply reduce the effectiveness with which the RLTC systems can deal with phase to phase fluctuations in traffic control, it is possible that a framework could be constructed with a continuous action space that controls the signal phasing and timing data it sends out, as well as the junction. Although such a framework would require far more training than the frameworks described in this thesis, it would allow for a significant increase in future signal plans with performance gains likely, but this could also introduce incorrect future signal plans.

Bibliography

[1] Selinger, M. Schmidt, L. (2010). Adaptive Traffic Control Systems in the United States: Updated Summary and Comparison. Available at: http://cdnassets.hw.net/da/ad/6d673e3b4202b71314ab5eff3446/3675-adaptive-traffic-control-systems-in-the-united-states-updated-summary-and-compa pdf (Accessed 7 Apr. 2021).

[2] Aavani, P. Sawant, M, K. Sawant, S. Deshmukh, R, S. (2017). A Review on Adaptive Traffic Controls Systems. International Journal of Latest Engineering and Management Research. Volume 2. Issue 1. p 52-57. Available at: http://www.ijlemr.com/papers/volume2-issue1/9-IJLEMR-22024.pdf (Accessed 9 Aug. 2020).

[3] Temple Inc. (2015). SIEMENS-ACS-LITE-Adaptive-Master-Control-Software.pdf. Available at: https://www.temple-inc.com/images/pdfs/Controllers-ControllerSoftware/SIEMENS-ACS-LITE-Adaptive-Master-Control-Software.pdf. (Accessed 13 Oct. 2023).

[4] Elkins, S. Niehus, G. (2012). InSync Adaptive Traffic Control System for the Veterans Memorial Hwy Corridor on Long Island, NY. Available at: https://rosap.ntl.bts.gov/view/dot/24966 (Accessed 28 Mar. 2021).

[5] Clark, J. (2013). InSync Adaptive Traffic Control Shows Initial Safety Benefits. Available at: https://trafficbot.rhythmtraffic.com/wp-content/uploads/2018/10/Safety_Benefits_of_InSync.pdf (Accessed 3 May. 2021).

[6] F. Abbracciavento, F. Zinnari, S. Formentin, A. G. Bianchessi, S. M. Savaresi. (2023). Multi-intersection traffic signal control: A decentralized MPC-based approach. doi: https://doi.org/10.1016/j.ifacsc.2022.100214.

[7] M. B. Younes, A. Boukerche, F. D. Rango. (2023). SmartLight: A smart efficient traffic light scheduling algorithm for green road intersections. doi: https://doi.org/10.1016/j.adhoc.2022.103061.

[8] D. Sauerburger. (2014). Actuation. Available at: https://sauerburger.org/comactuation.htm (Accessed 6 Aug. 2020).

[9] Pariota, L. Costanzo, L, D. Coppola, D'Aniello, A, C. Bifulco, G, N. (2019). Green Light Optimal Speed Advisory: a C-ITS to improve mobility and pollution. doi: https://doi.org/10.1109/EEEIC.2019.8783573.

[10] Zhao, X. Jia, Z. Wei, N. Guo, D. (2023). Green light optimized speed advisory achieves fuel savings and CO2 emission reduction by profoundly impacting driving behavior. doi: https://doi.org/10.1016/j.jclepro.2023.138634.

[11] Seredynski, M. Ruiz, P. Szczypiorski, K. Khadraoui, D. (2014). Improving Bus Ride Comfort Using GLOSA-Based Dynamic Speed Optimisation. doi: https://doi.org/10.1109/IPDPSW.2014.58.

[12] Jia, Z. Wei, N. Yin, J. Zhao, X. (2023). Energy saving and emission reduction effects from the application of green light optimized speed advisory on plug-in hybrid vehicle. doi: https://doi.org/10.1016/j.jclepro.2023.137452.

[13] Hong, J. Luo, X. Wu, H. Na, X. Chu, H. Gao, B. Chen, H. (2024). Energy-Saving Driving Assistance System Integrated With Predictive Cruise Control for Electric Vehicles. doi: https://doi.org/10.1109/tiv.2024.3358797.

[14] Stevanovic, A. Stevanovic, J. Kergaye, C. (2013). Green Light Optimized Speed Advisory Systems: Impact of Signal Phasing Information Accuracy. doi: https://doi.org/10.3141/2390-06.

[15] Bodenheimer, R. Brauer, A. Eckhoff, D. German, R. (2014). Enabling GLOSA for adaptive traffic lights. doi: https://doi.org/10.1109/VNC.2014.7013336.

[16] Genser, A. Makridis, M. Yang, K. Ambühl, L. Menendez, M. Kouvelas, A. (2022). Time-to-Green predictions for fully-actuated signal control systems with supervised learning. doi: https://doi.org/10.48550/arXiv.2208.11344.

[17] Chai, L. Cai, B. Guan, W, S. Wang, J. (2017). Connected and autonomous vehicles coordinating method at intersection utilizing preassigned slots. doi: https://doi.org/10.1109/ITSC.2017.8317934.

[18] Fayazi, S, A. Vahidi, A. Luckow, A. (2017). Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP. doi: https://doi.org/10.23919/ACC.2017.7963717.

[19] Ding, J. Xu, H. Hu, J. Zhang, Y. (2017). Centralized cooperative intersection control under automated vehicle environment. doi: https://doi.org/10.1109/IVS.2017.7995841.

[20] Zhang, Y. Malikopoulos, A, A. Cassandras, C, G. (2017). Decentralized optimal control for connected automated vehicles at intersections including left and right turns. doi: https://doi.org/10.1109/CDC.2017.8264312.

[21] Mirheli, A. Tajalli, M. Hajibabai, L. Hajbabaie, A. (2019). A consensus-based distributed trajectory control in a signal-free intersection. doi: https://doi.org/10.1016/j.trc.2019.01.004.

[22] Priest, J. Ghadbeigi, H. Ayvar-Soberanis, S. Liljerehn, A. Way, M. (2022). A modified Johnson-Cook constitutive model for improved thermal softening prediction of machining simulations in C45 steel. doi: https://doi.org/10.1016/j.procir.2022.03.022.

[23] MotherFrunker. (2022). Tesla FSD Timeline: Elon Musk Quotes on FSD. Available at: https://motherfrunker.ca/fsd/.

[24] Li, D. Zhu, F. Wu, J. Wong, Y, D. Chen, T. (2024). Managing mixed traffic at signalized intersections: An adaptive signal control and CAV coordination system based on deep reinforcement learning. doi: https://doi.org/10.1016/j.eswa.2023.121959.

[25] Guo, J. Cheng, L. Wang, S. (2023). CoTV: Cooperative Control for Traffic Light Signals and Connected Autonomous Vehicles Using Deep Reinforcement Learning. doi: https://doi.org/10.1109/TITS.2023.3276416.

[26] Brake, A, G. (2016). MIT researchers plan ”death of the traffic light” with smart intersections. de zeen. Available at: https://www.dezeen.com/2016/03/21/light-traffic-junctions-mit-research-smart-intersections-design-driverless-vehicles/

[27] Yang, J. Zhang, J. Wang, H. (2020). Urban Traffic Control in Software Defined Internet of Things via a Multi-Agent Deep Reinforcement Learning Approach. doi: https://doi.org/10.1109/TITS.2020.3023788.

[28] Peng, X. Gao, H. Han, G. Wang, H. Zhang, M. (2023). Joint Optimization of Traffic Signal Control and Vehicle Routing in Signalized Road Networks using Multi-Agent Deep Reinforcement Learning. doi: https://doi.org/10.48550/arXiv.2310.10856.

[29] Maaadi, S. Stein, S. Hong, J. Murray-Smith, R. (2022). Real-Time Adaptive Traffic Signal Control in a Connected and Automated Vehicle Environment: Optimisation of Signal Planning with Reinforcement Learning under Vehicle Speed Guidance. doi: https://doi.org/10.3390/s22197501.

[30] Zhang, H. Gu, J. Zhang, Z. Du, L. Backdoor attacks against deep reinforcement learning based traffic signal control systems. doi: https://doi.org/10.1007/s12083-022-01434-0.

[31] Seredynski, M. Dorronsoro, B. Khadraoui, D. (2013). Comparison of Green Light Optimal Speed Advisory Approaches. doi: https://doi.org/10.1109/ITSC.2013.6728552.

[32] Sharara, M. Ibrahim, M. Chalhoub, G. (2019). Impact of Network Performance on GLOSA. 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). doi: https://doi.org/10.1109/CCNC.2019.8651787.

[33] Katsaros, K. Kernchen, R. Dianati, M. Rieck, D. (2011). Performance study of a Green Light Optimized Speed Advisory (GLOSA) application using an integrated cooperative ITS simulation platform. doi: https://doi.org/10.1109/IWCMC.2011.5982524.

[34] Widodo, S. Hasegawa, T. Tsugawa, S. (2000). Vehicle fuel consumption and emission estimation in environment-adaptive driving with or without inter-vehicle communications. doi: https://doi.org/10.1109/IVS.2000.898373.

[35] Stebbins, S. Hickman, M. Kim, J. Vu, H, L. (2017). Characterising green light optimal speed advisory trajectories for platoon-based optimisation. doi: https://doi.org/10.1016/j.trc.2017.06.014.

[36] Li, J. Dribi, M. El-Moudni, A. (2014). Multi-vehicles green light optimal speed advisory based on the augmented lagrangian genetic algorithm. doi: https://doi.org/10.1109/ITSC.2014.6958080.

[37] Seredynski, M. Mazurczyk, W. Khadraoui, D. (2013). Multi-segment Green Light Optimal Speed Advisory. doi: https://doi.org/10.1109/IPDPSW.2013.157.

[38] Do, W. Saunier, N. (2022). Safety Benefits of Automated Speed Advisory Systems at Signalized Intersections. doi: https://doi.org/10.1177/03611981221115725.

[39] Stevanovic, A. Stevanovic, J. Kergaye, C. (2014). Comparative Evaluation of Benefits from Traffic Signal Retiming and Green Light Optimized Speed Advisory Systems. January 2014 Conference: 93rd TRB Annual Meeting, Transportation Research Board.

[40] Radivojevic, D. Stevanovic, J. Stevanovic, A. (2016). Impact of Green Light Optimized Speed Advisory on Unsignalized Side-Street Traffic. doi: https://doi.org/10.3141/2557-03.

[41] Zhang, Z. Zou, Y. Zhang, X. Zhang, T. (2020). Green Light Optimal Speed Advisory System Designed for Electric Vehicles Considering Queuing Effect and Driver's Speed Tracking Error. doi: https://doi.org/10.1109/ACCESS.2020.3037105.

[42] Masera, C, B, M. Developing a new optimal speed advisory algorithm for connected vehicles in signalised road networks. Available at: https://repository.lboro.ac.uk/articles/thesis/Developing_a_new_optimal_speed_advisory_algorithm_for_connected_vehicles_in_signalised_road_networks/19826146 (Accessed 07 Mar. 2024).

[43] Cantas, M, R. Surnilla, G. Sommer, M. (2022). Green Light Optimized Speed Advisory (GLOSA) with Traffic Preview. doi: https://doi.org/10.4271/2022-01-0152.

[44] Suzuki, H. Marumo, Y. (2019). Green Light Optimum Speed Advisory (GLOSA) System with Signal Timing Variations - Traffic Simulator Study. doi: https://doi.org/10.1007/978-3-030-27928-8_89.

[45] Xie, F. Naumann, S. Czogalla, O. (2020). Speed Adviser for Pedestrians to Choose the Optimal Path at Signaled Intersections. doi: https://doi.org/10.1007/978-3-030-39109-6_11.

[46] Ke, Y. (2015). A Fuel-Saving Green Light Optimal Speed Advisory for Signalized Intersection Using V2I Communication. doi: https://doi.org/10.7939/R3GX4529B.

[47] Bhattacharyya, K. Laharotte, P, A. Burianne, A. El-Faouzi, N, E. (2022). Assessing Connected Vehicle's Response to Green Light Optimal Speed Advisory From Field Operational Test and Scaling Up. doi: https://doi.org/10.1109/TITS.2022.3187532.

[48] Suramardhana, T, A. Jeong H, Y. (2014). A driver-centric green light optimal speed advisory (DC-GLOSA) for improving road traffic congestion at urban intersections. doi: https://doi.org/10.1109/APWiMob.2014.6920310.

[49] Jeong, H, Y. Suramardhana, T, A. Hung, N, H. (2014). Design and Implementation of Green Light Optimal Speed Advisory Based on Reference Mobility Models (GLOSA-RMM) in Cyber-Physical Intersection Systems (CPIS). doi: https://doi.org/10.7840/kics.2014.39B.8.544.

[50] Bradaï, B. Garnault, A. Picron, V. Gougeon, P. (2015). A Green Light Optimal Speed Advisor for Reduced CO2 Emissions. doi: https://doi.org/10.1007/978-3-319-19818-7_15.

[51] Suzuki, H. Marumo, Y. (2019). A New Approach to Green Light Optimal Speed Advisory (GLOSA) Systems for High-Density Traffic Flow. doi: https://doi.org/10.1109/ITSC.2018.8569394.

[52] Suzuki, H. Marumo, Y. (2020). Safety Evaluation of Green Light Optimal Speed Advisory (GLOSA) System in Real-World Signalized Intersection. doi: https://doi.org/10.20965/jrm.2020.p0598.

[53] Suzuki, H. Marumo, Y. (2020). Evaluating Green Light Optimum Speed Advisory (GLOSA) System in Traffic Flow with Information Distance Variations. doi: https://doi.org/10.1007/978-3-030-25629-6_78.

[54] Guo, S. Zhang, T. Liu, Y. (2022). Driving Velocity Tracking Error Analysis of Different Broadcast Methods Under Green Light Optimal Speed Advisory System. doi: https://doi.org/10.1007/978-981-16-5429-9_62.

[55] Preuk, K. Dotzauer, M. Jipp, M. (2018). Should drivers be informed about the equipment of drivers with green light optimal speed advisory (GLOSA)?. doi: https://doi.org/10.1016/j.trf.2018.06.040.

[56] Eckhoff, D. Halmos, B. German R. (2013). Potentials and limitations of Green Light Optimal Speed Advisory systems. doi: https://doi.org/10.1109/VNC.2013.6737596.

[57] Xu, B. Zhang, F. Wang, J. Li, K. (2015). B&B Algorithm-Based Green Light Optimal Speed Advisory Applied to Contiguous Intersections. doi: https://doi.org/10.1061/9780784479292.033.

[58] Simchon, L. Rabinovici, R. (2020). Real-Time Implementation of Green Light Optimal Speed Advisory for Electric Vehicles. doi: https://doi.org/10.3390/vehicles2010003.

[59] Luo, Y. Li, S. Zhang, S. Qin, Z. Li, K. (2017). Green light optimal speed advisory for hybrid electric vehicles. doi: https://doi.org/10.1016/j.ymssp.2016.04.016.

[60] Chen, H. Rakha, H, A. (2022). Developing and Field Testing a Green Light Optimal Speed Advisory System for Buses. doi: https://doi.org/10.3390/en15041491.

[61] Zhao, Y. Yao, S. Shao, H. Abdelzaher, T. (2018). CoDrive: Cooperative Driving Scheme for Vehicles in Urban Signalized Intersections. doi: https://doi.org/10.1109/ICCPS.2018.00037.

[62] Klöppel-Gersdorf, M. Grimm, J. Strobl, S. Auerswald, R. (2019). Performance Evaluation of GLOSA-Algorithms Under Realistic Traffic Conditions Using C2I-Communication. doi: https://doi.org/10.1007/978-3-030-02305-8_6.

[63] Guerrieri, M. Mauro, R. (2021). A Concise Introduction to Traffic Engineering: Theoretical Fundamentals and Case Studies. doi: https://doi.org/10.1007/978-3-030-60723-4.

[64] Papageorgiou, M., Ben-Akiva, M., Bottom, J., Bovy, P. H. L., Hoogendoorn, S. P., Hounsell, N. B., Kotsialos, A., McDonald, M., 2006. ITS and Traffic Management. Handbooks in Operations Research and Management Science. Ch11. pp743-754.

[65] Bell, M. C., Bretherton, R. D., 1986. Ageing of Fixed-Time Traffic Signal Plans. Second International Conference on Road Traffic Control, 15-18 April 1986. London: IEE

[66] Little, J, D, C. Kelson, M, D. Gartner, N, H. (1981). MAXBAND: A Versatile Program for Setting Signals on Arteries and Triangular Networks. Available at: `https://dspace.mit.edu/bitstream/handle/1721.1/1979/SWP-1185-08951478.pdf?sequence=1`. (Accessed 13 Nov. 2023)

[67] TRL Software. (2021). TRANSYT. Available at: `https://trlsoftware.com/products/junction-signal-design/transyt/#how-transyt-works` (Accessed 12 Apr. 2021).

