eng

MDPI

Article

Enhancing the Robustness of Traffic Signal Control with StageLight: A Multiscale Learning Approach

Gang Su and Jidong J. Yang *

Smart Mobility and Infrastructure Laboratory, College of Engineering, University of Georgia, Athens, GA 30602, USA; gang.su@uga.edu
* Correspondence: jidong.yang@uga.edu

Abstract: The continuous evolution of artificial intelligence and cyber–physical systems has presented promising opportunities for optimizing traffic signal control in densely populated urban areas, with the aim of alleviating traffic congestion. One area that has garnered significant interest from both researchers and practitioners is the application of deep reinforcement learning (DRL) in traffic signal control. However, DRL-based algorithms often suffer from instability due to the dynamic nature of traffic flows. Discrepancies between the environments used for training and those encountered during deployment often lead to operational failures. Moreover, conventional DRL-based traffic signal control algorithms tend to reveal vulnerabilities when faced with unforeseen events, such as sensor failure. These challenges highlight the need for innovative solutions to enhance the robustness and adaptability of such systems. To address these pertinent issues, this paper introduces StageLight, a novel two-stage multiscale learning approach, which involves learning optimal timings on a coarse time scale in stage 1, while finetuning them on a finer time scale in stage 2. Our experimental results demonstrate StageLight's remarkable capability to generalize across diverse traffic conditions and its robustness to various sensor-failure scenarios.

Keywords: traffic signal control; reinforcement learning; deep Q-learning; sensor failure; robustness

Citation: Su, G.; Yang, J.J. Enhancing the Robustness of Traffic Signal Control with StageLight: A Multiscale Learning Approach. *Eng* **2024**, *5*, 104–115. https://doi.org/10.3390/eng5010007

Academic Editor: Gian Carlo Cardarilli

Received: 7 December 2023
Revised: 1 January 2024
Accepted: 4 January 2024
Published: 8 January 2024

1. Introduction

The persistence of traffic congestion in metropolitan areas remains a critical challenge, primarily attributed to the ever-expanding population and the continuous increase in vehicular traffic. This ongoing issue is compounded by restricted right-of-way and limited budgets and resources. In response to these challenges, the utilization of intelligent traffic control systems to optimize traffic signal timings has emerged as a pivotal and cost-effective strategy for transportation agencies combating congestion. The recent development of sophisticated detection technologies has revolutionized the collection of highly precise traffic data. The integration of these advanced technologies into traffic management systems holds significant promise for effectively addressing the complexities of urban traffic congestion. They provide a robust data foundation, offering transportation agencies reliable and comprehensive data sources for employing powerful, learning-based control methods [1]. Over the past decade, researchers have increasingly resorted to reinforcement learning (RL) as a promising algorithmic approach to enhance traffic signal control and alleviate congestion. RL offers a framework where an agent learns optimal behavior through extensive trial-and-error interactions with its environment [2]. In the realm of traffic control, the utilization of RL enables traffic lights to dynamically adapt to fluctuating traffic conditions through data-driven mechanisms. What sets RL-based control apart from conventional adaptive control methods is its intrinsic ability to self-learn, guided by pertinent reward signals derived from data and experiences rather than relying solely on pre-defined rules within the control loop. This distinctive characteristic of RL-based traffic signal control holds the potential to revolutionize traditional traffic management

systems by facilitating more adaptive, responsive, and context-sensitive adjustments in real time, thereby fostering improved traffic flow and congestion mitigation in urban settings. Most recent works in the traffic control domain have been largely concentrated on the application of deep reinforcement learning (DRL) algorithms with respective to novel traffic data sources acquired through modern sensing technologies. Deep Q-Net (DQN) works have been commonly used, and the main variations lie in the definition of state and reward, and agent networks. For instance, Bouktif et al. [3] introduced a DRL-based traffic signal control approach centered around the idea of striving for consistency between state and reward. Their work explored three variants of consistent dual definition for state and reward based on the queue length, number of vehicles, and waiting time. In another study, Liu and Li [4] proposed a multiagent-based algorithm that enhanced interagency communication and gauged traffic congestion in a more reasonable way. They introduced a new reward that incorporated both the waiting time and queue length, aiming for a comprehensive congestion assessment.

While infrastructure-based sensing has dominated, the emergence of connected vehicle (CV) sensing has significantly augmented traffic-monitoring capabilities. Leveraging the growing prevalence of smart vehicles equipped with communication technologies, these systems facilitate the exchange of data between vehicles and infrastructure. Through vehicle-to-infrastructure and vehicle-to-vehicle communication, connected vehicles enable more dynamic and accurate traffic monitoring. Shi et al. [5] presented a method that used position and speed information from CVs with an assumption of 100% CV penetration. Additionally, their approach employed an autoencoder to learn compressed representations of traffic states. However, the state representation could be further simplified by focusing on travel lanes only. As pointed out by the authors, the proposed method used the position and speed information of CVs only at the end of a control cycle, neglecting valuable temporal information. Similarly exploring CV technology, Kumar et al. [6] utilized the exact location of each vehicle via dedicated short-range communications (DSRC) vehicle-to-infrastructure communication and also accounted for various vehicle types' impacts on traffic delays. They assigned different weights to vehicle types, reflecting their varying impacts on traffic flow (e.g., lightweight = 1, moderate weight = 2, heavyweight = 3, and no vehicle = 0). Their approach offered a nuanced understanding of traffic dynamics by considering vehicle-specific attributes. These diverse approaches underscore the evolving landscape of traffic control approaches, juxtaposing infrastructure-based sensing with the opportunities arising from connected vehicle technologies.