[68] Moore, P. (2010). LinSig, Version 3, User Guide & Reference. doi=10.1.1.732.9811.

[69] Al-Turki, M. Jamal, A. Al-Ahmadi, H, M. Al-Sughaiyer, M, A. Zahid, M. (2020). On the Potential Impacts of Smart Traffic Control for Delay, Fuel Energy Consumption, and Emissions: An NSGA-II-Based Optimization Case Study from Dhahran, Saudi Arabia. doi: `https://doi.org/10.3390/su12187394`.

[70] Yang, X. Cheng, Y. Chang, G. A multi-path progression model for synchronization of arterial traffic signals. doi: `https://doi.org/10.1016/j.trc.2015.02.010`.

[71] Gartner, N, H. Assman, S, F. Lasaga, F. Hou, D, L. (1991). A multi-band approach to arterial traffic signal optimization. Transportation Research Part B: Methodological, Volume 25, Issue 1. p 55-74. doi: `https://doi.org/10.1016/0191-2615(91)90013-9`.

[72] Little, J, D, C. Kelson, M, D. Gartner, N, H. (1981). MAXBAND: A Program for Setting Signals on Arteries and Triangular Networks. Available at: `https://onlinepubs.trb.org/Onlinepubs/trr/1981/795/795-007.pdf`. (Accessed 13 Nov. 2023)

[73] Cabezas, X. García, S. (2018). A Heuristic Algorithm for Traffic Light Synchronization Based on the MAXBAND Model. doi: `https://doi.org/10.48550/arXiv.1805.03982`.

[74] Stamatiadis, C. Gartner, N. (1996). MULTIBAND-96: A Program for Variable-Bandwidth Progression Optimization of Multiarterial Traffic Networks. doi: `https://doi.org/10.1177/0361198196155400102`.

[75] Zhou, Y. Jia, S. Mao, B. Ho, T, K. Wei, W. (2016). An Arterial Signal Coordination Optimization Model for Trams Based on Modified AM-BAND. Discrete Dynamics in Nature and Society. vol. 2016. Article ID 5028095. doi: `https://doi.org/10.1155/2016/5028095`.

[76] Zhang, C. Xie, Y. Gartner, N, H. Stamatiadis, C. Arsava, T. (2015). AM-Band: An Asymmetrical Multi-Band model for arterial traffic signal coordination. Transportation Research Part C: Emerging Technologies. Volume 58. September 2015. P. 515-531. doi: `https://doi.org/10.1016/j.trc.2015.04.014`.

[77] Florek, K. (2020). Arterial Traffic Signal Coordination for General and Public Transport Vehicles Using Dedicated Lanes. Journal of Transportation Engineering, Part A: Systems. Vol. 146. Issue. 7. doi: https://doi.org/10.1061/JTEPBS.0000374.

[78] Wong, S, C. (1996). Group-based optimisation of signal timings using the TRANSYT traffic model. Transportation Research Part B: Methodological. Vol: 30. Issue: 3. PP. 217-244. doi: https://doi.org/10.1016/0191-2615(95)00028-3.

[79] Robertson, D, I. (1986). Research on the TRANSYT and SCOOT Methods of Signal Coordination. doi: https://doi.org/10.1.1.417.7154.

[80] Odhiambo, E, O. Evaluation of Signal Optimization Software. Available at: https://www.diva-portal.org/smash/get/diva2:1351990/FULLTEXT01.pdf. (Accessed on 19 Nov. 2023).

[81] Stevanovic, A. Martin, P, T. (2005). Assessing the Ageing of Pre-Timed Traffic Signal Control Using Synchro and SimTraffic. https://www.researchgate.net/profile/Aleksandar-Stevanovic/publication/274137225_Assessing_the_Ageing_of_Pretimed_Traffic_Signal_Control_Using_Synchro_and_SimTraffic/links/597a902fa6fdcc61bb166d5d/Assessing-the-Ageing-of-Pretimed-Traffic-Signal-Control-Using-Synchro-and-SimTraffi
pdf.

[82] Xhevat, P. Arlinda, A. (2013). A sample of semi-actuated traffic control versus fixed-time traffic control. Journal of Mechanics Engineering anti Automation. Vol: 3. 334-337. https://d1wqtxts1xzle7.cloudfront.net/31714940/HD-Conceptual_Design_and_Aerodynamic_Study_of_Joined-Wing_Business_Jet_Aircraft-JMEA_Vol.3__No.5__2013-libre.pdf?1392461914=&response-content-disposition=inline%3B+filename%3DHD_Conceptual_Design_and_Aerodynamic_Stu.pdf&Expires=1700426620&Signature=Wdg2Q1oiv4~oPlO5RJOO4UAXvBTbdmWx4Cgea5OGaUrJ48D5nlD62rRLE~cjEOyXx-wB4ki0CLUpvofBqnDe
_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=77.

[83] Stevanovic, A. Martin, P, T. (2014). Split-Cycle Offset Optimization Technique and Coordinated Actuated Traffic Control Evaluated Through Microsimulation. Transportation Research Record Journal of the Transportation Research Board December 2008. doi: https://doi.org/10.3141/2080-06.

[84] Salem, O. Chen, J. Salman, B. (2015). Enhancing TSM&O Strategies through Life Cycle Benefit/Cost Analysis: Life Cycle Benefit/Cost Analysis & Life Cycle Assessment of Adaptive Traffic Control Systems and Ramp Metering Systems. Available at: https://rosap.ntl.bts.gov/view/dot/29007 (Accessed 3 May. 2021).

[85] Shafik, S. I. (2017). Field Evaluation of Insync Adaptive Traffic Signal Control System in Multiple Environments Using Multiple Approachesin Multiple Environments Using Multiple Approaches. https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=6674&context=etd. (Accessed 20 Oct. 2023).

[86] Ban, X. Kamga, C. Wang, X. Wojtowicz, J. Klepadlo, E. Sun, Z. Mouskos, K. (2014). ADAPTIVE TRAFFIC SIGNAL CONTROL SYSTEM (ACS-LITE) FOR WOLF ROAD, ALBANY, NEW YORK. Available at: https://www.dot.ny.gov/divisions/engineering/technical-services/trans-r-and-d-repository/C-10-13%20Final%20Report_10-7-14.pdf. (Accessed 28 Apr. 2021).

[87] Head, L. Mirchandani, P. Shelby, S. (1998). The RHODES prototype: a description and some results. Available at: https://www.researchgate.net/publication/260399900_The_RHODES_prototype_a_description_and_some_results (Accessed 24 Apr. 2021).

[88] Hamilton, A. (2015). Improving Traffic Movement in an Urban Environment. UNIVERSITY OF SOUTHAMPTON. FACULTY OF ENGINEERING AND THE ENVIRONMENT. PhD Thesis. Available at: https://eprints.soton.ac.uk/377283/. (Accessed 15 Mar. 2024).

[89] Wikipedia. (2022). Split Cycle Offset Optimisation Technique. Available at: https://en.wikipedia.org/wiki/Split_Cycle_Offset_Optimisation_Technique. (Accessed 12 Oct. 2023).

[90] Dubizzle Cars. (2021). All about SCOOT Traffic Control System. Available at: https://www.dubizzle.com/blog/cars/scoot-traffic-control-system/. (Accessed 12 Oct. 2023).

[91] Department for Transport. (1999). The "SCOOT" Urban Traffic Control System. Traffic Advisory Leaflet 7/99. Available at: https://webarchive.nationalarchives.gov.uk/20090511035005/http://www.dft.gov.uk/adobepdf/165240/244921/244924/TAL_7-991. (Accessed 16 Jun. 2020).

[92] Transport Research Laboratory. (2018). SCOOT, Adaptive Traffic Control System. Available at: https://trlsoftware.com/wp-content/uploads/2018/08/SCOOT.pdf. (Accessed 12 Oct. 2023).

[93] Transport Research Laboratory. (2014). Pedestrian SCOOT System. Available at: https://trl.co.uk/projects/pedestrian-scoot-system. (Accessed 12 Oct. 2023).

[94] Dubey, K. Gupta, T. (2020). Adaptive Traffic Control System: The Smart And Imperative Traffic Control System For India. 2020 International Conference on Intelligent Engineering and Management. doi: https://doi.org/10.1109/ICIEM48762.2020.9160175.

[95] Robertson, D, I. Bretherton, R, D. (1991). Optimizing networks of traffic signals in real time-the SCOOT method. doi: https://doi.org/10.1109/25.69966

[96] Stallard, C. Owen, L, E. (1998). Evaluating adaptive signal control using CORSIM. doi: https://doi.org/10.1109/WSC.1998.745972

[97] Hansen, B, G. Martin, P, T. Perrin, H, J. (2000). SCOOT Real-Time Adaptive Control in a CORSIM Simulation Environment. doi: https://doi.org/10.3141/1727-04.

[98] Chilukuri, B, R. Perrin, J. Martin, P, T. (2004). SCOOT and Incidents: Performance Evaluation in Simulated Environment. doi: https://doi.org/10.3141/1867-26.

[99] Khaleel, N. Uthayasooriyan, A. Hartley, J. Traffic Occupancy Prediction Using a Non-linear Autoregressive Exogenous Neural Network. doi: https://doi.org/10.52549/ijeei.v10i3.3833.

[100] Shelby, S, G. Bullock, D, M. Gettman, D. Ghaman, R, S. Sabra, Z, A. Soyke, N. Overview and Performance Evaluation of ACS Lite: Low-Cost Adaptive Signal Control System. Available at: https://www.researchgate.net/publication/228652186_Overview_and_Performance_Evaluation_of_ACS_Lite_Low-Cost_Adaptive_Signal_Control_System (Accessed 1 May. 2021).

[101] Mladenovic, M. Stevanovic, A. Kosonen, L. Glavic, D. (2015). Guidelines for Development of Functional Requirements and Evaluation of Adaptive Traffic Control Systems. Mobil.TUM 2015 - International Scientific Conference on Mobility and Transport – Technologies, Solutions and Perspectives for Intelligent Transport Systems. https://www.researchgate.net/publication/325115411_Guidelines_for_Development_of_Functional_Requirements_and_Evaluation_of_Adaptive_Traffic_Control_Systems (Accessed 20 May. 2021).

[102] Jin, J. Ma, X. (2012). A Decentralized Traffic Light Control System Based on Adaptive Learning. IFAC-PapersOnLine, Volume 50, Issue 1. p 5301-5306. doi: https://doi.org/10.1016/j.ifacol.2017.08.958.

[103] Luyanda, F. Gettman, D. Head, L. Shelby, S. Bullock, D. Mirchandani, P. (2003). ACS-Lite Algorithmic Architecture: Applying Adaptive Control System Technology to Closed-Loop Traffic Signal Control Systems. Transportation Research Record, 1856(1), 175-184. doi: https://doi.org/10.3141/1856-19.

[104] Weichenmeier, F. Hildebrandt, R. Szarata, A. (2015). THE TRISTAR AND KRAK ÓW SYSTEMS A PTV BALANCE AND PTV EPICS CASE STUDY. The JCT Traffic Signal Symposium, Warwick, 18th September 2015. Available at: http://www.jctconsultancy.co.uk/Symposium/Symposium2015/

PapersForDownload/The%20Tristar%20&%20Krakow%20system%20-%20a%20PTV%20Balance%20and%20PTV%20Epics%20case%20study.pdf. (Accessed 19 Oct. 2023).

[105] Sha, R. (2017). Design and Performance Analysis of Urban Traffic Control Systems. Available at: https://core.ac.uk/download/pdf/154747275.pdf (Accessed 6 Apr. 2021).

[106] Issuu. (2021). TRAFFIC TRANSPORTATION MODELLING - Issuu. Available at: https://issuu.com/landorlinks/docs/d_myearbook_complete/s/12349257. (Accessed 12 Oct. 2023).

[107] Tafazoli, P. (2018). Integration of PTV Optima and Balance aiming at Intelligent Traffic Management System (ITMS). Available at: https://web.uniroma1.it/cdaingtrasporti/sites/default/files/Thesis_Tafazoli_MTRR_29oct18.pdf. (Accessed 19 Oct. 2023).

[108] Friedrich, B. (2002). Adaptive Signal Control: An Overview. Available at: http://www.iasi.cnr.it/ewgt/13conference/102_friedrich.pdf (Accessed 15 Apr. 2021).

[109] Tavladakis, K. Voulgaris, N. (1999). Development Of An Autonomous Adaptive Traffic Control System. doi: https://doi.org/10.1.1.126.8511.

[110] Mohr, W. (2019). APPLICATION OF REAL-TIME TRAFFIC ADAPTIVE SIGNAL CONTROL ON THE R44 ARTERIAL, STELLENBOSCH. Available at: https://scholar.sun.ac.za/server/api/core/bitstreams/0a9882b6-2eda-4e67-8d55-822f4be0f179/content. (Accessed 20 Oct. 2023).

[111] Traffic Infra Tech. (2016). Smart Signaling by PTV Balance and PTV Epics. Available at: https://trafficinfratech.com/smart-signaling-by-ptv-balance-and-ptv-epics/. (Accessed 20 Oct. 2023)

[112] Stevanovic, A. Zlatkovic, M. (2012). Evaluation of InSync Adaptive Traffic Signal Control in Microsimulation Environment. 92nd TRB Annual Meeting, Transportation Research Board, Washington D.C.. Available at: https://www.researchgate.net/publication/274137363_Evaluation_of_InSync_Adaptive_Traffic_Signal_Control_in_Microsimulation_Environment (Accessed 31 Aug 2020).

[113] Chandra, R, J. Bley, J, W. Penrod, S, S. (2011). ADAPTIVE CONTROL SYSTEMS AND METHODS. US Patent No. 8050854 B1. Available at: https://patents.google.com/patent/US8050854. (Accessed 20 Oct. 2023).

[114] Chandra, R, J. Bley, J, W. Penrod, S, S. (2012). External adaptive control systems and methods. US Patent No. US8103436B1. Available at: https://patents.google.com/patent/US8103436. (Accessed 20 Oct. 2023).

[115] Dakic, I. Stevanovic, A. Zlatkovic, M. Kergaye, C. (2016). Refinement of Performance Measures based on High-Resolution Signal and Detection Data. 19th EURO Working Group on Transportation Meeting, EWGT2016, 5-7 September 2016, Istanbul, Turkey. doi: https://doi.org/10.1016/j.trpro.2017.03.055.

[116] Stevanovic, A. Dakic, I. Zlatkovic, M. (2016). Comparison of adaptive traffic control benefits for recurring and non-recurring traffic conditions. 22nd ITS World Congress, Bordeaux 2015. doi: https://doi.org/10.1049/iet-its.2016.0032.

[117] Rhythm Engineering. InSync FAQs. Available at: https://rhythmtraffic.com/downloads/resources/InSync/Product_Information/InSync_FAQs.pdf.

[118] Stevanovic, A. Kergaye, C. Martin, P, T. (2009). SCOOT and SCATS: A Closer Look into Their Operations. Conference: 88th TRB Annual Meeting, Transportation Research Board. Available at: https://www.researchgate.net/publication/274137098_SCOOT_and_SCATS_A_Closer_Look_into_Their_Operations (Accessed 27 Apr. 2021).

[119] Samadi, S. Rad, A. P. Kazemi, F. M. Jafarian, H. (2012). Performance Evaluation of Intelligent Adaptive Traffic Control Systems: A Case Study. Journal of Transportation Technologies, Vol.2 No.3, July 2012. doi: https://doi.org/10.4236/jtts.2012.23027.

[120] Dutta, U. McAvoy, D. Lynch, J. Vandeputte, L. (2008). Evaluation of the Scats Control System. Available at: https://rosap.ntl.bts.gov/view/dot/34361. (Accessed 23 Oct. 2023).

[121] McCann, B. (2014). A REVIEW OF SCATS OPERATION AND DEPLOYMENT IN DUBLIN. Available at: https://api.semanticscholar.org/CorpusID:204841706.

[122] Sims, A, G. Dobinson, K, W. (1980). The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits. IEEE Transactions on Vehicular Technology, Volume: 29, Issue: 2, May 1980. doi: https://doi.org/10.1109/T-VT.1980.23833.

[123] Chong-White, C. Millar, G. Johnson, F. Shaw, S. (2011). The SCATS and the environment study: introduction and preliminary results. Australasian Transport Research Forum 2011 Proceedings 28 - 30 September 2011, Adelaide, Australia. Available at: https://australasiantransportresearchforum.org.au/wp-content/uploads/2022/03/2011_ChongWhite_Millar_Johnson_Shaw.pdf.

[124] Hunter, M, P. Wu, S, K. Kim, H, K. Suh, W. (2012). A Probe-Vehicle-Based Evaluation of Adaptive Traffic Signal Control. IEEE Transactions on Intelligent Transportation Systems, Volume: 13, Issue: 2, June 2012. doi: https://doi.org/10.1109/TITS.2011.2178404.

[125] Slavin, C. Feng, W. Figliozzi, M. Koonce, P. (2019). Statistical Study of the Impact of Adaptive Traffic Signal Control on Traffic and Transit Performance. Transportation Research Record: Journal of the Transportation Research Board, Volume 2356, Issue 1. doi: https://doi.org/10.1177/0361198113235600114.

[126] Gross, N, R. SCATS Adaptive Traffic System. TRB Adaptive Traffic Control Workshop January 2000. Available at: https://slideplayer.com/slide/6994957/.

[127] Stevanovic, A. (2010). NCHRP SYNTHESIS 403, Adaptive Traffic Control Systems: Domestic and Foreign State of Practice, A Synthesis of Highway Practice. Avialable at: http://elibrary.pcu.edu.ph:9000/digi/NA02/2010/14364.pd. (Accessed 10 Nov. 2023).

[128] Campbell, R. Skabardonis, A. (2014). Issues Affecting Performance of Adaptive Traffic Control Systems in Oversaturated Conditions. Transportation Research Record, 2438(1), 23-32. doi: https://doi.org/10.3141/2438-03.

[129] ITS International. (2012). Germany's approach to adaptive traffic control. https://www.itsinternational.com/its8/feature/germanys-approach-adaptive-traffic-control (Accessed 9 Aug. 2012).

[130] Brilon, W. Wietholt, T. (2013). Experiences with Adaptive Signal Control in Germany. Available at: https://journals.sagepub.com/doi/pdf/10.1177/0361198113235600102?casa_token=LXEPV-dyhTAAAAAA:w7j08EXfm3OasBgDvy6MpmNBH38L2FlqEsDYFAxvP0bpV0jRMEGPXr7_TPBvktFHD24uyOuUFZTCIA (Accessed 9 Aug. 2020).

[131] Hamilton, A. Waterson, B. Cherrett, T. Robinson, A. Snell, I. (2012). The evolution of urban traffic control: changing policy and technology. Universities' Transport Study Group UK Annual Conference 2012, Issue 1, Volume 36, 2013. doi: https://doi.org/10.1080/03081060.2012.745318.

[132] Busch, F. Kruse, G. (2001). MOTION for SITRAFFIC - a modern approach to urban traffic control. ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings. doi: https://doi.org/10.1109/ITSC.2001.948630.

[133] Montazeri, F. (2023). A study on traffic signal patterns and performance measures accounting for pedestrian and vehicle flows. Available at: https://espace.etsmtl.ca/id/eprint/3220/1/MONTAZERI_Farzaneh.pdf. (Accessed 10 Nov. 2023).

[134] Kupčuljaková, J. (2012). THE CONTROL SYSTEMS AT SIGNAL CONTROLLED JUNCTIONS FOR BUS PRIORITY. Available at: https://tac.uniza.sk/pdfs/das/2012/02/32.pdf (Accessed on 10 Nov. 23).

[135] Ghaman, R. Gettman, D. Head, L. Mirchandani, P, B. (2002). Adaptive control software for distributed systems. IEEE 2002 28th Annual Conference of the Industrial Electronics Society. doi: https://doi.org/10.1109/IECON.2002.1182892.

[136] Shao, C. (2009). Adaptive control strategy for isolated intersection and traffic network. Available at: https://www.proquest.com/docview/304830983?pq-origsite=gscholar&fromopenview=true.

[137] Liao, L, C. (1998). A Review of the Optimized Policies forAdaptive Control Strategy (OPAC). doi=10.1.1.496.6966.

[138] Gartner, N. (1983). OPAC: A Demand-responsive Strategy for Traffic Signal Control. Transportation Research Record Journal of the Transportation Research Board. No. 906. 75-81. Available at: https://www.researchgate.net/publication/243768686_OPAC_A_Demand-responsive_Strategy_for_Traffic_Signal_Control

[139] Gartner, N. Pooran, F. Optimized Policies for Adaptive Control Strategy in Real-Time Traffic Adaptive Control Systems: Implementation and Field Testing. Transportation Research Record Journal of the Transportation Research Board, Volume 1811, Issue 1. p 148-156. doi: https://doi.org/10.3141/1811-18.

[140] Shelby, S, G. (2004). Single-Intersection Evaluation of Real-Time Adaptive Traffic Signal Control Algorithms. Journal of the Transportation Research Board, Volume 1867, Issue 1. p 183-192. doi: https://doi.org/10.3141/1867-21.

[141] Mladenovic, M. Abbas, M. (2013). A Survey of Experiences with Adaptive Traffic Control Systems in North America. Available at: https://www.researchgate.net/publication/260020117_A_Survey_of_Experiences_with_Adaptive_Traffic_Control_Systems_in_North_America (Accessed 12 May. 2021).

[142] Mirchandani, P. Wang, F. (2005). RHODES to intelligent transportation systems. doi: https://doi.org/10.1109/MIS.2005.15.

[143] Mirchandani, P. Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. Transportation Research Part C: Emerging Technologies. Volume 9, Issue 6. December 2001. p. 415-432. doi: https://doi.org/10.1016/S0968-090X(00)00047-4.

[144] Mirchandani, P. Knyazyan, A. Head, L. Wu, W. (2001). An Approach Towards the Integration of Bus Priority, Traffic Adaptive Signal Control, and Bus Information/Scheduling Systems. Computer-Aided Scheduling of Public Transport. Lecture Notes in Economics and Mathematical Systems. vol 505. Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/978-3-642-56423-9_18.

[145] Head, K, L. Mirchandani, P, B. Sheppard, D. (2000). Hierarchical Framework for Real-Time Traffic Control. Available at: https://onlinepubs.trb.org/Onlinepubs/trr/1992/1360/1360-014.pdf.

[146] Sisso, T, J, C. (2016). Adaptive and Scalable Urban Traffic Control in an Indian Metropolis. https://essay.utwente.nl/70800/1/Sisso_MA_BMS.pdf (Accessed 31 May. 2021).

[147] George, G. (2016). Adaptive Signal Control Technology: State of Practice. International Journal for Scientific Research & Development. Vol 4. Issue 3. Available at: https://d1wqtxts1xzle7.cloudfront.net/79064485/IJSRDV4I30543-libre.pdf?1642581731=&response-content-disposition=inline%3B+filename%3DAdaptive_Signal_Control_Technology_State.pdf&Expires=1699825569&Signature=eBHqytXAUf-deqbIG9YQX~qAfgnA15Ka0zSCRXifMSxfZfYHr3h6dwdJKFGyQrzEPkhTDMoOb3OX0coGlfam
_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. (Accessed on 12 Nov. 2023).

[148] Dubey, K. (2019). Adaptive Traffic Control System: The Smart and Imperative Traffic Monitoring System for India. International Journal for Research in Applied Science & Engineering Technology. Vol 7. Issue 12. Available at: https://d1wqtxts1xzle7.cloudfront.net/62336492/35_IJRASET25969210-21820200311-43540-2k5gqa-libre.pdf?1584024029=&response-content-disposition=inline%3B+filename%3DAdaptive_Traffic_Control_System_The_Smar.pdf&Expires=1699825597&Signature=gGN7NAULatsLNTF0XMtulykq~XNw9jQwN3poeAONLAxebsE03jIV7tJNqLtP0NqLQlSO~HMsTcz5OLsUHYCs
_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. (Assessed on 12 Nov. 2023).

[149] CoSiCoSt-EnV. Composite Signal Control Strategy-Enhanced Version. Available at: http://intranse.in/sites/default/files/products/CoSiCoSt-EnVBrochure.pdf.

[150] Kouvelas, A. Aboudolas, K. Papageorgiou, M. Kosmatopoulos, E, B. (2011). A Hybrid Strategy for Real-Time Traffic Signal Control of Urban Road Networks. IEEE Transactions on Intelligent Transportation Systems. Vol: 12. Issue: 3. doi: https://doi.org/10.1109/TITS.2011.2116156.

[151] Bělinová, Z. Tichý, T. Přikryl, J. Cikhardtová, K. (2015). Smarter traffic control for middle-sized cities using adaptive algorithm. 2015 Smart Cities Symposium Prague (SCSP). doi: https://doi.org/10.1109/SCSP.2015.7181545.

[152] Kosmatopoulos, E. Papageorgiou, M. Bielefeldt, C. Dinopoulou, V. Morris, R. Mueck, J. Richards, A. Weichenmeier, F. (2006). International comparative field evaluation of atraffic-responsive signal control strategy in three cities. Transportation Research Part A: Policy and Practice Volume 40, Issue 5. p 399-413. doi.org/10.1016/j.tra.2005.08.004.

[153] Papageorgiou, M. Kouvelas, A. Kosmatopoulos, E. Dinopoulou, V. Smaragdis, E. Application of the Signal Control Strategy TUC in Three Traffic Networks: Comparative Evaluation Results. 2006 2nd International Conference on Information & Communication Technologies. doi: https://doi.org/10.1109/ICTTA.2006.1684460.

[154] Tenekeci, G. (2015). A modelling technique for assessing linked MOVA. Proceedings of the Institution of Civil Engineers - Transport. Vol: 160. Issue: 3. pp. 125-138. doi: https://doi.org/10.1680/tran.2007.160.3.125.

[155] Cabrejas-Egeaa, A. Zhang, R. Walton, N. Reinforcement Learning for Traffic Signal Control: Comparison with Commercial Systems. 14th Conference on Transport Engineering. doi: https://doi.org/10.48550/arXiv.2104.10455.

[156] Cai, C. Wong, C, K. Heydecker, B, G. (2009). Adaptive traffic signal control using approximate dynamic programming. Transportation Research Part C: Emerging Technologies, Volume 17, Issue 5. p 456-474. doi: https://doi.org/10.1016/j.trc.2009.04.005.

[157] Transport for Greater Manchester. (2018). Smart traffic light network, Integrating the Operations. Available at: https://tfgm.com/corporate/business-plan/case-studies/scoot-mova. (Accessed on: 15 Nov. 23)

[158] Transport Research Laboratory. (2021). MOVA. Available at: https://trlsoftware.com/products/traffic-control/mova/. (Accessed 15 Nov. 2023).

[159] Henry, J, J. (1989). PRODYN tests and future experiments on ZELT. Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS '89). doi: https://doi.org/10.1109/VNIS.1989.98779.

[160] Fox, K. Chen, H. Montgomery, F. Smith, M. Jones, S. (1998). Selected Vehicle Priority in the UTMC Environment (UTMC01). Available at: https://www.its.leeds.ac.uk/projects/spruce/utmc1rev.html#_Toc428247152. (Accessed on 16 Nov. 2023).

[161] Henry, J, J. Farges, J, L. (1989). PRODYN. Control, Computers, Communications in Transportation. Selected Papers from the IFAC/IFIP/IFORS Symposium, Paris, France. PP. 253-255. doi: https://doi.org/10.1016/B978-0-08-037025-5.50043-8.

[162] Henry, J, J. Farges, J, L. Tuffal, J. (1983). THE PRODYN REAL TIME TRAFFIC ALGORITHM. Control in Transportation Systems Proceedings of the 4th IFAC/IFIP/IFORS Conference, Baden-Baden, Federal Republic of Germany. PP. 305-310. doi: https://doi.org/10.1016/B978-0-08-029365-3.50048-1.

[163] Wahlstedt, J. (2013). Evaluation of the two self-optimising traffic signal systems Utopia/Spot and ImFlow, and comparison with existing signal control in Stockholm, Sweden.

16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). doi: https://doi.org/10.1109/ITSC.2013.6728449.

[164] Pavleski, D. Koltovska-Nechoska, D. Ivanjko, E. (2017). Evaluation of adaptive traffic control system UTOPIA using microscopic simulation. 2017 International Symposium ELMAR. doi: https://doi.org/10.23919/ELMAR.2017.8124425.

[165] Shepherd, S, P. (1992). A Review of Traffic Signal Control. Available at: https://eprints.whiterose.ac.uk/2217/1/ITS253_WP349_uploadable.pdf (Accessed 19 Apr. 2021).

[166] Shepherd, S. (2007). Traffic control in over-saturated conditions Foreign summaries. doi: https://doi.org/10.1080/01441649408716864.

[167] Yunex Traffic. (2023). Adaptive Traffic Control. Available at: https://www.yunextraffic.com/portfolio/smart-intersection/adaptive-traffic-control/.

[168] Trivedi, J, D. Devi, M, S. Dave, D, H. (2021). A Vision-Based Real-Time Adaptive Traffic Light Control System Using Vehicular Density Value And Statistical Block Matching Approach. doi:https://doi.org/10.2478/ttj-2021-0007.

[169] Sahri, N, H, M. (2005). The Effectiveness of The Implementation of ITACA System to Traffic Light Junction in Putrajaya. Available at: https://utpedia.utp.edu.my/id/eprint/8124/1/2005%20-%20The%20Effectiveness%20of%20The%20Implementation%20of%20ITACA%20System%20to%20Traffic%20Light%20Junction%20in%20Putra.pdf.

[170] Kraus, W. de Souza, F, A. Carlson, R, C. Papageorgiou, M. Dantas, L, D. Camponogara, E. Cost Effective Real-Time Traffic Signal Control Using the TUC Strategy. doi: 10.1109/MITS.2010.939916.

[171] Department for Transport. TRAFFIC ADVISORY LEAFLET: General Principles of Traffic Control by Light Signals Part 2 of 4. Available at: https://tsrgd.co.uk/pdf/tal/2006/tal-1-06_2.pdf.

[172] Land Transport Authority. (2019). Green Link Determining System. Available at: https://www.lta.gov.sg/content/ltagov/en/getting_around/driving_in_singapore/intelligent_transport_systems/green_link_determining_system.html.

[173] Bodenheimer, R. Eckhoff, D. German, R. (2015). GLOSA for adaptive traffic lights: Methods and evaluation. doi: https://doi.org/10.1109/RNDM.2015.7325247.

[174] Saldivar-Carranza, E. Li, H. Kim, W. Mathew, J. Bullock, D. Sturdevant, J. (2020). Effects of a Probability-Based Green Light Optimized Speed Advisory on Dilemma Zone Exposure. doi: https://doi.org/10.4271/2020-01-0116.

[175] Zhao, Y. Li, S. Hu, S. Su, L. (2017). GreenDrive: a smartphone-based intelligent speed adaptation system with real-time traffic signal prediction. doi: https://doi.org/10.1145/3055004.3055009.

[176] Ding, L. Zhao, D. Zhu, B. Wang, Z. Tan, C. Tong, J. Ma, H. (2023). SpeedAdv: Enabling Green Light Optimized Speed Advisory for Diverse Traffic Lights. doi: https://doi.org/10.1109/TMC.2023.3319697.

[177] Sumitomo Electric. (2019). Vehicle-infrastructure cooperation improves traffic safety and convenience. Available at: https://sumitomoelectric.com/products/universal-traffic-management-systems-utms.

[178] Li, J. Yu, C. Shen, Z. Su, Z. Ma, W. (2023). A survey on urban traffic control under mixed traffic environment with connected automated vehicles. doi: https://doi.org/10.1016/j.trc.2023.104258.

[179] Feng, Y. Yu, C. Liu, H, X. (2018). Spatiotemporal intersection control in a connected and automated vehicle environment. doi: https://doi.org/10.1016/j.trc.2018.02.001.

[180] Erdmann, J. (2013). Combining adaptive junction control with simultaneous Green-Light-Optimal-Speed-Advisory. doi: https://doi.org/10.1109/wivec.2013.6698230.

[181] Guo, Y. Ma, J. Xiong, C. Li, X. Zhou, F. Hao, W. (2019). Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach. doi: https://doi.org/10.1016/j.trc.2018.11.010.

[182] Yao, Z. Zhao, B. Yuan, T. Jiang, H. Jiang, Y. (2020). Reducing gasoline consumption in mixed connected automated vehicles environment: A joint optimization framework for traffic signals and vehicle trajectory. doi: https://doi.org/10.1016/j.jclepro.2020.121836.

[183] Pace, R, D. Fiori, C. Pariota, L. Storani, F. (2020). Centralised Traffic Control and Green Light Optimal Speed Advisory Procedure in Mixed Traffic Flow: An Integrated Modelling Framework. doi: https://doi.org/10.5772/intechopen.95247.