Regardless of different sensing options, few researchers have considered the robustness of traffic signal control to various disturbances, such as sensor errors, traffic accidents, demand surges, and sensor failure [7]. Any one of these problems could lead to catastrophic consequences. To enhance the traffic signal control performance in response to diverse traffic flow dynamics and unexpected events like sensor failure, we introduce StageLight, a two-stage control approach, empowered by precise sensing and multiscale learning. Through extensive experiments simulating realistic traffic dynamics in Simulation of Urban Mobility (SUMO), our approach showcases superior performances compared to the traditional and state-of-the-art methods, especially for moderate and heavy traffic conditions.

2. Method

Recently, many studies have explored DRL-based approaches, such as Rainbow [8], DDPG [9], AC [10], A2C [11], A3C [12], PPO [13], and DQN [14]. Among these, deep Q learning has been proven to be capable of handling realistic situations with complex road geometries and traffic scenarios. Notably, DQN-based methods have gained significant traction in the domain of traffic signal control due to their sample efficiency and stable performance. In our study, we adopt the DQN-based control framework illustrated in Figure 1, where the agent undergoes iterative interactions with a simulated environment to learn and optimize timings.

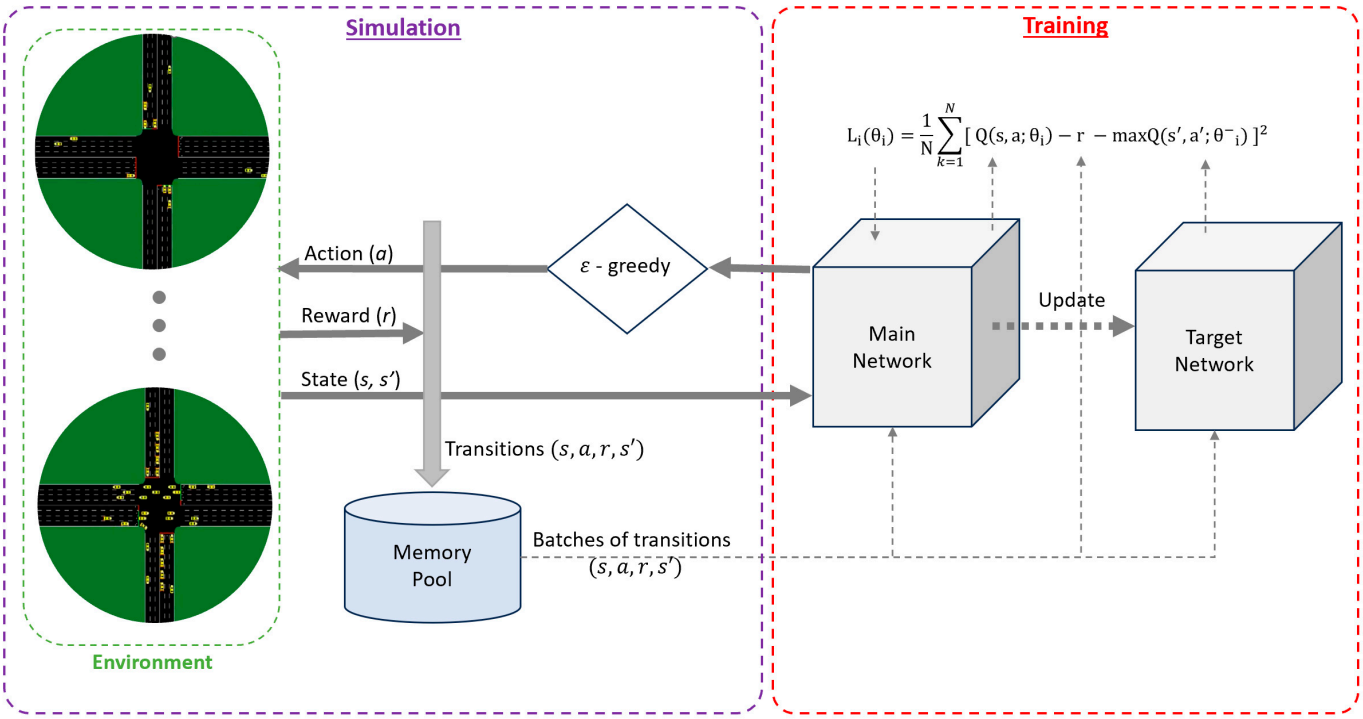


Figure 1. Illustration of DQN for traffic signal control.

Specifically, the signal control agent observes the current traffic state from the simulated environment and passes this information to the main network, also known as the online network. This network serves to map traffic states and corresponding actions to a value function, commonly referred to as the Q function. This Q function assesses the desirability of each action based on the expected reward from the current state. Once an action is selected, it is executed in the traffic-simulation environment, which generates feedback in the form of a reward signal, and the system moves to the next state. These transitions, encapsulated as quadruples (state, action, reward, next state), are stored in a memory pool. This memory pool serves as a repository of collected experience used to train the main network. For the training, a batch of transitions is sampled from the memory pool, and the loss is calculated using Equation (1).

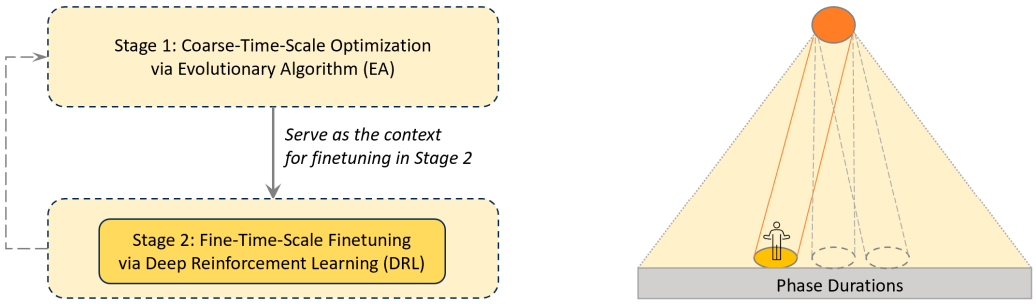
$$L_i(\theta_i) = \frac{1}{N}\sum_{k=1}^{N}\left[Q(s,a;\theta_i) - r - \max Q\left(s',a';\theta^-_i\right)\right]^2 \quad (1)$$

where θ^- represent target network parameters. a′ is the next action with the maximum Q value that the target network predicts based on state s′. The target network mirrors the main network but with its parameters being updated (i.e., copied from the main network) every τ steps.

In this study, we aim to improve upon the previous studies and address the DRL-based traffic signal control problem by introducing a two-stage multi-scale learning approach. In particular, it continuously learns a good initialization state on a coarse time scale in the first stage, followed by finetuning on a finer time scale in the second stage. The first stage aims to establish a generally good state to start with, while the second stage incrementally adapts signal timings to actual traffic demands in real time. By starting at a vintage point, which largely defines the region of exploration, this approach allows the agent to explore more effectively while maintaining robustness.

The name StageLight has a two-fold meaning: (1) the algorithm includes two stages (Figure 2a), where the initial stage focuses on coarse time-scale optimization, which establishes the context for the subsequent finetuning stage performed on a finer time scale;

(2) its work mechanism is analogous to a real stage light (Figure 2b), where the stage sets the context, and the performer is tracked (finetuned) by the spotlight.



(a) Two-stage multiscale learning (b) Analogy of a stage light

Figure 2. The concept of StageLight.

Previous studies only considered the flexibility of actions on a fine time scale while neglecting the convergence and robustness. Our approach explicitly addresses these aspects by firstly establishing a vantage point within the action space on a coarse time scale. This facilitates exploration within a perturbed region around this point during the subsequent stage, where a DQN is utilized for finetuning on a finer time scale.

Particularly, our approach resonates with the current practice in traffic signal timing, where time-of-day (TOD) plans are usually developed and implemented by the time of day. These TOD plans are periodically updated via retiming. The main difference lies in the fact that StageLight's two-stage operation (i.e., optimization on a coarse time scale and finetuning on a finer time scale) will be continuously carried out as opposed to the periodic re-timing in practice.

2.1. Coarse-Time-Scale Learning (Stage 1)

To facilitate continuous learning, we proposed an evolutionary gradient-based method, described in Algorithm 1, to learn optimal timings on a coarse time scale (e.g., optimized over an hour). The basic idea follows deterministic policy gradient ascent, where the gradients are estimated via an antithetic sampling of actions in a randomly sampled direction.

Algorithm 1: StageLight Stage 1—Learning optimal timing on a coarse time scale

1 : Initialize timing $A = [a_1, a_2, a_3, a_4]$; Roll out A to obtain the total reward, R_A
2 : Generate random perturbations : $\Delta = [\delta_1, \delta_2, \delta_3, \delta_4]$
3 : $A^+ := A + \Delta$; $A^- := A - \Delta$
4 : Roll out A^+ and A^- to obtain R_{A^+} and R_{A^-}
5 : If $\underset{A^+,A,A^-}{\text{argmin}}[R_{A^+}, R_A, R_{A^-}] == A$:
6 : Go to step 2
7 : Else:
8 : $R_A := \min[R_{A^+}, R_{A^-}]$
9 : $A' := A + \frac{\eta}{2\Delta}(R_{A^+} - R_{A^-})$ note: η is a hyperparameter, indicating learning rate)
10 : Roll out A' to obtain R'_A
11 : $A := \underset{A,A'}{\text{argmin}}[R_A, R_{A'}]$
12 : Go to step 2

2.2. Fine-Time-Scale Finetuning (Stage 2)

The phase duration obtained from Algorithm 1 serves as initial state as well as contextual information for further finetuning, which is conducted via a defined sequence of

phases. The action set contains three actions: $\{-\delta, 0, +\delta\}$, where δ is a small-time perturbation in seconds. The subsequent action (adjustment) is implemented by Algorithm 2 upon the outcome of the previous action in a cumulative fashion. To avoid drifting too far from the optimal solution learned on the coarse time scale from Stage 1, the finetuning is subject to the minimum green duration and bounded by a predefined range.