[184] Nguyen, V. Kim, O, T, T. Dang, T, N. Moon, S, I. Hong, C, S. (2016). An efficient and reliable Green Light Optimal Speed Advisory system for autonomous cars. doi: https://doi.org/10.1109/APNOMS.2016.7737260.

[185] Chen, H. Wi, F. Qiu, T, Z. (2024). Achieving Energy-Efficient and Travel Time-Optimized Trajectory and Signal Control for CAEVs. doi: https://doi.org/10.1109/TITS.2024.3355040.

[186] Shi, X. Zhang, J. Jiang, X. Chen, J. Hao, W. Wang, B. (2024). Learning eco-driving strategies from human driving trajectories. doi: https://doi.org/10.1016/j.physa.2023.129353.

[187] Chen, D. Zhang, K. Wang, Y. Yin, X. Li, Z. Filev, D. (2024). Communication-Efficient Decentralized Multi-Agent Reinforcement Learning for Cooperative Adaptive Cruise Control. doi: https://doi.org/10.1109/TIV.2024.3368025.

[188] Jayasinghe, R. (2019). Reinforcement learning based traffic optimization at an intersection with GLOSA. Available at: https://monarch.qucosa.de/api/qucosa%3A36114/attachment/ATT-0/.

[189] Tajalli, M. Mehrabipour, M. Hajbabaie, A. (2020). Network-Level Coordinated Speed Optimization and Traffic Light Control for Connected and Automated Vehicles. doi: https://doi.org/10.1109/TITS.2020.2994468.

[190] Nguyen, C, H, P. Hoang, N, H. Lee, S. Vu, H, L. (2021). A System Optimal Speed Advisory Framework for a Network of Connected and Autonomous Vehicles. doi: https://doi.org/10.1109/TITS.2021.3056696.

[191] Tan, C. Yang, K. (2023). Privacy-Preserving Adaptive Traffic Signal Control in a Connected Vehicle Environment. doi: https://doi.org/10.48550/arXiv.2305.07212.

[192] Rashidi, T, H. Najmi, A. Haider, A. Wang, C. Hoseinzadeh, F. (2019). What we know and do not know about connected and autonomous vehicles. doi: https://doi.org/10.1080/23249935.2020.1720860.

[193] Campisi, T. Severino, A. Al-Rashid M, A. Pau, G. (2021). The Development of the Smart Cities in the Connected and Autonomous Vehicles (CAVs) Era: From Mobility Patterns to Scaling in Cities. doi: https://doi.org/10.3390/infrastructures6070100.

[194] Jiang, L. Chen, H. Paschalidis, E. (2023). Diffusion of connected and autonomous vehicles concerning mode choice, policy interventions and sustainability impacts: A system dynamics modelling study. doi: https://doi.org/10.1016/j.tranpol.2023.07.029.

[195] Mnih, V. Kavukcuoglu, K. Silver, D. Graves, A. Antonoglou, I. Wierstra, D. Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. doi: https://doi.org/10.48550/arXiv.1312.5602.

[196] Mnih, V. Badia, A, P. Mirza, M. Graves, A. Lillicrap, T, P. Harley, T. Silver, D. Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. doi: https://doi.org/10.48550/arXiv.1602.01783.

[197] Schulman, J. Wolski, F. Dhariwal, P. Radford, A. Klimov, O. (2017). Proximal Policy Optimization Algorithms. doi: https://doi.org/10.48550/arXiv.1707.06347.

[198] Hasselt, H. (2010). Double Q-learning. Available at: https://proceedings.neurips.cc/paper_files/paper/2010/file/091d584fced301b442654dd8c23b3fc9-Paper.pdf.

[199] Hasselt, H. Guez, A. Silver, D. (2015). Deep Reinforcement Learning with Double Q-learning. doi: https://doi.org/10.48550/arXiv.1509.06461.

[200] Schaul, T. Quan, J. Antonoglou, I. Silver, D. (2015). Prioritized Experience Replay. doi: https://doi.org/10.48550/arXiv.1511.05952.

[201] Wang, Z. Schaul, T. Hessel, M. Hasselt, H. Lanctot, M. Freitas, N. (2016). Dueling Network Architectures for Deep Reinforcement Learning. doi: https://doi.org/10.48550/arXiv.1511.06581.

[202] Fortunato, M. Gheshlaghi Azar, M. Piot, B. Menick, J. Osband, I. Graves, A. Mnih, V. Munos, R. Hassabis, D. Pietquin, O. Blundell, C. Legg, S. (2019). Noisy Networks for Exploration. doi: https://doi.org/10.48550/arXiv.1706.10295.

[203] Osband, I. Van Roy, B. Russo, D. Wen, Z. (2019). Deep Exploration via Randomized Value Functions. doi: https://doi.org/10.48550/arXiv.1703.07608.

[204] Bellemare, M, G. Dabney, W. Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. doi: https://doi.org/10.48550/arXiv.1707.06887.

[205] Meng, L. Gorbet, R. Kulić, D. (2020). The Effect of Multi-step Methods on Overestimation in Deep Reinforcement Learning. doi: https://doi.org/10.48550/arXiv.2006.12692.

[206] Hernandez-Garcia, J, F. Sutton, R, S. (2019). Understanding Multi-Step Deep Reinforcement Learning: A Systematic Study of the DQN Target. doi: https://doi.org/10.48550/arXiv.1901.07510.

[207] Breuel, T, M. (2015). The Effects of Hyperparameters on SGD Training of Neural Networks. doi: https://doi.org/10.48550/arXiv.1508.02788.

[208] Frazier, P, I. (2018). A Tutorial on Bayesian Optimization. doi: https://doi.org/10.48550/arXiv.1807.02811.

[209] Mathworks. Gaussian Process Regression Models. Available at: https://www.mathworks.com/help/stats/gaussian-process-regression-models.html. (Accessed 12 Mar. 24).

[210] ekamperi. (2021). Acquisition functions in Bayesian Optimization. Available at: https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html#upper-confidence-bound-ucb. (Accessed 12 Mar. 24).

[211] Schneider, C. (2020). Deep Q-Learning in Traffic Signal Control: A Literature Review. Available at: https://repository.tudelft.nl/islandora/object/uuid:259debb3-4583-4bb1-9cd9-a8d5b88186e4/datastream/OBJ1/download (Accessed 8 Nov. 2021).

[212] Yunex Traffic. (2024). Aimsun. Available at: https://www.aimsun.com/. (Accessed 18 Mar. 2024)

[213] Zhang, R. Ishikawa, A. Wang, W. Striner, B. Tonguz, O, K. (2020). Using Reinforcement Learning With Partial Vehicle Detection for Intelligent Traffic Signal Control. doi: https://doi.org/10.1109/TITS.2019.2958859.

[214] PTV Planung Transport Verkehr AG. (2023). PTV Vissim. Available at: https://www.ptvgroup.com/en/products/ptv-vissim. (Accessed 18 Mar. 2024)

[215] Genders, W. Razavi, S. (2019) An Open-Source Framework for Adaptive Traffic Signal Control. doi: https://doi.org/10.48550/arXiv.1909.00395.

[216] Eclipse. (2024). SUMO User Documentation. Available at: https://sumo.dlr.de/docs/index.html. (Accessed 18 Mar. 2024)

[217] Zeng, J. Hu, J. Zhang, Y. (2019). Training Reinforcement Learning Agent for Traffic Signal Control under Different Traffic Conditions. doi: https://doi.org/10.1109/ITSC.2019.8917342.

[218] Kimber, R, M. McDonald, M. Hounsell, N, B. The Prediction of Saturation Flows for Road Junctions Controlled by Traffic Signals

[219] Choe, C. Baek, S. Woon, B. Kong, S. (2018). Deep Q Learning with LSTM for Traffic Light Control. doi: https://doi.org/10.1109/APCC.2018.8633520.

[220] APSEd. Traffic Signal Design Webster's Formula for Optimum Cycle Length. Available at: https://www.apsed.in/post/traffic-signal-design-webster-s-formula-for-optimum-cycle-length.

[221] Abdoos, M. Mozayani, N. Bazzan, A. (2013). Holonic multi-agent system for traffic signals control. doi: https://doi.org/10.1016/j.engappai.2013.01.007.

[222] JCT Consultancy. LinSig 3. Available at: http://www.jctconsultancy.co.uk/Software/LinSigV3/linsigv3.php.

[223] Liang, X. Du, X. Wang, G. Han, X. (2018). Deep reinforcement learning for traffic light control in vehicular networks. doi: https://doi.org/10.48550/arXiv.1803.11115.

[224] sumo.dlr.de. GLOSA. Available at:https://sumo.dlr.de/docs/Simulation/GLOSA.html.

[225] Zheng, L. Wu, B. (2022). A Reinforcement Learning Based Traffic Control Strategy in a Macroscopic Fundamental Diagram Region. doi: https://doi.org/10.1155/2022/5681234.

[226] JCT Consultancy. LinSig 3.2 User Guide SCATS Version. Available at: https://www.jctconsultancy.co.uk/Support/Manuals/LinSig32_User_Guide_SCATS.pdf.

[227] Yang, S. Yang, B. Zeng, Z. Kang, Z. (2023). Causal inference multi-agent reinforcement learning for traffic signal control. doi: https://doi.org/10.1016/j.inffus.2023.02.009.

[228] Guo, M. Wang, P. Chan, C, Y. Askary, S. (2019). A Reinforcement Learning Approach for Intelligent Traffic Signal Control at Urban Intersections. doi: https://doi.org/10.1109/ITSC.2019.8917268.

[229] Genders, W. Razavi, S. (2016). Using a Deep Reinforcement Learning Agent for Traffic Signal Control. doi: https://doi.org/10.48550/arXiv.1611.01142.

[230] Zhao, T. Wang, P. Li, S. (2019). Traffic Signal Control with Deep Reinforcement learning. doi: https://doi.org/10.1109/ICICAS48597.2019.00164.

[231] Kim, J. Jung, S. Kim, K. Lee, S. (2019). The Real-Time Traffic Signal Control System for the Minimum Emission using Reinforcement Learning in Vehicle-to-Everything (V2X) Environment. Chemical Engineering Transactions, 72, 91-96. doi: https://doi.org/10.3303/CET1972016.

[232] Nawar, M. Fares, A. Al-Sammak, A. (2019). Rainbow Deep Reinforcement Learning Agent for Improved Solution of the Traffic Congestion. doi: https://doi.org/10.1109/JAC-ECC48896.2019.9051262.

[233] Zeng, J. Hu, J. Zhang, Y. (2018). Adaptive Traffic Signal Control with Deep Recurrent Q-learning. doi: https://doi.org/10.1109/IVS.2018.8500414.

[234] Mousavi, S, S. Schukat, M. Howley, E. (2018). Deep Reinforcement Learning: An Overview. Lecture Notes in Networks and Systems. 426-440. doi: https://doi.org/10.1007/978-3-319-56991-8_32.

[235] Wang, L. Ma, Z. Dong, C. Wang, H. (2022). Human-centric multimodal deep (HMD) traffic signal control. doi: https://doi.org/10.1049/itr2.12300.

[236] Zhao, X. Flocco, D. Azarm, S. Balachandran, B. (2023). Deep Reinforcement Learning for the Co-Optimization of Vehicular Flow Direction Design and Signal Control Policy for a Road Network. doi: https://doi.org/10.1109/ACCESS.2023.3237420.

[237] Kolat, M. Kővári, B. Bécsi, T. Aradi, S. (2023). Multi-Agent Reinforcement Learning for Traffic Signal Control: A Cooperative Approach. doi: https://doi.org/10.3390/su15043479.

[238] Cao, Y. Ireson, N. Bull, L. Miles, R. (1999). Design of a traffic junction controller using classifier system and fuzzy logic. doi: https://doi.org/10.1007/3-540-48774-3_40.

[239] Ritcher, S. (2007). Traffic light scheduling using policy-gradient reinforcement learning. Available at: https://icaps07-satellite.icaps-conference.org/dc/dc-23.pdf (Accessed 10 Mar. 2024).

[240] Anusha, S, P. Vanajakshi, L, D. and Sharma, A. (2013). A Simple Method for Estimation of Queue Length. Civil Engineering Faculty Publications. 48. Available at: http://digitalcommons.unl.edu/civilengfacpub/48. (Accessed 7 Mar. 2024).

[241] Wiering, M. Vreeken, J. Van Veenen, J. Koopman, A. (2004). Simulation and optimization of traffic in a city. doi: https://doi.org/10.1109/IVS.2004.1336426.

[242] Dusparic, I. Cahill, V. (2009). Using reinforcement learning for multi-policy optimization in decentralized autonomic systems–an experimental evaluation. doi: https://doi.org/10.1007/978-3-642-02704-8_9.

[243] Chanloha, P. Chinrungrueng, J. Usaha, W. Aswakul, C. (2013). Cell transmission model-based multiagent q-learning for network-scale signal control with transit priority. doi: https://doi.org/10.1093/comjnl/bxt126.

[244] Aziz, H, A. Zhu, F. Ukkusuri, S, V. Learning-based traffic signal control algorithms with neighborhood information sharing: An application for sustainable mobility. doi: https://doi.org/10.1080/15472450.2017.1387546.

[245] Jin, J. Ma, X. (2015). Adaptive group-based signal control using reinforcement learning with eligibility traces. doi: https://doi.org/10.1109/ITSC.2015.389.

[246] Lu, C. Wen, F. Gen, M. (2017). Traffic lights dynamic timing algorithm based on reinforcement learning. doi: https://doi.org/10.1007/978-3-319-59280-0_147.

[247] Ren, F. Dong, W. Zhao, X. Zhang, F. Kong, Y. Yang, Q. (2023). Two-layer coordinated reinforcement learning for traffic signal control in traffic network. doi: https://doi.org/10.1016/j.eswa.2023.121111.

[248] Ghanadbashi, S. Golpayegani, F. (2022). Using ontology to guide reinforcement learning agents in unseen situations. doi: https://doi.org/10.1007/s10489-021-02449-5.

[249] Prashanth, L. Bhatnagar, S. (2011). Reinforcement learning with average cost for adaptive control of traffic lights at intersections. doi: https://doi.org/10.1109/ITSC.2011.6082823.

[250] Mikami, S. Kakazu, Y. (1994). Genetic reinforcement learning for cooperative traffic signal control. doi: https://doi.org/10.1109/ICEC.1994.350012.

[251] Müller, A. Rangras, V. Ferfers, T. Hufen, F. Schreckenberg, L. Jasperneite, J. Schnittker, G. Waldmann, M. Friesen, M. Wiering M. (2021). Towards Real-World Deployment of Reinforcement Learning for Traffic Signal Control. doi: https://doi.org/10.1109/ICMLA52953.2021.00085.

[252] Chu, T. Qu, S. Wang, J. (2016). Large-scale multi-agent reinforcement learning using image-based state representation. doi: https://doi.org/10.1109/CDC.2016.7799442.

[253] Li, L. Lv, Y. Wang, R. (2016). Traffic signal timing via deep reinforcement learning. IEEE/CAA Journal of Automatica Sinica, Volume 3, Issue 3. doi: https://doi.org/10.1109/JAS.2016.7508798.

[254] Wei, H. Zheng, G. Yao, H. Li, Z. (2018). IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. doi: https://doi.org/10.1145/3219819.3220096.

[255] Aziz, H, M, A. Wang, H. Young, S. Bin al islam, S, M, A. (2019). Investigating the Impact of Connected Vehicle Market Share on the Performance of Reinforcement-Learning Based Traffic Signal Control. United States: N. p., 2019. Web. doi:10.2172/1566974.

[256] Fang, S. Chen, F. Liu, H. (2019). Dueling Double Deep Q-Network for Adaptive Traffic Signal Control with Low Exhaust Emissions in A Single Intersection. doi: https://doi.org/10.1088/1757-899X/612/5/052039.

[257] Chen, P. Zhu, Z. Lu, G. (2019). An Adaptive Control Method for Arterial Signal Coordination Based on Deep Reinforcement Learning. doi: https://doi.org/10.1109/ITSC.2019.8917051.

[258] Box, S. Waterson, B. (2012). An automated signalized junction controller that learns strategies from a human expert. doi: https://doi.org/10.1016/j.engappai.2011.09.008.

[259] Wang, Z. Liu, C. (2006). An Empirical Evaluation of the Loop Detector Method for Travel Time Delay Estimation. doi: https://doi.org/10.1080/15472450500237254.

[260] Cao, Y. Ireson, N. Bull, L. Miles, R. (2000). Distributed Learning Control of Traffic Signals. doi: https://doi.org/10.1007/3-540-45561-2_12.