Algorithm 2: StageLight: Stage 2—Adjusting timings with DRL on a fine time scale

Start with the outcome of Stage 1 $\varnothing = [t_1, t_2, t_3, t_4]$
Initialize the Main Network, Target Network, and Memory Pool
Initialize the Environment
While episode < Maximum number of episodes:
    //Simulation//
    Observe environment state s_0
    While step $<$ total timesteps (N_s):
        $a_t = \begin{cases} a\ random\ action\ a_t,\ if\ (p < \varepsilon), \\ argmax_a Q(s_t, a; \theta),\ otherwise. \end{cases}$
        Greenduration $= \varnothing[phase] + a_t$
        Check the bound and minimum green duration, then run in environment.
        Update $\varnothing$ with incremental adjustments.
        Get r_t and environment state s_{t+1}
        Add transitions (s, a, r, s') into the Memory Pool
    ε declines linearly.
    //Network training//
    While epoch < Maximum number of epochs:
        Prioritized sampling a batch of transitions from the Memory Pool
        Update the Main Network parameter θ based on Adam [15]
        If epoch % N_e == 0: note: N_e = number of epochs for updating target network
            Target Network ← Main Network
Save Main Network.

2.3. State Representation

Following the discrete traffic state encoding (DTSE) in the previous study [16], our state definition utilizes high-spatial-resolution data that can be acquired via modern sensors, such as cameras. Specifically, grid cells are defined and tracked on each approaching lane of an intersection. Figure 3 shows our state representations, consisting of three maps, i.e., position map, speed map, and delay map. For the position map, if a cell is occupied by a vehicle, then the cell status is indicated by 1; otherwise, 0. For the speed map, actual speed values are used in corresponding cells. For the delay map, the cumulative count of positions at each cell over time is reported, which is equivalent to the delay. This delay map contains the running delay history of the current phase, which could be interpreted as a partial reward signal. Thus, inclusion of this delay map serves as a dynamically encoded cue to facilitate learning.

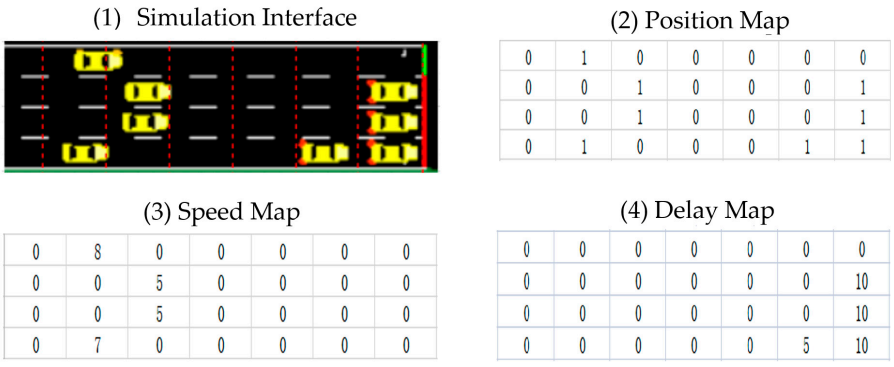


Figure 3. Example of state representation (speed in meters/second; delay in seconds).

It should be noted that, besides the state map representation, the phase durations learned in stage 1 are encoded as contextual information as part of the main network to stabilize the learning process.

2.4. Reward Definition

The average control delay is a commonly used service measure at signalized intersections. For practicality, we used average stop delay because the stopped vehicles can be conveniently tracked using modern traffic sensors. The reward is thus defined as negative average stop delay by Equation (2).

$$r_t = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} w_{i_t} \quad (2)$$

where w_{i_t} is computed as the i^{th} vehicle's cumulative stop time in the t^{th} interval, $N =$ the total number of stopped vehicles, and $T =$ the total number of simulation steps. w_{i_t} is computed as the presence of a stopped vehicle i in its corresponding cell (indicated by 0 or 1) multiplied by the uniform sampling interval (1 s) for each simulation step t.

3. Experiments

The popular traffic simulation software, SUMO version 1.14.0 [17], was employed for our experiments. Traci (SUMO API) is utilized as the interface to provide traffic state information and execute action in the traffic simulator. The detailed simulation settings are described in the following subsection.

We compares StageLight with three conventional methods, including FixedTime [18], MaxPressure [19,20], and Actuated Control [21], as well as the state-of-the-art DRL methods, including Change Stay [16], FRAP++ [22], and MPLight [23].

- FixedTime [18]: Phase durations are found by minimizing the average delay and then implemented as fixed timings.
- Actuated Control [21]: Adapt to traffic demand by extending the green phase duration or skipping a phase. The phase order is fixed.
- MaxPressure [20]: Choose the phase which has the maximum pressure according to the incoming and outgoing vehicle numbers. The optimal detect length of the incoming and outgoing lanes is usually determined based on the experimental results. The phase order is flexible.
- Change Stay: Similar to the actuated control, the required optimized variables are the phase extension and phase skip, subject to the minimum and maximum greens. Hence, the action can be a binary choice {0, 1}, indicating whether to skip to the next phase in the sequence (Change) or extend the current green phase (Stay) [24].
- FRAP++ [22]: Improved on FRAP [25] with further modifications. Specifically, FRAP++ uses the mean representation across lanes. The action definition is similar to the MaxPressure method.
- MPLight [23]: A state-of-the-art model that was introduced in a multi-intersection setting. In this study, we implemented it in a single intersection setting. MPLight uses the FRAP++ network architecture as the backbone. State and reward are expressed in the pressure form.