[261] Wiering, M, A. (2000). Multi-agent reinforcement learning for traffic light control. Available at: https://www.researchgate.net/publication/221346141_Multi-Agent_Reinforcement_Learning_for_Traffic_Light_Control (Accessed 10 Mar. 2024).

[262] Abdulhai, B. Pringle, R. Karakoulas, G, J. (2003). Reinforcement Learning for True Adaptive Traffic Signal Control. doi: https://doi.org/10.1061/(ASCE)0733-947X(2003)129:3(278).

[263] Da Silva, B, C. Basso, E, W. Bazzan, A. Engel, P, M. (2006). Dealing with non-stationary environments using context detection. doi: https://doi.org/10.1145/1143844.1143872.

[264] de Oliveira, D. Bazzan, A. da Silva, B, C, Basso, E, W. Nunes, L. Rossetti, R. (2006). Reinforcement learning based control of traffic lights in non-stationary environments. doi: https://doi.org/10.1016/j.jer.2023.100017.

[265] Su, S. Tham, C, K. (2007). Sensor Grid for Real-Time Traffic Management. doi: https://doi.org/10.1109/ISSNIP.2007.4496884.

[266] Salkham, A. Cunningham, R. Garg, A. Cahill, V. (2008). A Collaborative Reinforcement Learning Approach to Urban Traffic Control Optimization. doi: https://doi.org/10.1109/WIIAT.2008.88.

[267] Lu, S. Liu, X. Dai, S. (2008). Adaptive and Coordinated Traffic Signal Control Based on Q-Learning and MULTIBAND Model. doi: https://doi.org/10.1109/ICCIS.2008.4670819.

[268] Dusparic, I. Cahill, V. (2009). Distributed w-learning: Multi-policy optimization in self-organizing systems. doi: https://doi.org/10.1109/SASO.2009.23.

[269] Li, C, G. Wang, M. Sun, Z, G. Lin, F, Y. Zhang, Z, F. (2009). Urban traffic signal learning control using fuzzy actor-critic methods. doi: https://doi.org/10.1109/ICNC.2009.374.

[270] Dai, Y. Zhao, D. Yi, J. (2010). A comparative study of urban traffic signal control with reinforcement learning and adaptive dynamic programming. doi: https://doi.org/10.1109/IJCNN.2010.5596480.

[271] Medina, J, C. Hajbabaie, A. Benekohal, R, F. (2010). Arterial traffic control using reinforcement learning agents and information from adjacent intersections in the state and reward structure. doi: https://doi.org/10.1109/ITSC.2010.5624977.

[272] Waskow, S, J. Bazzan, A. (2010). Improving space representation in multiagent learning via tile coding. doi: https://doi.org/10.1007/978-3-642-16138-4_16.

[273] Bazzan, A. De Oliveira, D. da Silva, B, C. (2010). Learning in groups of traffic signals. doi: https://doi.org/10.1016/j.engappai.2009.11.009.

[274] Prashanth, L, A. Bhatnagar, S. (2010). Reinforcement Learning With Function Approximation for Traffic Signal Control. doi: https://doi.org/10.1109/TITS.2010.2091408.

[275] Arel, I. Liu, C. Urbanik, T. Kohls, A, G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. doi: https://doi.org/10.1049/iet-its.2009.0070.

[276] Box, S. Waterson, B. (2011). An automated signalized junction controller that learns strategies from a human expert. doi: https://doi.org/10.1016/j.engappai.2011.09.008.

[277] Heinen, M, R. Bazzan, A. Engel, P, M. (2011). Dealing with continuous-state reinforcement learning for intelligent control of traffic signals. doi: https://doi.org/10.1109/ITSC.2011.6083107.

[278] Dai, Y. Hu, J. Zhao, D. Zhu, F. (2011). Neural network based online traffic signal controller design with reinforcement training. doi: https://doi.org/10.1109/ITSC.2011.6083027.

[279] Abdoos, M. Mozayani, N. Bazzan, A. (2011). Traffic light control in non-stationary environments based on multi agent q-learning. doi: https://doi.org/10.1109/ITSC.2011.6083114.

[280] Khamis, M, A. Gomaa, W. (2012). Enhanced Multiagent Multi-Objective Reinforcement Learning for Urban Traffic Light Control. doi: https://doi.org/10.1109/ICMLA.2012.108.

[281] El-Tantawy, S. Abdulhai, B. (2012). Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto. doi: https://doi.org/10.1109/TITS.2013.2255286.

[282] Khamis, M, A. Gomaa, W. El-Shishiny, H. (2012). Multi-objective traffic light control system based on Bayesian probability interpretation. doi: https://doi.org/10.1109/ITSC.2012.6338853.

[283] Prashanth, L, A. Bhatnagar, S. (2012). Threshold Tuning Using Stochastic Optimization for Graded Signal Control. doi: https://doi.org/10.1109/TVT.2012.2209904.

[284] Pham, T, T. Brys, T. Taylor, M, E. (2013). Learning Coordinated Traffic Light Control. Available at: https://www.researchgate.net/publication/235759037_Learning_Coordinated_Traffic_Light_Control (Accessed 10 Mar. 2024).

[285] Nuli, S. Mathew, T, V. (2013). Online coordination of signals for heterogeneous traffic using stop line detection. doi: https://doi.org/10.1016/j.sbspro.2013.11.171.

[286] Chin, Y, K. Tham, H, J. Rao, N, K. Bolong, N. Teo, K, T, K. (2013). Optimization of urban multi-intersection traffic flow via q-learning. doi: https://doi.org/10.21917/ijsc.2013.0073.

[287] Xu, L, H. Xia, X, H. Luo, Q. (2013). The Study of Reinforcement Learning for Traffic Self-Adaptive Control under Multiagent Markov Game Environment. doi: https://doi.org/10.1155/2013/962869.

[288] Moghadam, M, H. Mozayani, N. (2013). Urban Traffic Control Using Adjusted Reinforcement Learning in a Multi-agent System. doi: https://doi.org/10.19026/rjaset.6.3676.

[289] Teo, K, T, K. Yeo, K, B. Chin, Y, K. Chuo, H, S, E. Tan, M, K. (2014). Agent-based optimization for multiple signalized intersections using q-learning International Journal of Simulation. doi: https://doi.org/10.5013/IJSSST.a.15.06.10.

[290] Brys, T. Nowé, A. Kudenko, D. Taylor, M. (2014). Combining multiple correlated reward and shaping signals by measuring confidence. doi: https://doi.org/10.1609/aaai.v28i1.8998.

[291] Liu, W. Liu, J. Peng, J. Zhu, Z. (2014). Cooperative multi-agent traffic signal control system using fast gradient-descent function approximation for v2i networks. doi: https://doi.org/10.1109/ICC.2014.6883709.

[292] Abdoos, M. Mozayani, N. Bazzan, A. (2014). Hierarchical control of traffic signals using Q-learning with tile coding. doi: https://doi.org/10.1007/s10489-013-0455-3.

[293] Marsetič, R. Šemrov, D. Žura, M. (2014). Road artery traffic light optimization with use of the reinforcement learning. doi: https://doi.org/10.7307/ptt.v26i2.1318.

[294] Jadhao, N, S. Jadhao, A, S. (2014). Traffic signal control using reinforcement learning. doi: https://doi.org/10.1109/CSNT.2014.231.

[295] Zhu, F. Aziz, H, A. Qian, X. Ukkusuri, S, V. (2015). A junction-tree based learning algorithm to optimize network wide traffic control: A coordinated multi-agent framework. doi: https://doi.org/10.1016/j.trc.2014.12.009.

[296] Mashayekhi, M. List, G. (2015). A Multiagent Auction-based Approach for Modelling of Signalized Intersections. Available at: https://www.researchgate.net/publication/283052078_A_Multiagent_Auction-based_Approach_for_Modeling_of_Signalized_Intersections (Accessed 10 Mar. 2024).

[297] Yin, B. Dridi, M. Moudni, A, E. (2015). Adaptive traffic signal control for multi-intersection based on microscopic model. doi: https://doi.org/10.1109/ICTAI.2015.21.

[298] Prabuchandran, K. An, H, K. Bhatnagar, S. (2015). CVLight: Decentralized Learning for Adaptive Traffic Signal Control with Connected Vehicles. doi: https://doi.org/10.48550/arXiv.2104.10340.

[299] Araghi, S. Khosravi, A. Creighton, D. (2015). Distributed q-learning controller for a multi-intersection traffic network. doi: https://doi.org/10.1007/978-3-319-26532-2_37.

[300] Ajorlou, A. Awasthi, A. Aghdam, A, G. (2015). Distributed urban traffic control based on locally observable cell occupancies. doi: https://doi.org/10.1109/ACC.2015.7170869.

[301] Tahifa, M. Boumhidi, J. Yahyaouy, A. (2015). Swarm reinforcement learning for traffic signal control based on cooperative multi-agent framework. doi: https://doi.org/10.1109/ISACV.2015.7105536.

[302] Abdoos, M. Mozayani, N. Bazzan, A. (2015). Towards reinforcement learning for holonic multi-agent systems Intelligent Data Analysis. doi: https://doi.org/10.3233/IDA-150714.

[303] Van der Pol, E. Oliehoek, F, A. (2016). Coordinated deep reinforcement learners for traffic light control. Available at: https://pure.uva.nl/ws/files/136941048/vanderpol_oliehoek_nipsmalic2016.pdf (Accessed 10 Mar. 2024).

[304] Van der Pol, E. (2016). Deep reinforcement learning for coordination in traffic light control. doi: https://doi.org/10.48550/arXiv.2302.03669.

[305] Rosyadi, A, R. Wirayuda, T, A, B. Al-Faraby, S. (2016). Intelligent traffic light control using collaborative q-learning algorithms. doi: https://doi.org/10.1109/ITSC.2017.8317730.

[306] Gaikwad, V, V. Kadarkar, S, S. Kasbekar, G, S. (2016). Intelligent traffic signal duration adaptation using q-learning with an evolving state space. doi: https://doi.org/10.1109/VTCFall.2016.7881050.

[307] Jin, J. Ma, X. (2017). A multi-objective agent-based approach for road traffic controls: application for adaptive traffic signal systems. Available at: https://www.diva-portal.org/smash/get/diva2:1205233/FULLTEXT01.pdf (Accessed 10 Mar. 2024).

[308] Gao, J. Shen, Y. Liu, J. Ito, M. Shiratori, N. (2017). Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. doi: https://doi.org/10.48550/arXiv.1705.02755.

[309] Liu, W. Qin, G. He, Y. Jiang, F. (2017). Distributed cooperative reinforcement learning-based traffic signal control that integrates v2x networks' dynamic clustering. doi: https://doi.org/10.1109/TVT.2017.2702388.

[310] Vidhate, D, A. Kulkarni, P. (2017). Exploring cooperative multi-agent reinforcement learning algorithm (cmrla) for intelligent traffic signal control. doi: https://doi.org/10.1007/978-981-13-1423-0_9.

[311] Liu, Y. Liu, L. Chen, W, P. (2017). Intelligent traffic light control using distributed multi-agent q learning. doi: https://doi.org/10.1109/ITSC.2017.8317730.

[312] Darmoul, S. Elkosantini, S. Louati, A. Said, L, B. (2017). Multi-agent immune networks to control interrupted flow at signalized intersections. doi: https://doi.org/10.1016/j.trc.2017.07.003.

[313] Kristensen, T. Ezeora, N, J. (2017). Simulation of intelligent traffic control for autonomous vehicles. doi: https://doi.org/10.1109/ICInfA.2017.8078952.

[314] Mousavi, S, S. Schukat, M. Howley, E. (2017). Traffic light control using deep policy-gradient and value-function-based reinforcement learning. doi: https://doi.org/10.48550/arXiv.1704.08883.

[315] Li, C. Yan, F. Zhou, Y. Wu, J. Wang, X. (2018). A regional traffic signal control strategy with deep reinforcement learning. doi: https://doi.org/10.23919/ChiCC.2018.8483361.

[316] Torabi, B. Wenkstern, R, Z. Saylor, R. (2018). A Self-Adaptive Collaborative Multi-Agent based Traffic Signal Timing System. doi: https://doi.org/10.1109/ISC2.2018.8656659.

[317] Vinitsky, E. Kreidieh, A. Flem, L, L. Kheterpal, N. Jang, K. Wu, C. (2018). Benchmarks for reinforcement learning in mixed-autonomy traffic. Available at: https://proceedings.mlr.press/v87/vinitsky18a.html (Accessed 10 Mar. 2024).

[318] Lemos, L, L. Bazzan, A. Pasin, M. (2018). Co-adaptive reinforcement learning in microscopic traffic systems. doi: https://doi.org/10.1109/CEC.2018.8477713.

[319] Aslani, M. Seipel, S. Wiering, M. (2018). Continuous residual reinforcement learning for traffic signal control optimization. doi: https://doi.org/10.1139/cjce-2017-0408.

[320] Shi, S. Chen, F. (2018). Deep recurrent q-learning method for area traffic coordination control. doi: https://doi.org/10.9734/JAMCS/2018/41281.

[321] Daeichian, A. Haghani, A. (2018). Fuzzy q-learning-based multi-agent system for intelligent traffic control by a game theory approach. doi: https://doi.org/10.1007/s13369-017-3018-9.

[322] Jin, J. Ma, X. (2018). Hierarchical multi-agent control of traffic lights based on collective learning. doi: https://doi.org/10.1016/j.engappai.2017.10.013.

[323] Al Islam, S, B. Aziz, H, A. Wang, H. Young, S, E. (2018). Minimizing energy consumption from connected signalized intersections by reinforcement learning. doi: https://doi.org/10.1109/ITSC.2018.8569891.

[324] Aslani, M. Seipel, S. Mesgari, M, S. Wiering, M. (2018). Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran. doi: https://doi.org/10.1016/j.aei.2018.08.002.

[325] Wan, C, H. Hwang, M, C. (2018). Value-based deep reinforcement learning for adaptive isolated intersection signal control. doi: https://doi.org/10.1049/iet-its.2018.5170.

[326] Liang, X. (2019). A deep reinforcement learning network for traffic light cycle control. doi: https://doi.org/10.1109/TVT.2018.2890726.

[327] Jin, J. Ma, X. (2019). A multi-objective agent-based control approach with application in intelligent traffic signal system. doi: https://doi.org/10.1109/TITS.2019.2906260.

[328] Gan, X. Guo, H. Li, Z. (2019). A new multi-agent reinforcement learning method based on evolving dynamic correlation matrix. doi: https://doi.org/10.1109/ACCESS.2019.2946848.

[329] Wei, H. Xu, N. Zhang, H. Zheng, G. Zang, X. Chen, C. Zhang, W. Zhu, Y. Xu, K. Li, Z. (2019). Colight: Learning network-level cooperation for traffic signal control. doi: https://doi.org/10.48550/arXiv.1905.05717.

[330] Ge, H. Song, Y. Wu, C. Ren, J. Tan, G. (2019). Cooperative deep q-learning with value transfer for multi-intersection signal control. doi: https://doi.org/10.1109/ACCESS.2019.2907618.

[331] Huang, R. Hu, J. Huo, Y. Pei, X. (2019). Cooperative multi-intersection traffic signal control based on deep reinforcement learning. doi: https://doi.org/10.1109/ACCESS.2020.3034419.

[332] Yang, S. Yang, B. Wong, H, S. Kang, Z. (2019). Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm. doi: https://doi.org/10.1016/j.knosys.2019.07.026.

[333] Gong, Y. Abdel-Aty, M. Cai, Q. Rahman, M, S. (2019). Decentralized network level adaptive signal control by multi-agent deep reinforcement learning,. doi: `https://doi.org/10.1016/j.trip.2019.100020`.

[334] Tan, K, L. Poddar, S. Sarkar, S. Sharma, A. (2019). DEEP REINFORCEMENT LEARNING FOR ADAPTIVE TRAFFIC SIGNAL CONTROL. Available at: `https://www.semanticscholar.org/reader/fd5775d6bef6c5e55b50b1df995a56bd0851af37` (Accessed 10 Mar. 2024).

[335] Wang, S. Xie, X. Huang, K. Zeng, J. Cai, Z. (2019). Deep reinforcement learning-based traffic signal control using high-resolution event-based data. doi: `https://doi.org/10.3390/e21080744`.

[336] Aslani, M. Mesgari, M, S. Seipel, S. Wiering, M. (2019). Developing adaptive traffic signal control by actor–critic and direct exploration methods. doi: `https://doi.org/10.1680/jtran.17.00085`.

[337] Zhou, P. Braud, T. Alhilal, A. Hui, P. Kangasharju, J. (2019). ERL: Edge Based Reinforcement Learning for Optimized Urban Traffic Light Control. doi: `https://doi.org/10.1109/PERCOMW.2019.8730706`.

[338] Shu, L. Wu, J. Li, Z. (2019). Hierarchical regional control for traffic grid signal optimization. doi: `https://doi.org/10.1109/ITSC.2019.8917513`.

[339] Reda, M. Mountassir, F. Mohamed, B. (2019). Introduction to coordinated deep agents for traffic signal. doi: `https://doi.org/10.1109/WITS.2019.8723846`.

[340] 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. doi: `https://doi.org/10.1145/3357384.3357900`.

[341] Chu, T. Wang, J. Codecà, L. Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. doi: `https://doi.org/10.48550/arXiv.1903.04527`.

[342] Higuera, C. Lozano, F. Camacho, E, C. Higuera, C, H. (2019). Multiagent reinforcement learning applied to traffic light signal control. doi: `https://doi.org/10.1007/978-3-030-24209-1_10`.

[343] Shabestray, S, M, A. Abdulhai, B. (2019). Multimodal intelligent deep (mind) traffic signal controller. doi: `https://doi.org/10.1109/ITSC.2019.8917493`.

[344] 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. doi: `https://doi.org/10.1145/3292500.3330949`.

[345] Horsuwan, T. Aswakul, C. (2019). Reinforcement learning agent under partial observability for traffic light control in presence of gridlocks. Available at: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjVwKmI9rSCAxUMSUEAHaerCT4QFnoECBIQAQ&url=https%3A%2F%2Feasychair.org%2Fpublications%2Fdownload%2FnsT5&usg=AOvVaw3wlyMCL9sUY-OqqgoQ_z3-&opi=89978449 (Accessed 10 Mar. 2024).

[346] Rizzo, S, G. Vantini, G. Chawla, S. (2019). Reinforcement learning with explainability for traffic signal control. doi: https://doi.org/10.1109/ITSC.2019.8917519.

[347] Rizzo, S, G. Vantini, G. Chawla, S. (2019). Time critic policy gradient methods for traffic signal control in complex and congested scenarios. doi: https://doi.org/10.1145/3292500.3330988.

[348] Tan, T. (2020). Cooperative deep reinforcement learning for large-scale traffic grid signal control. doi: https://doi.org/10.1109/TCYB.2019.2904742.

[349] Kim, D. Jeong, O. (2020). Cooperative traffic signal control with traffic flow prediction in multi-intersection. doi: https://doi.org/10.3390/s20010137.

[350] Kumar, N. Rahman, S, S. Dhakad, N. (2020). Fuzzy Inference Enabled Deep Reinforcement Learning-Based Traffic Light Control for Intelligent Transportation System. doi: https://doi.org/10.1109/TITS.2020.2984033.

[351] Wang, X. Ke, L. Qiao, Z. Chai, X. (2020). Large-Scale Traffic Signal Control Using a Novel Multiagent Reinforcement Learning. doi: https://doi.org/10.1109/TCYB.2020.3015811.

[352] Zang, X. Yao, H. Zheng, G. Xu, N. Xu, K. Li, Z. (2020). MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control. doi: https://doi.org/10.1609/aaai.v34i01.5467.

[353] Chu, T. Wang, J. Codecà, L. Li, Z. (2020). Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. doi: https://doi.org/10.1109/TITS.2019.2901791.

[354] Wu, T. Zhou, P. Liu, K. Yuan, Y. Wang, X. Huang, H. Wu, D, O. (2020). Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks. doi: https://doi.org/10.1109/TVT.2020.2997896.

[355] Xu, M. Wu, J. Huang, L. Zhou, R. Wang, T. Hu, D. (2020). Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning. doi: https://doi.org/10.1080/15472450.2018.1527694.

[356] Genders, W. Razavi, S. (2020). Policy analysis of adaptive traffic signal control using reinforcement learning. doi: https://doi.org/10.1061/(ASCE)CP.1943-5487.0000859.

[357] Padakandla, S. Prabuchandran, K, J. Bhatnagar, S. (2020). Reinforcement learning algorithm for non-stationary environments. doi: https://doi.org/10.1007/s10489-020-01758-5.

[358] Lee, J. Chung, J. Sohn, K. (2020). Reinforcement learning for joint control of traffic signals in a transportation network. doi: https://doi.org/10.1109/TVT.2019.2962514.

[359] Wang, Y. Xu, T. Niu, X. Tan, C. Chen, E. Xiong, H. (2020). STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control. doi: https://doi.org/10.1109/TMC.2020.3033782.

[360] Chen, C. 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. doi: https://doi.org/10.1609/aaai.v34i04.5744.

[361] Joo, H. Ahmed, S, H. Lim, Y. (2020). Traffic signal control for smart cities using reinforcement learning. doi: https://doi.org/10.1016/j.comcom.2020.03.005.

[362] Ma, D. Zhou, B. Song, X. Dai, H. (2021). A Deep Reinforcement Learning Approach to Traffic Signal Control With Temporal Traffic Pattern Mining. doi: https://doi.org/10.1109/TITS.2021.3107258.

[363] Boukerche, A. Zhong, D. Sun, P. (2021). A Novel Reinforcement Learning-Based Cooperative Traffic Signal System Through Max-Pressure Control. doi: https://doi.org/10.1109/TVT.2021.3069921.

[364] Shashi, F, I. Sultan, S, M. Khatun, A. Sultana, T. Alam, T. (2021). A Study on Deep Reinforcement Learning Based Traffic Signal Control for Mitigating Traffic Congestion. doi: https://doi.org/10.1109/ECBIOS51820.2021.9510422.

[365] Wang, T. Cao, J. Hussain, A. (2021). Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. doi: https://doi.org/10.1016/j.trc.2021.103046.

[366] Guo, G. Wang, Y. (2021). An integrated MPC and deep reinforcement learning approach to trams-priority active signal control. doi: https://doi.org/10.1016/j.conengprac.2021.104758.

[367] Zeng, Z. (2021). GraphLight: Graph-based Reinforcement Learning for Traffic Signal Control. doi: https://doi.org/10.1109/ICCCS52626.2021.9449147.

[368] Abdoos, M. Bazzan, A, L, C. (2021). Hierarchical traffic signal optimization using reinforcement learning and traffic prediction with long-short term memory. doi: https://doi.org/10.1016/j.eswa.2021.114580.

[369] Xu, B. Wang, Y. Wang, Z. Jia, H. Lu, Z. (2021). Hierarchically and Cooperatively Learning Traffic Signal Control. doi: https://doi.org/10.1609/aaai.v35i1.16147.

[370] Devailly, F, X. Larocque, D. Charlin, L. (2021). IG-RL: Inductive Graph Reinforcement Learning for Massive-Scale Traffic Signal Control. doi: https://doi.org/10.1109/TITS.2021.3070835.

[371] Yang, S. Yang, B. Kang, Z. Deng, L. (2021). IHG-MA: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control. doi: https://doi.org/10.1016/j.neunet.2021.03.015.

[372] Li, Z. Yu, H. Zhang, G. Dong, S. Xu, C, Z. (2021). Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. doi: https://doi.org/10.1016/j.trc.2021.103059.

[373] Mushtaq, A. Haq, I, U. Imtiaz, M, U. Khan, A. Shafiq, O. (2021). Traffic Flow Management of Autonomous Vehicles Using Deep Reinforcement Learning and Smart Rerouting. doi: https://doi.org/10.1109/ACCESS.2021.3063463.

[374] Song, L. Fan, W. (2021). Traffic Signal Control Under Mixed Traffic With Connected and Automated Vehicles: A Transfer-Based Deep Reinforcement Learning Approach. doi: https://doi.org/10.1109/ACCESS.2021.3123273.

[375] Chu, K, F. Lam, A, Y, S. Li, V, O, K. (2021). Traffic Signal Control Using End-to-End Off-Policy Deep Reinforcement Learning. doi: https://doi.org/10.1109/TITS.2021.3067057.

[376] Bouktif, S. Cheniki, A. Ouni, A. (2021). Traffic Signal Control Using Hybrid Action Space Deep Reinforcement Learning. doi: https://doi.org/10.3390/s21072302.

[377] Kővári, B. Szőke, L. Bécsi, T. Aradi, S. Gáspár, P. (2021). Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission. doi: https://doi.org/10.3390/su132011254.

[378] Wang, M. Wu, L. Li, J. He, L. (2021). Traffic Signal Control With Reinforcement Learning Based on Region-Aware Cooperative Strategy. doi: https://doi.org/10.1109/TITS.2021.3062072.

[379] Mao, F. Li, Z. Li, L. (2022). A Comparison of Deep Reinforcement Learning Models for Isolated Traffic Signal Control. doi: https://doi.org/10.1109/MITS.2022.3144797.

[380] Korecki, M. (2022). Adaptability and sustainability of machine learning approaches to traffic signal control. doi: https://doi.org/10.1038/s41598-022-21125-3.

[381] Mohamad, S. Shabestary, A. Abdulhai, B. (2022). Adaptive Traffic Signal Control With Deep Reinforcement Learning and High Dimensional Sensory Inputs: Case Study and Comprehensive Sensitivity Analyses. doi: https://doi.org/10.1109/TITS.2022.3179893.

[382] Ibrokhimov, B. Kim, Y, J. Kang, S. (2022). Biased Pressure: Cyclic Reinforcement Learning Model for Intelligent Traffic Signal Control. doi: https://doi.org/10.3390/s22072818.

[383] Balint, K. Tamas, T. Tamas, B. (2022). Deep Reinforcement Learning based approach for Traffic Signal Control. doi: https://doi.org/10.1016/j.trpro.2022.02.035.

[384] Bouktif, S. Cheniki, A. Ouni, A. El-Sayed, H. (2022). Deep reinforcement learning for traffic signal control with consistent state and reward design approach. doi: https://doi.org/10.1016/j.knosys.2023.110440.

[385] Wu, Q. Wu, J. Shen, J. Du, B. Telikani, A. Fahmideh, M. Liang, C. (2022). Distributed agent-based deep reinforcement learning for large scale traffic signal control. doi: https://doi.org/10.1016/j.knosys.2022.108304.

[386] Zhang, L. Wu, Q. Shen, J. Lü, L. Du, B. Wu, J. (2022). Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control. Available at: https://proceedings.mlr.press/v162/zhang22ah/zhang22ah.pdf (Accessed 10 Mar. 2024).

[387] Wang, C. Xu, Y. Zhang, J. Ran, B. (2022). Integrated Traffic Control for Freeway Recurrent Bottleneck Based on Deep Reinforcement Learning. doi: https://doi.org/10.1109/TITS.2022.3141730.

[388] Zhao, W. Ye, Y. Ding, J. Wang, T. Wei, T. Chen, M. (2022). IPDALight: Intensity- and phase duration-aware traffic signal control based on Reinforcement Learning. doi: https://doi.org/10.1016/j.sysarc.2021.102374.

[389] Fang, Z. Zhang, F. Wang, T. Lian, X. Chen M. (2022). Monitor Light: Reinforcement Learning-based Traffic Signal Control Using Mixed Pressure Monitoring. doi: https://doi.org/10.1145/3511808.3557400.

[390] Jiang, Q. Qin, M. Shi, S. Sun, W. Zheng, B. (2022). Multi-Agent Reinforcement Learning for Traffic Signal Control through Universal Communication Method. doi: https://doi.org/10.48550/arXiv.2204.12190.

[391] Zeynivand, A. Javadpour, A. Bolouki, S. Sangaiah, A, K. Ja'fari, F. Pinto, P. Zhang, W. (2022). Traffic flow control using multi-agent reinforcement learning. doi: https://doi.org/10.1016/j.jnca.2022.103497.

[392] Du, Y. ShangGuan, W. Chai, L. (2022). Traffic signal control in mixed traffic environment based on advance decision and reinforcement learning. doi: https://doi.org/10.1093/tse/tdac027.

[393] Kodama, N. Harada, T. Miyazaki, K. (2022). Traffic Signal Control System Using Deep Reinforcement Learning With Emphasis on Reinforcing Successful Experiences. doi: https://doi.org/10.1109/ACCESS.2022.3225431.

[394] Li, X. Li, J. Shi, H. (2023). A multi-agent reinforcement learning method with curriculum transfer for large-scale dynamic traffic signal control. doi: https://doi.org/10.1007/s10489-023-04652-y.

[395] Guzmán, J, A. Pizarro, G. Núñez, F. (2023). A Reinforcement Learning-Based Distributed Control Scheme for Cooperative Intersection Traffic Control. doi: https://doi.org/10.1109/ACCESS.2023.3283218.

[396] Wu, C. Kim, I. Ma, Z. (2023). Deep Reinforcement Learning Based Traffic Signal Control: A Comparative Analysis. doi: https://doi.org/10.1016/j.procs.2023.03.036.

[397] Ducrocq, R. Farhi, N. (2023). Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection. doi: https://doi.org/10.1007/s13177-023-00346-4.

[398] Yang, S. (2023). Hierarchical graph multi-agent reinforcement learning for traffic signal control. doi: https://doi.org/10.1016/j.ins.2023.03.087.

[399] Korecki, M. Dailisan, D. Helbing, D. (2023). How Well Do Reinforcement Learning Approaches Cope With Disruptions? The Case of Traffic Signal Control. doi: https://doi.org/10.1109/ACCESS.2023.3266644.

[400] Yazdani, M. Sarvi, M. Bagloee, S, A. Nassir, N. Price, J. Parineh, H. (2023). Intelligent vehicle pedestrian light (IVPL): A deep reinforcement learning approach for traffic signal control. doi: https://doi.org/10.1016/j.trc.2022.103991.

[401] Lee, H. Han, Y. Kim, Y. (2023). Reinforcement learning for traffic signal control: Incorporating a virtual mesoscopic model for depicting oversaturated traffic conditions. doi: https://doi.org/10.1016/j.engappai.2023.107005.

[402] Du, W. Ye, J. Gu, J. Li, J. Wei, H. Wang, G. (2023). SafeLight: A Reinforcement Learning Method toward Collision-Free Traffic Signal Control. doi: https://doi.org/10.1609/aaai.v37i12.26729.

[403] Du, T. Wang, B. Hu, L. (2023). Single Intersection Traffic Light Control by Multi-agent Reinforcement Learning. doi: https://doi.org/10.1088/1742-6596/2449/1/012031.

[404] Liu, J. Qin, S. Su, M. Luo, Y. Zhang, S. Wang, Y. Yang, S. (2023). Traffic signal control using reinforcement learning based on the teacher-student framework. doi: https://doi.org/10.1016/j.eswa.2023.120458.

[405] Pytorch. (2017). Reinforcement Learning (DQN) Tutorial. Available at: https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html. (Accessed 21 Apr. 2025).