3.1. Experiment Settings

3.1.1. Intersection Geometry

The simulated intersection geometry includes four approaches. For north–south approaches, there are two through lanes and one left-turn lane (a total of six approaching lanes). For east–west approaches, there are three through lanes and one left-turn lane (a total of eight approaching lanes). Each lane is 750 m long, resulting in 100 grid cells by defining the cell to be 7.5 m long.

3.1.2. Traffic Scenarios

To realistically model traffic dynamics, we assume the vehicle arrival time follows Weibull distribution, as illustrated in Figure 4. Four different traffic-demand scenarios (500 vehicles/h, 1000 vehicles/h, 1500 vehicles/h, 2000 vehicles/h), representing light, moderate, heavy, and congested traffic flows, respectively, were simulated.

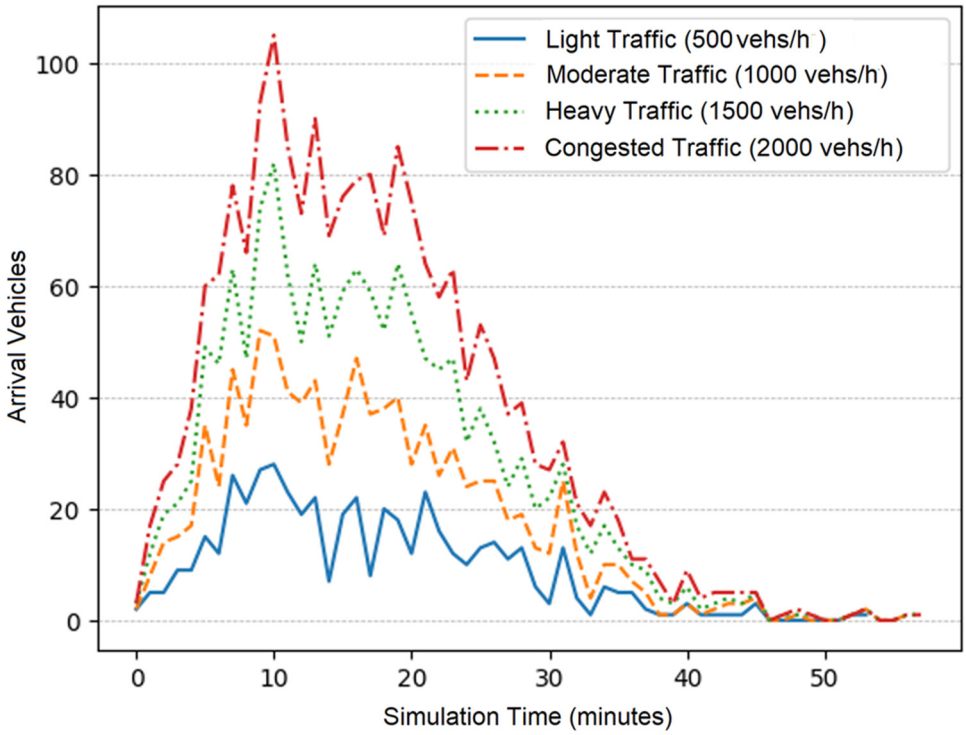


Figure 4. An example of simulated traffic scenarios: 500, 1000, 1500, 2000 vehicle arrivals in 60 min.

Upon generating the arrival vehicles, the simulation environment will randomly distribute them to different approaches and movements by following the probability in Table 1.

Table 1. Arrival probabilities by approach and movement.

Approach	Arrival Probabilities		
	Left Turn	**Through**	**Right Turn**
Northbound	1/32	3/16	1/32
Southbound	1/32	3/16	1/32
Eastbound	1/32	3/16	1/32
Westbound	1/32	3/16	1/32

3.1.3. Neural Network Architecture

The network architecture adopted is shown in Figure 5. The input shape is [3, 14, 100]. The first dimension represents three maps (i.e., position, speed, and delay). The second dimension denotes the 14 approaching lanes to the intersection. The third dimension indicates the length of detection area per lane, which is equivalent to 100 cells. For feature extraction, 2D convolutions are applied, followed by ReLU for nonlinearity. No pooling layers are used to prevent informational loss. Phase durations from Stage 1 and one-hot encoding for the current phase are concatenated with the state representation to provide contextual information.