Appendix A: Reinforcement Learning Traffic Control Systems

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Genetic Reinforcement Learning for Cooperative Traffic Signal Control[250]	1994	Genetic Algorithm	Switch	Approaching Vehicles, Neighbouring Intersection Signals	Approaching Vehicles
Design of a traffic junction controller using classifier system and fuzzy logic[238]	1999	Learning classifier system	Phase and Duration	Queue Length (0-3 scale)	Queue Length
Distributed Learning Control of Traffic Signals[260]	2000	Learning classifier system	Phase	Queue Length (0-3 scale), Neighbouring Intersection Signals	Queue Length
Multi-agent reinforcement learning for traffic light control[261]	2000	Reinforcement Learning	Phase	Waiting Times	Waiting Times
Reinforcement Learning for True Adaptive Traffic Signal Control[262]	2003	Q-learning	Switch	Queue Length, Elapsed Phase Time	Delay
Simulation and optimisation of traffic in a city[241]	2004	Reinforcement Learning	Phase	Vehicle Positions (Discretised)	Waiting Times
Dealing with non-stationary environments using context detection[263]	2006	Q-learning	Plans (priority to an exit or each splits)	Queue Length	Queue Length
Reinforcement learning based control of traffic lights in non-stationary environments[264]	2006	Reinforcement Learning	Duration	Queue Length, Vehicle Types	Delay
Sensor Grid for Real-Time Traffic Management[265]	2007	Q-learning	Phase and Duration	Queue Lengths, Time Between Arrivals	Delay

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Traffic light scheduling using policy-gradient reinforcement learning[239]	2007	Policy-Gradient	Phase	Queue Length (limited to 8)	Approaching Vehicles
A Collaborative Reinforcement Learning Approach to Urban Traffic Control Optimisation[266]	2008	Q-learning	Phase	Approaching Vehicles	Vehicles Passing, Queue Length
Adaptive and Coordinated Traffic Signal Control Based on Q-Learning and MULTIBAND Model[267]	2008	Q-learning	Phase	Delay	Delay
Distributed w-learning: Multi-policy optimisation in self-organizing systems[268]	2009	Q-Learning	Phase	Queue Length	Queue Length
Urban traffic signal learning control using fuzzy actor-critic methods[269]	2009	Actor-Critic Method	Switch	Queue Length	Waiting Time
Using reinforcement learning for multi-policy optimisation in decentralised autonomic systems–an experimental evaluation[242]	2009	Q-learning	Phase	Queue Length, Emergency Vehicles	Queue Length, Emergency Vehicles
A comparative study of urban traffic signal control with reinforcement learning and adaptive dynamic programming[270]	2010	ADP	Phase	Queue Length, Elapsed Phase Time	Queue Length, Elapsed Phase Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Arterial traffic control using reinforcement learning agents and information from adjacent intersections in the state and reward structure[271]	2010	Q-learning	Phase	Vehicles Approaching, Journey Time	Approaching Vehicles, Vehicles Served, Delay, Downstream Conditions
Improving space representation in multiagent learning via tile coding[272]	2010	Q-learning	Plans (priority to an exit or each splits)	Queue Length	Queue Length
Learning in groups of traffic signals[273]	2010	Q-learning	Phase	Queue Length	Queue Length
Reinforcement Learning With Function Approximation for Traffic Signal Control[274]	2010	Q-learning	Phase	Queue Length, Red Indicator	Queue Length, Red Time
Reinforcement learning-based multi-agent system for network traffic signal control[275]	2010	Q-learning	Phase	Delay	Delay
An automated signalised junction controller that learns strategies from a human expert[276]	2011	TD	Phase	Vehicle Speed, Distance to Junction	Queue Length, Equitability of Queues
Dealing with continuous-state reinforcement learning for intelligent control of traffic signals[277]	2011	Q-learning	Phase	Queue Length	Approaching Vehicles

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Neural network-based online traffic signal controller design with reinforcement training[278]	2011	Reinforcement Learning	Switch	Queue Length	Queue Length
Reinforcement Learning with Average Cost for Adaptive Control of Traffic Lights at Intersections[249]	2011	Multiple	Phase	Queue Length, Red Indicator	Queue Length, Waiting Time
Traffic light control in non-stationary environments based on multi-agent q-learning[279]	2011	Q-learning	Plans	Queue Length	Queue Length
Enhanced Multiagent Multi-Objective Reinforcement Learning for Urban Traffic Light Control[280]	2012	Reinforcement Learning	Phase	Vehicle Positions, Vehicle Destinations	Queue Length
Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto[281]	2012	Q-learning	Phase	Queue Length, Current Phase, Elapsed Phase Time	Delay
Multi-objective traffic light control system based on Bayesian probability interpretation[282]	2012	Reinforcement Learning	Phase	Vehicle Positions, Vehicle Destinations	Queue Length

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Threshold Tuning Using Stochastic Optimisation for Graded Signal Control[283]	2012	Multiple RL	Phase	Queue Length, Red Indicator	Queue Length, Waiting Time
Cell transmission model-based multiagent q-learning for network-scale signal control with transit priority[243]	2013	Q-learning	Switch	Vehicles Approaching (separated by low and high priority vehicles)	Red Light Delay
Holonic multi-agent system for traffic signals control[221]	2013	Q-learning	Plan (100 seconds)	Queue Length	Queue Length
Learning Coordinated Traffic Light Control[284]	2013	Q-learning	Switch (2 seconds)	Elapsed Phase Time, Last Phase Length, Waiting Times	Waiting Time
Online coordination of signals for heterogeneous traffic using stop line detection[285]	2013	Actor Critic	Plan	Latest Green Times	Vehicles Passing
Optimisation of urban multi-intersection traffic flow via q-learning[286]	2013	Q-learning	Switch (1/5 seconds)	Queue Length	Idle Green Time
The Study of Reinforcement Learning for Traffic Self-Adaptive Control under Multiagent Markov Game Environment[287]	2013	Q-learning	Duration	Neighbouring Intersection Signals	Vehicles Passed, Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Urban Traffic Control Using Adjusted Reinforcement Learning in a Multi-agent System[288]	2013	Q-learning	Plan (1 Cycle)	Vehicles Approaching	Approaching Vehicles
Agent-based optimisation for multiple signalised intersections using q-learning International Journal of Simulation[289]	2014	Q-learning	Switch	Queue Length	Queue Length, Idle Green Time
Combining multiple correlated reward and shaping signals by measuring confidence[290]	2014	Q-learning	Switch (2 seconds)	Elapsed Phase Time, Last Phase Length, Queue Length	Vehicles Served, Delay
Cooperative multi-agent traffic signal control system using fast gradient-descent function approximation for v2i networks[291]	2014	Q-learning	Phase	Queue Length, Waiting Time	Queue Length, Waiting Time
Hierarchical control of traffic signals using Q-learning with tile coding[292]	2014	Q-learning	Plan (every cycle)	Queue Length	Queue Length
Road artery traffic light optimisation with use of the reinforcement learning[293]	2014	Q learning	Phase (4/10 seconds)	Queue Length, Phase, Elapsed Phase Time	Queue Length
Traffic signal control using reinforcement learning[294]	2014	Q-learning	Duration	Images	Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
A junction-tree based learning algorithm to optimise network wide traffic control: A coordinated multi-agent framework[295]	2015	Q-learning	Phase	Queue Length	Queue length
A Multiagent Auction-based Approach for Modelling of Signalised Intersections[296]	2015	Q-learning	Phase	Queue Length	Vehicles Served, Delay
Adaptive Group-Based Signal Control Using Reinforcement Learning with Eligibility Traces[245]	2015	Q-learning	Switch (1/2/3/4 seconds)	Distances Between Vehicles, Occupancy, Elapsed Green Time, Phase	Delay
Adaptive traffic signal control for multi-intersection based on microscopic model[297]	2015	ADP	Switch	Queue Length, Phase	Waiting Time
CVLight: Decentralized Learning for Adaptive Traffic Signal Control with Connected Vehicles[298]	2015	DQN	Phase (7 seconds)	Phase, Queue Length, Vehicles Passed	Queue length, Vehicles Served
Distributed q-learning controller for a multi-intersection traffic network[299]	2015	Q-learning	Duration (10/20/.../80 seconds)	Approaching Vehicles	Delay
Distributed urban traffic control based on locally observable cell occupancies[300]	2015	Q-learning	Phase	Vehicle Positions	Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Swarm reinforcement learning for traffic signal control based on cooperative multi-agent framework[301]	2015	Q-learning	Plan	Waiting Time	Queue length, Vehicles Served
Towards reinforcement learning for holonic multi-agent systems Intelligent Data Analysis[302]	2015	Q-learning	Plan (every cycle)	Queue Length	Queue Length
Coordinated deep reinforcement learners for traffic light control[303]	2016	DQN	Phase	Vehicle Positions (Discretised)	Jams, Crashes, Emergency Brakes, Signal Change, Delay
Deep reinforcement learning for coordination in traffic light control[304]	2016	DDPG	Switch	Queue Length, Phase	Queue Length
Intelligent traffic light control using collaborative q-learning algorithms[305]	2016	Q-Learning	Phase	Queue Length	Queue Length, Vehicles Served, Pedestrian Queue
Intelligent traffic signal duration adaptation using q-learning with an evolving state space[306]	2016	Q-Learning	Plan (240 seconds)	Queue Lengths	Queue Length
Large-scale multi-agent reinforcement learning using image-based state representation[252]	2016	DQN	Phase	Images	Queue length
Traffic signal timing via deep reinforcement learning[253]	2016	DQN	Switch	Queue Length	Delay, Queue Length

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Using a Deep Reinforcement Learning Agent for Traffic Signal Control[229]	2016	DQN	Phase	DTSE (Position, Velocity, Phase)	Delay
A multi-objective agent-based approach for road traffic controls: application for adaptive traffic signal systems[307]	2017	SARSA	Duration	Approaching Vehicles, Traffic Flow	Queue Length, Vehicles Passing
Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network[308]	2017	Q-learning	Phase	DTSE (Position, Velocity)	Waiting Time
Distributed cooperative reinforcement learning-based traffic signal control that integrates v2x networks' dynamic clustering[309]	2017	Q-learning	Phase	Queue Length, Waiting Time	Queue Length, Waiting Time
Exploring cooperative multi-agent reinforcement learning algorithm (cmrla) for intelligent traffic signal control[310]	2017	Q-learning	Switch	Queue Length, Vehicles Approaching	Queue Length, Vehicles Served
Intelligent traffic light control using distributed multi-agent q learning[311]	2017	Q-learning	Phase	Queue Length (Vehicles and Pedestrians)	Queue Length

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Multi-agent immune networks to control interrupted flow at signalised intersections[312]	2017	Reinforcement Learning	Plan	Queue Length, Queue Delay	Vehicles Served
Simulation of intelligent traffic control for autonomous vehicles[313]	2017	Q-learning	Phase	Phase, Emergency Vehicles	Wait Times
Traffic light control using deep policy-gradient and value-function-based reinforcement learning[314]	2017	DQN	Phase	Images	Delay
Traffic lights dynamic timing algorithm based on reinforcement learning[246]	2017	Q-learning	Phase	Traffic Density	Travel Time
A regional traffic signal control strategy with deep reinforcement learning[315]	2018	DQN	Switch	Queue Lengths (Local and Neighbouring Junctions)	Queue Length
A Self-Adaptive Collaborative Multi-Agent based Traffic Signal Timing System[316]	2018	Q-learning	Phase	Flow Rates	Vehicles Served
Adaptive Traffic Signal Control with Deep Recurrent Q-learning[233]	2018	DQN	Switch	Vehicles Approaching, Speed	Queue Length
Benchmarks for reinforcement learning in mixed-autonomy traffic[317]	2018	DQN	Switch	Position, Speed	Average Velocity, Crashes

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Co-adaptive reinforcement learning in microscopic traffic systems[318]	2018	Q-learning	Phase	Phase, Elapsed Time, Queue Length	Queue Length
Continuous residual reinforcement learning for traffic signal control optimisation[319]	2018	Q-learning	Duration (20/30/.../90 seconds)	Vehicles Approaching	Approaching Vehicles
Deep Q Learning with LSTM for Traffic Light Control[219]	2018	DQN	Switch	Vehicles Approaching, Velocity	Approaching Vehicles
Deep recurrent q-learning method for area traffic coordination control[320]	2018	DQN	Phase (10 seconds)	DTSE (Position, Speed, Acceleration)	Delay
Deep reinforcement learning for traffic light control in vehicular networks[223]	2018	Q-learning	Plan (can alter a phase by 5 seconds)	DTSE (Position, Speed)	Waiting Time
Fuzzy q-learning-based multi-agent system for intelligent traffic control by a game theory approach[321]	2018	Q-learning	Duration	Green Duration, Neighbour Green Duration, Queue Length	Queue Length, Delay
Hierarchical multi-agent control of traffic lights based on collective learning[322]	2018	SARSA	Switch	Elapsed Green Time, Time Between Vehicles, Flow Rates	Delay
IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control[254]	2018	DQN	Switch	Queue Length, Approaching Vehicles, Waiting Time, Image, Current Phase, Next Phase	Queue Length, Delay, Waiting Time, Signal Changes, Vehicles Served, Travel Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Learning-based traffic signal control algorithms with neighbourhood information sharing: An application for sustainable mobility[244]	2018	TD	Phase	Queue Density, Phase with Largest Queue, Neighbour with Largest Queue	Queue Length, Delay
Minimizing energy consumption from connected signalised intersections by reinforcement learning[323]	2018	Q-learning	Switch	Queue Length	Delay, Energy Consumption, Queue Length
Traffic signal optimisation through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran[324]	2018	Multiple RL	Duration (5/10/.../70 seconds)	Current Phase, Time of Day, Queue Length (Up and Downstream)	Queue Length
Value-based deep reinforcement learning for adaptive isolated intersection signal control[325]	2018	DQN	Phase	Current Phase, Elapsed Green Time, Left Turn Queue, Approaching Vehicles	Delay
A deep reinforcement learning network for traffic light cycle control[326]	2019	DQN	Plan (phases can be changed by 5 seconds)	DTSE (Position and Speed)	Waiting Time
A multi-objective agent-based control approach with application in intelligent traffic signal system[327]	2019	Q-learning	Plan (phases altered by -10 to 10 seconds (discrete))	Approaching Vehicles, Traffic Flow	Queue Length, Vehicles Served
A new multi-agent reinforcement learning method based on evolving dynamic correlation matrix[328]	2019	Q-learning	Phase	Approaching Vehicles, Vehicles Served	Approaching Vehicles

Paper	Year	Algorithm	Action Space	State Space	Reward Space
A Reinforcement Learning Approach for Intelligent Traffic Signal Control at Urban Intersections[228]	2019	DQN	Phase	Queue Length	Queue Length
An Adaptive Control Method for Arterial Signal Coordination Based on Deep Reinforcement Learning[257]	2019	DQN	Phase	Queue Length, Vehicle Speeds	Queue Length, Vehicles Served
An Open-Source Framework for Adaptive Traffic Signal Control[215]	2019	DDPG	Duration (continuous)	Current Phase, Traffic Density, Queue Length	Delay
Colight: Learning network-level cooperation for traffic signal control[329]	2019	DQN	Phase	Approaching Vehicles, Current Phase	Queue Length
Cooperative deep q-learning with value transfer for multi-intersection signal control[330]	2019	DQN	Phase	DTSE (Position and Speed)	Queue Length
Cooperative multi-intersection traffic signal control based on deep reinforcement learning[331]	2019	Actor Critic	Switch	DTSE (Position)	Approaching Vehicles
Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm[332]	2019	A3C	Phase	Image	Delay

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Decentralised network level adaptive signal control by multi-agent deep reinforcement learning,[333]	2019	DQN	Phase	DTSE (Position), Current Phase	Waiting Time
Deep Reinforcement Learning for Adaptive Traffic Signal Control[334]	2019	DQN	Switch	Average Travel Time, Queue Length	Queue Length, Delay, Vehicles Served, Residual Queue
Deep reinforcement learning-based traffic signal control using high-resolution event-based data[335]	2019	DDQN	Phase	Variation on DTSE for loop detectors (Position)	Vehicles Served, Waiting Time
Developing adaptive traffic signal control by actor-critic and direct exploration methods[336]	2019	Q-learning	Duration (10/20/.../90 seconds)	Approaching Vehicles	Queue Length
Dueling Double Deep Q-Network for Adaptive Traffic Signal Control with Low Exhaust Emissions in A Single Intersection[256]	2019	DDQN	Switch	DTSE (Position, Speed, Acceleration), Current Phase	Emissions
ERL: Edge Based Reinforcement Learning for Optimised Urban Traffic Light Control[337]	2019	DQN	Plan (changes a threshold metric that controls the frequency of light changes)	Traffic Flow, Average Vehicle Speed, Current phase	Speed, Approaching Vehicles

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Hierarchical regional control for traffic grid signal optimisation[338]	2019	DQN	Phase	Vehicle Positions, Vehicle Speeds, Current phase, Elapsed Phase Duration	Queue Length
Introduction to coordinated deep agents for traffic signal[339]	2019	DQN	Phase	DTSE (Position, Speed, Phase (local and neighbours))	Waiting Time, Approaching Vehicles
Learning phase competition for traffic signal control[340]	2019	DQN	Phase	Queue Length, Current Phase	Queue Length
Multi-agent deep reinforcement learning for large-scale traffic signal control[341]	2019	A2C	Phase	Approaching Vehicles, Delay	Queue Length, Delay
Multiagent reinforcement learning applied to traffic light signal control[342]	2019	Q-learning	Phase	Time of day, Queue Length, Waiting Time	Queue Lengths, Waiting Time
Multimodal intelligent deep (mind) traffic signal controller[343]	2019	DQN	Phase	DTSE (Position and Speed)	Delay
Presslight: Learning max pressure control to coordinate traffic signals in arterial network[344]	2019	DQN	Phase	Vehicles Approaching, Vehicles Passed, Current Phase	Vehicles Approaching, Vehicle Served
Rainbow Deep Reinforcement Learning Agent for Improved Solution of the Traffic Congestion[232]	2019	DQN	Phase	DTSE (Position, Speed, Waiting Time)	Phase change, Emergency Stop, Speed Delay, Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Reinforcement learning agent under partial observability for traffic light control in presence of gridlocks[345]	2019	DQN	Phase	Occupancy, Current Phase	Occupancy, Vehicle Served
Reinforcement learning with explainability for traffic signal control[346]	2019	Policy-Gradient	Phase	Vehicles Approaching	Vehicles Served
Time critic policy gradient methods for traffic signal control in complex and congested scenarios[347]	2019	Policy-Gradient	Phase	Vehicles Approaching	Vehicles Served
Traffic Signal Control with Deep Reinforcement learning[230]	2019	DQN	Phase	DTSE (Position, Speed)	Delay
Training Reinforcement Learning Agent for Traffic Signal Control under Different Traffic Conditions[217]	2019	DQN	Switch	DTSE (Position, Speed), Phase	Vehicles Served, Queue Length, Phase Change, Waiting Time
Cooperative deep reinforcement learning for large-scale traffic grid signal control[348]	2020	DDPG	Phase	Queue Length	Queue Length, Moving Vehicles
Cooperative traffic signal control with traffic flow prediction in multi-intersection[349]	2020	DQN	Phase	Approaching Vehicles	Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Fuzzy Inference Enabled Deep Reinforcement Learning-Based Traffic Light Control for Intelligent Transportation System[350]	2020	DQN	Switch	DTSE (Position, Speed)	Waiting Time, Delay, Queue Length, Speed
Large-Scale Traffic Signal Control Using a Novel Multiagent Reinforcement Learning[351]	2020	Q-Learning	Phase (4 seconds)	Lead Car Position, Delay, Queue Length	Waiting Time, Queue Length
MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control[352]	2020	DQN	Phase	Approaching Vehicles, Queue Length	Queue Length
Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control[353]	2020	A2C	Phase	Wait Times, Approaching Vehicles	Wait Times, Queue Length
Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks[354]	2020	DDPG	Switch	DTSE (Position, Speed) + Queue Length	Queue Length, Waiting Time, Delay, Vehicles Served, Phase Changes
Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning[355]	2020	DQN	Switch	Vehicles approaching, Average speed (Local and Neighbours), Current Phases of Neighbours	Delay