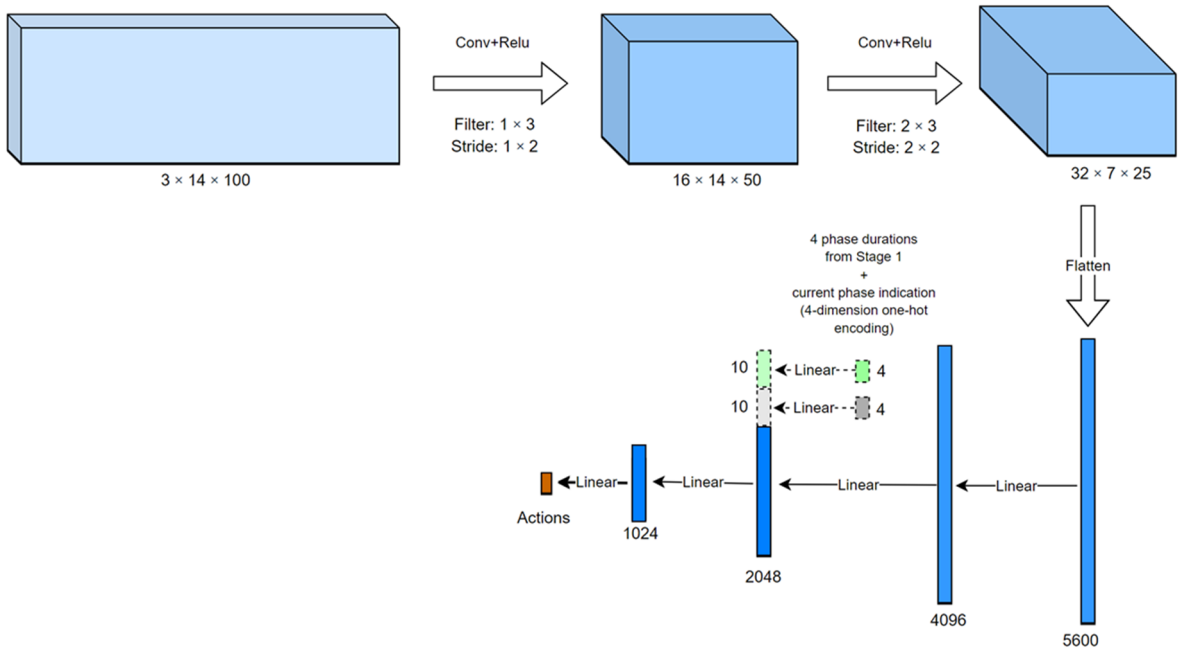


Figure 5. Neural Network Architecture.

3.1.4. Parameter Settings

The simulation and training parameters are summarized in Table 2. The parameter settings for each method are detailed in Table 3. For balancing exploration and exploitation, we employ the ε-greedy method, where ε decreases in a linear fashion for the first 250 episodes.

Table 2. Experiment settings.

Simulation Parameter	**Value**
Training episodes	500
Testing episodes	100
Total timesteps per episode (N_s)	3600
Total entering vehicles per episode	Light traffic: 500 vehs/h
	Moderate traffic: 1000 vehs/h
	Heavy traffic: 1500 vehs/h
	Congested traffic: 2000 vehs/h
Minimum green	7 s
Clearance interval (yellow + all red)	4 s
Detection zone length	75 m for FRAP++; 150 m for MPLight
Training Parameter	**Value**
Training batch size	100
Learning rate	0.0001
Training epochs	300
Number of epochs to update target network (N_e)	50
Discount factor	0.8

Table 3. Parameter settings of different methods.

Methods	Parameters
FixedTime	Light traffic: [12, 9, 16, 7] * Moderate traffic: [18, 7, 16, 7] * Heavy traffic: [30, 11, 22, 12] * Congested traffic: [41, 19, 31, 14] *
Actuated	Minimum green: 7 s Action: extend 3 s or move to the next phase in sequence
Change Stay	
MaxPressure	Minimum green: 7 s Action: one of the four phases
FRAP++	
MPLight	
StageLight	Action: {−3 s, 0, +3 s} Bound: [−9 s, +9 s]

* Indicates the durations in seconds for the four phases in sequence.

3.2. Experimental Results

The overall performances of all methods are presented in Table 4. The table shows that MaxPressure works well when traffic is light, while the three modern DRL methods, i.e., FRAP++, MPLight, and StageLight, demonstrate similar performance. In fact, the MaxPressure mechanism is similar to the MPLight and FRAP++ mechanisms, except that it ignores the evaluation of the previous action and the expected state value of the next action, resulting in worse adaptation to flow dynamics. As traffic increases, the benefits of the DRL methods become evident, and our proposed StageLight outperforms all other methods under moderate, heavy, and congested traffic conditions.

Table 4. Performance comparison (average delay in seconds).

Model	Traffic Scenarios			
	Light	**Moderate**	**Heavy**	**Congested**
MaxPressure	**7.54**	16.01	46.82	115.59
Actuated	14.27	17.55	39.9	50.42
FixedTime	16.36	19.16	30.74	54.84
Change Stay	13.28	17.17	29.79	95.37
FRAP++	8.38	14.92	27.83	56.88
MPLight	7.82	13.42	24.19	42.65
StageLight	8.19	**11.6**	**17.94**	**40.04**

Bold indicates the best performance.

3.2.1. Generalizability

For generalizability evaluation, we compare the three modern DRL-based methods by training them on one traffic-flow scenario while testing them on a different traffic scenario. In this setting, we alternate training and testing on two traffic-flow scenarios: moderate traffic flow (1000 vehs/h) and congested traffic flow (2000 vehs/h). The performances of the three DRL models are presented in Table 5, showing the superior generalizability of StageLight as compared to FRAP++ and MPLight. The positive percentages denote the increase in delay.

Table 5. Generalizability analysis on FRAP++, MPLight, and StageLight.

		Test on	
		Moderate Traffic (1000 vehs/h)	**Congested Traffic (2000 vehs/h)**
Train on	Moderate traffic (1000 vehs/h)	0	$+20\%^{(a)}$, $+3\%^{(b)}$, $+1\%^{(c)}$
	Congested traffic (2000 vehs/h)	$+2\%^{(a)}$, $+3\%^{(b)}$, $0\%^{(c)}$	0

Note: (a) FRAP++, (b) MPLight, (c) StageLight.

In general, an agent trained in a more dynamic traffic environment can be easily adapted to the less dynamic traffic environment, largely due to the fact that a high dynamic traffic environment usually contains more diverse trajectory patterns, covering those of less dynamic ones.

3.2.2. Robustness

Robustness to sensor failure is a practical concern for traffic control systems. Nevertheless, few researchers have considered the effect of sensor failure in the design of traffic control algorithms. In practice, the sensor failure would typically trigger a fallback mode, such as time-of-day (TOD) plans. In this evaluation, we aim to show the robustness of StageLight to different sensor-failure scenarios, as shown in Figure 6.

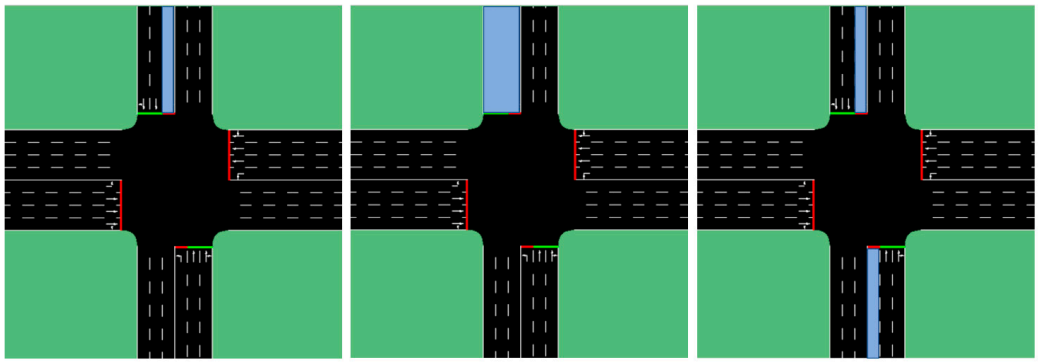


Figure 6. Sensor-failure scenarios (Note: the masked area in blue indicates sensor failure, where all sensor outputs are suppressed to 0).

For demonstration purposes, this evaluation was conducted for the congested traffic scenario (2000 vehs/h), where sensor failure likely triggers network gridlock. The average delays for the three modern DRL models are summarized in Table 6.

Table 6. Average delay (seconds) for different sensor-failure scenarios in congested traffic.

Model	**Baseline**	**Sensor-Failure Scenario**		
		1	**2**	**3**
FRAP++	56.88	73.85 (+30%)	153.19 (+169%)	127.38 (+124%)
MPLight	42.65	45.83 (+7%)	46.57 (+9%)	115.66 (+171%)
StageLight	40.04	40.64 (+1%)	40.69 (+2%)	40.66 (+2%)

Note: the percentages in parenthesis indicate the percent increases in delay compared to the baseline.

As shown in Table 6, for all sensor-failure scenarios, FRAP++ exhibits the most significant performance decline, followed by MPLight. Conversely, StageLight displays the least susceptibility to these failures. This robustness of StageLight can be attributed to

its inherent design. Primarily, the optimal timing learned on the coarse time scale works well with perturbed flows in the region and also serves as contextual information in the finetuning stage. This unique feature, coupled with the phase-duration bound imposed during the finetuning stage, has led to StageLight's robust performance across various sensor-failure scenarios.

4. Conclusions

The paper introduces StageLight, a two-stage multiscale learning methodology specifically developed for robust traffic signal control. StageLight operates by initially acquiring optimal signal timing on a coarse time scale and subsequently leveraging this knowledge as contextual information for fine tuning on a finer time scale. This approach adeptly handles varied traffic flow dynamics arising from natural fluctuations and unexpected events such as sensor failures, ensuring consistently reliable signal operations.

Based on our experiments, the MaxPressure method performs well in light traffic scenarios characterized by highly random vehicle arrivals lacking distinct patterns. However, its performance deteriorates as the traffic density increases and congestion ensues. The Actuated and FixedTime methods exhibit slightly better performance in the heavy and congested traffic scenarios. Conversely, the three modern DRL methods, FRAP++, MPLight, and StageLight, demonstrate comparable performance to MaxPressure in light traffic scenarios. Their advantages become evident as the traffic intensity grows and diverse, learnable traffic patterns emerge. Notably, the StageLight method outperforms other methods under moderate, heavy, and congested traffic scenarios.

In addition to its superior performance across a wide spectrum of traffic conditions, StageLight showcases remarkable generalizability by adapting to traffic environments different from those it was trained on. Moreover, it demonstrates outstanding resilience against various sensor-failure scenarios. Nevertheless, it could be beneficial to study even more diverse traffic and sensor-failure scenarios. Future research endeavors could focus on extending the framework to multiple intersections on the corridor or network levels. In such settings, different network architectures, such as graph neural networks or transformers, might be more suitable for capturing dynamic coordination among intersections. Furthermore, while the robustness tests were conducted specifically on sensor-failure scenarios, expanding these tests to encompass various cyberattack scenarios would undoubtedly enhance the system stability and security.