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Policy analysis of adaptive traffic signal control using reinforcement learning[356]	2020	A3C	Phase (15 seconds)	Queue Length, Traffic Density	Queue Length
Reinforcement learning algorithm for non-stationary environments[357]	2020	Q-learning	Duration (20/25/.../70 seconds)	Queue Length	Queue Length
Reinforcement learning for joint control of traffic signals in a transportation network[358]	2020	DQN	Phase	Images	Approaching Vehicles
STMARL: A Spatio-Temporal Multi-Agent Reinforcement Learning Approach for Cooperative Traffic Light Control[359]	2020	DQN	Phase and Switch	Queue Length, Speed, Approaching Vehicles	Queue Length
Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control[360]	2020	DQN	Phase	Current Phase, Approaching Vehicles, Vehicles Served	Queue Length, Vehicles Served
Traffic signal control for smart cities using reinforcement learning[361]	2020	Q-Learning	Phase	Vehicles Approaching	Queue Length, Vehicles Served
Using Reinforcement Learning With Partial Vehicle Detection for Intelligent Traffic Signal Control[41]	2020	DQN	Switch	Vehicles Approaching, Lead Car Position, Amber Phase Indicator, Current Time, Current Phase	Travel Time, Delay

Paper	Year	Algorithm	Action Space	State Space	Reward Space
A Deep Reinforcement Learning Approach to Traffic Signal Control With Temporal Traffic Pattern Mining[362]	2021	Actor Critic	Switch (1 seconds)	Images	Queue Length
A Novel Reinforcement Learning-Based Cooperative Traffic Signal System Through Max-Pressure Control[363]	2021	DQN	Phase	Current Phase, Vehicles Approaching, Average Speed, Average Acceleration	Approaching Vehicles, Vehicles Served
A Study on Deep Reinforcement Learning Based Traffic Signal Control for Mitigating Traffic Congestion[364]	2021	DQN	Phase	Flow	Approaching Vehicles, Vehicles Served
Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning[365]	2021	DQN	Green Wave / Normal Operation	Queue Length, Current Phase, Most Congested Intersections	Approaching Vehicles, Vehicles Served
An integrated MPC and deep reinforcement learning approach to trams-priority active signal control[366]	2021	PPO	Phase	DTSE (Position and Speed)	Queue Length, Tram Stops
GraphLight: Graph-based Reinforcement Learning for Traffic Signal Control[367]	2021	A2C	Phase	Vehicle Positions, Speeds, Wait Times	Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
”Hierarchical traffic signal optimisation using reinforcement learning and traffic prediction with long-short term memory[368]	2021	DQN	Plan (once per cycle, fixed phase order)	Queue Length, Delay, Last Action	Delay
” Hierarchically and Cooperatively Learning Traffic Signal Control[369]	2021	Actor Critic, DQN, PPO	Switch	Current Phase, Next Phase, Queue Length	Queue Length, Waiting Time, Delay
IG-RL: Inductive Graph Reinforcement Learning for Massive-Scale Traffic Signal Control[370]	2021	DQN	Switch	Phase, Approaching Vehicles	Queue Length
IHG-MA: Inductive heterogeneous graph multi-agent reinforcement learning for multi-intersection traffic signal control[371]	2021	Actor Critic	Phase and Duration (Continuous)	Current Phase, Elapsed Phase Duration, Queue Length, Approaching Vehicles, Average Vehicle Speed, Wait Time, DTSE (Position, Speed, Delay)	Queue Length, Waiting Time
Network-wide traffic signal control optimisation using a multi-agent deep reinforcement learning[372]	2021	DDPG	Two Options: Green Wave or Normal Operation	Approaching Vehicles, Queue Length, Current Phase, Most Congested Intersections	Approaching Vehicles, Vehicles Served
Towards Real-World Deployment of Reinforcement Learning for Traffic Signal Control[251]	2021	DQN	Phase	Queue Length, Lead Vehicle Positions, Speed, Wait Time (Vehicles and Pedestrians), Current Phase, Elapsed Phase Length	Queue Length, Waiting Time (Vehicles and Pedestrians)

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Traffic Flow Management of Autonomous Vehicles Using Deep Reinforcement Learning and Smart Rerouting[373]	2021	DQN	Phase	DTSE (Position and Speed)	Waiting Time
Traffic Signal Control Under Mixed Traffic With Connected and Automated Vehicles: A Transfer-Based Deep Reinforcement Learning Approach[374]	2021	DQN	Phase	Approaching Vehicles	Waiting Time
Traffic Signal Control Using End-to-End Off-Policy Deep Reinforcement Learning[375]	2021	DQN	Duration	Images	Waiting Time
Traffic Signal Control Using Hybrid Action Space Deep Reinforcement Learning[376]	2021	DQN	Phase and Duration (Continuous)	Queue Length, Current Phase	Queue Length
Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission[377]	2021	DQN	Phase	Approaching Vehicles	Approaching Vehicles, Distribution of Vehicles
Traffic Signal Control With Reinforcement Learning Based on Region-Aware Cooperative Strategy[378]	2021	A2C	Phase	Waiting Time, Queue Length	Waiting Time, Queue Length

Paper	Year	Algorithm	Action Space	State Space	Reward Space
A Comparison of Deep Reinforcement Learning Models for Isolated Traffic Signal Control[379]	2022	DQN/DDQN/PG/A2C/PPO/SAC	Phase or Switch or Duration	Approaching Vehicles, Queue Length, Speed, Current Phase	Queue Length, Vehicle Speed, Phase Changes, Approaching Vehicles
A Reinforcement Learning Based Traffic Control Strategy in a Macroscopic Fundamental Diagram Region[225]	2022	Q-learning	Phase and Switch	Queue Length, Vehicle Served	Delay, Approaching Vehicles
Adaptability and sustainability of machine learning approaches to traffic signal control[380]	2022	DQN	Phase	Approaching Vehicles	Vehicles Approaching, Vehicles Passed
Adaptive Traffic Signal Control With Deep Reinforcement Learning and High Dimensional Sensory Inputs: Case Study and Comprehensive Sensitivity Analyses[381]	2022	DQN	Phase	DTSE (Position and Speed)	Delay
Biased Pressure: Cyclic Reinforcement Learning Model for Intelligent Traffic Signal Control[382]	2022	A2C	Phase	Current Phase, Elapsed Phase Length, Queue Length, Vehicles Served	Vehicles Approaching, Vehicles Passed
Deep Reinforcement Learning based approach for Traffic Signal Control[383]	2022	PG	Phase	Approaching Vehicles	Distribution of Vehicles

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Deep reinforcement learning for traffic signal control with consistent state and reward design approach[384]	2022	DDQN	Phase	Approaching Vehicles, Queue Length, Wait Time	Approaching Vehicles, Queue Length, Wait Time
Distributed agent-based deep reinforcement learning for large scale traffic signal control[385]	2022	A2C	Phase	Queue Length	Waiting Time
Expression might be enough: representing pressure and demand for reinforcement learning based traffic signal control[386]	2022	DQN	Phase	Current Phase, Approaching Vehicles, Vehicles Served	Approaching Vehicles, Vehicles Served
Human-centric multimodal deep (HMD) traffic signal control[235]	2022	D3QN	Switch and Phase	Traffic Density, Pedestrians, Waiting Time, Current Phase	Waiting Time
Integrated Traffic Control for Freeway Recurrent Bottleneck Based on Deep Reinforcement Learning[387]	2022	DDPG/TD3	Speed Limits and Ramp Outflow	DTSE (large segments) (Inflow, Outflow, Average Speed, Waiting Vehicles, Traffic Density)	Average Speed, Queue Length
IPDALight: Intensity- and phase duration-aware traffic signal control based on Reinforcement Learning[388]	2022	DQN	Phase	Vehicle Speeds, Vehicle Positions	Vehicle Speeds and Positions
Monitor Light: Reinforcement Learning-based Traffic Signal Control Using Mixed Pressure Monitoring[389]	2022	DQN	Phase	Queue Length, Vehicles Served	Vehicles Approaching, Vehicles Served

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Multi-Agent Reinforcement Learning for Traffic Signal Control through Universal Communication Method[390]	2022	DQN	Phase	Vehicles Approaching, Current Phase	Vehicles Approaching
Real-Time Adaptive Traffic Signal Control in a Connected and Automated Vehicle Environment: Optimisation of Signal Planning with Reinforcement Learning under Vehicle Speed Guidance[29]	2022	A2C	Phase	Queue Length, Current Phase, Elapsed Phase Length, Time Since Phases	Queue Length
Traffic flow control using multi-agent reinforcement learning[391]	2022	Q-learning	Duration	Vehicles Approaching	Vehicles Approaching
Traffic signal control in mixed traffic environment based on advance decision and reinforcement learning[392]	2022	3DQN	Phase	DTSE (Position, Speed)	queue length, vehicle speed
Traffic Signal Control System Using Deep Reinforcement Learning With Emphasis on Reinforcing Successful Experiences[393]	2022	DQN	Phase	Current Phase, Queue Length, Average Speed	Waiting Time

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Using ontology to guide reinforcement learning agents in unseen situations[248]	2022	DQN	Phase	Current Phase, Elapsed Phase Length, Queue Length, Approaching Vehicles, Lead Vehicle (Type, Position, Waiting Time)	Waiting Time
A multi-agent reinforcement learning method with curriculum transfer for large-scale dynamic traffic signal control[394]	2023	DQN/DDPG	Phase	Queue Length	queue length
A Reinforcement Learning-Based Distributed Control Scheme for Cooperative Intersection Traffic Control[395]	2023	PPO	Phase	Vehicle Density, Queue Length, Vehicle Positions, Mean Speed	change in queue length
Causal inference multi-agent reinforcement learning for traffic signal control[227]	2023	Actor Critic	Phase and Duration	Waiting Time, Approaching Vehicles	Delay
Deep Reinforcement Learning Based Traffic Signal Control: A Comparative Analysis[396]	2023	DQN	Switch	Phase, Approaching Vehicles/DTSE (Position and Speed)	queue length/waiting time/throughput/vehicles approaching
Deep Reinforcement Learning for the Co-Optimisation of Vehicular Flow Direction Design and Signal Control Policy for a Road Network[236]	2023	DQN	Phase	Lead Car Position and Route, Approaching Vehicles	Queue length, Vehicles passed

Paper	Year	Algorithm	Action Space	State Space	Reward Space
Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection[397]	2023	DQN	Phase	DTSE (Position, Speed, Phase)	Delay
Hierarchical graph multi-agent reinforcement learning for traffic signal control[398]	2023	Actor Critic	Phase and Duration	Previous Action, Queue Length, Approaching Vehicles, Average Vehicle Speed, Wait Time, DTSE (Position, Speed, Delay)	Queue Length, Waiting Time
How Well Do Reinforcement Learning Approaches Cope With Disruptions? The Case of Traffic Signal Control[399]	2023	Q-learning	Phase (10 seconds)	Vehicles Approaching, Vehicles Served, Phase	Vehicles Approaching, Vehicles Passed
Intelligent vehicle pedestrian light (IVPL): A deep reinforcement learning approach for traffic signal control[400]	2023	DDQN	Duration	DTSE (Position, Speed), Number of Pedestrians, Current Phase	Delay (Vehicles and Pedestrians)
Multi-Agent Reinforcement Learning for Traffic Signal Control: A Cooperative Approach[237]	2023	Policy Gradient	Phase	Queue Length	Queue Length
Reinforcement learning for traffic signal control: Incorporating a virtual mesoscopic model for depicting oversaturated traffic conditions[401]	2023	DDPG	Phase	Approaching Vehicles	Delay

Paper	Year	Algorithm	Action Space	State Space	Reward Space
SafeLight: A Reinforcement Learning Method toward Collision-Free Traffic Signal Control[402]	2023	3DQN	Phase or Duration	Queue Length	Waiting Time
Single Intersection Traffic Light Control by Multi-agent Reinforcement Learning[403]	2023	DQN	Phase	DTSE (Position)	Queue Length
Traffic signal control using reinforcement learning based on the teacher-student framework[404]	2023	DDQN	Phase	Approaching Vehicles, Queue Length, Current Phase	Queue Length
Two-layer coordinated reinforcement learning for traffic signal control in traffic network[247]	2023	A2C/DQN	Phase	Queue Length, Current Phase, Approaching Vehicles, Emissions	Delay, Emissions

Appendix B: ANOVA Test Statistics

		Experiment 1	Experiment 2		
			TPR	EPR	Interaction
55%	Average Speed	73.81	1815.70	1039.40	11.68
	Junction Speed	159.51	1079.08	192.43	9.61
	Number of Stops	212.04	1751.05	13.89	15.52
		Experiment 3	Experiment 4		
			TPR	EPR	Interaction
55%	Average Speed	1433.22	85.13	8768.70	27.17
	Waiting Time	8.69	14.06	58.64	3.91
	Number of Stops	2585.61	35550.60	3811.99	6166.18
65%	Average Speed	687.33	18.18	3650.92	12.54
	Waiting Time	49.26	3.85	180.07	5.84
	Number of Stops	303.77	982.89	113.25	174.96
75%	Average Speed	360.77	34.18	2455.49	16.05
	Waiting Time	61.34	41.00	469.32	15.62
	Number of Stops	45.40	286.06	8.99	50.04
		Experiment 5	Experiment 6		
			TPR	EPR	Interaction
55%	Average Speed	1589.39	65.32	10893.12	76.13
	Waiting Time	25.05	24.78	124.46	12.42
	Number of Stops	4402.99	18856.08	908.15	2400.52
65%	Average Speed	439.57	42.72	3273.63	11.62
	Waiting Time	36.16	150.42	528.14	17.20
	Number of Stops	411.88	1466.86	58.46	211.22
75%	Average Speed	551.51	33.19	2868.62	26.33
	Waiting Time	84.17	61.99	540.28	29.70
	Number of Stops	320.74	872.20	19.56	155.10

Table 7: ANOVA Test Statistics

Appendix C: Journal and Conference Papers Based on This Thesis

- Paine, W, R. Waterson, B. Snowdon, J. (2025). Potential Approaches for Combined Green Light Optimised Speed Advisory and Responsive Traffic Control Systems for Non-Autonomous Vehicles. Transport Reviews. (Submitted)
- Paine, W, R. Waterson, B. Snowdon, J. (2025). Applying Green Light Optimal Speed Advisory Systems to Reinforcement Learning Adaptive Traffic Control Frameworks at Varied Penetration Rates. IET Intelligent Transport Systems. (Submitted)
- Paine, W, R. Waterson, B. Snowdon, J. (2025). Applying Green Light Optimal Speed Advisory Systems to Reinforcement Learning Adaptive Traffic Control Frameworks at Varied Penetration Rates on Arterial Flows. Transportmetrica B: Transport Dynamics. (Submitted)
- Paine, W, R. Waterson, B. Snowdon, J. (2025). Unlocking the Potential of GLOSA. University Transport Study Group 2025. (Submitted)