Author Contributions: Conceptualization, J.J.Y.; Software, G.S.; Formal analysis, G.S.; Investigation, G.S. and J.J.Y.; Resources, J.J.Y.; Writing—original draft, G.S.; Writing—review & editing, J.J.Y.; Project administration, J.J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tan, J.; Yuan, Q.; Guo, W.; Xie, N.; Liu, F.; Wei, J.; Zhang, X. Deep Reinforcement Learning for Traffic Signal Control Model and Adaptation Study. *Sensors* **2022**, *22*, 8732. [CrossRef] [PubMed]
2. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **2005**, *16*, 285–286. [CrossRef]
3. Bouktif, S.; Cheniki, A.; Ouni, A.; El-Sayed, H. Deep reinforcement learning for traffic signal control with consistent state and reward design approach. *Knowl.-Based Syst.* **2023**, *267*, 110440. [CrossRef]
4. Liu, D.; Li, L. A traffic light control method based on multi-agent deep reinforcement learning algorithm. *Sci. Rep.* **2023**, *13*, 9396. [CrossRef] [PubMed]
5. Shi, Y.; Wang, Z.; LaClair, T.J.; Wang, C.; Shao, Y.; Yuan, J. A Novel Deep Reinforcement Learning Approach to Traffic Signal Control with Connected Vehicles. *Appl. Sci.* **2023**, *13*, 2750. [CrossRef]

6. Kumar, R.; Sharma, N.V.K.; Chaurasiya, V.K. Adaptive traffic light control using deep reinforcement learning technique. *Multimed. Tools Appl.* **2023**, 1–22. [CrossRef]
7. Rodrigues, F.; Azevedo, C.L. Towards Robust Deep Reinforcement Learning for Traffic Signal Control: Demand Surges, Incidents and Sensor Failures. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3559–3566.
8. Nawar, M.; Fares, A.; Al-Sammak, A. Rainbow Deep Reinforcement Learning Agent for Improved Solution of the Traffic Congestion. In Proceedings of the 7th International Japan-Africa Conference on Electronics, Communications, and Computations, (JAC-ECC), Alexandria, Egypt, 15–16 December 2019; pp. 80–83.
9. Pang, H.; Gao, W. Deep Deterministic Policy Gradient for Traffic Signal Control of Single Intersection. In Proceedings of the Chinese Control and Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 5861–5866.
10. Mousavi, S.; Schukat, M.; Corcoran, P.; Howley, E. Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning. *arXiv* **2017**, arXiv:1704.08883. [CrossRef]
11. Xiong, Y.; Zheng, G.; Xu, K.; Li, Z. Learning Traffic Signal Control from Demonstrations. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2289–2292.
12. Wang, M. Traffic Signal Control Method Based on A3C Reinforcement Learning. In Proceedings of the 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), Guiyang, China, 23–25 July 2021; pp. 127–132.
13. Li, Y.; He, J.; Gao, Y. Intelligent Traffic Signal Control with Deep Reinforcement Learning at Single Intersection. In Proceedings of the 7th International Conference on Computing and Artificial Intelligence, Tianjin, China, 23–26 April 2021; pp. 399–406. [CrossRef]
14. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
15. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
16. Genders, W.; Razavi, S. Using a Deep Reinforcement Learning Agent for Traffic Signal Control. *arXiv* **2016**, arXiv:1611.01142.
17. Urbanik, T.; Tanaka, A.; Lozner, B.; Lindstrom, E.; Lee, K.; Quayle, S.; Beaird, S.; Tsoi, S.; Ryus, P. *Signal Timing Manual*, 2nd ed.; NCHRP Report 812; Transportation Research Board: Washington, DC, USA, 2015.
18. Koonce, P.; Rodegerdts, L.; Lee, K.; Quayle, S.; Beaird, S.; Braud, C.; Bonneson, J.; Tarnoff, P.; Urbanik, T. *Traffic Signal Timing Manual*; Publication Number: FHWA-HOP-08-024; United States. Federal Highway Administration: Washington, DC, USA, 2008.
19. Varaiya, P. The max-pressure controller for arbitrary networks of signalized intersections. In *Advances in Dynamic Network Modeling in Complex Transportation Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 27–66.
20. Wei, H.; Chen, C.; Zheng, G.; Wu, K.; Gayah, V.; Xu, K.; Li, Z. PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1290–1298. [CrossRef]
21. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent Development and Applications of SUMO—Simulation of Urban MObility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138. Available online: https://www.iariajournals.org/systems_and_measurements/ (accessed on 6 December 2023).
22. Zang, X.; Yao, H.; Zheng, G.; Xu, N.; Xu, K.; Li, Z. MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020. [CrossRef]
23. Chen, C.; Wei, H.; Xu, N.; Zheng, G.; Yang, M.; Xiong, Y.; Xu, K.; Li, Z. Toward a Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020. [CrossRef]
24. Zeng, J.; Hu, J.; Zheng, Y. Adaptive Traffic Signal Control with Deep Recurrent Q-learning. *IEEE Intell. Veh. Symp. (IV)* **2018**, *34*, 1215–1220.
25. Zheng, G.; Xiong, Y.; Zang, X.; Feng, J.; Wei, H.; Zhang, H.; Li, Y.; Xu, K.; Li, Z. Learning Phase Competition for Traffic Signal Control. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1963–1972.