applied sciences

MDPI

Article

A Novel Deep Reinforcement Learning Approach to Traffic Signal Control with Connected Vehicles†

Yang Shi [1], Zhenbo Wang [1,*], Tim J. LaClair [2], Chieh (Ross) Wang [2], Yunli Shao [2] and Jinghui Yuan [2]

[1] Department of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, Knoxville, TN 37996, USA
[2] Buildings and Transportation Science Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
* Correspondence: zwang124@utk.edu

Abstract: The advent of connected vehicle (CV) technology offers new possibilities for a revolution in future transportation systems. With the availability of real-time traffic data from CVs, it is possible to more effectively optimize traffic signals to reduce congestion, increase fuel efficiency, and enhance road safety. The success of CV-based signal control depends on an accurate and computationally efficient model that accounts for the stochastic and nonlinear nature of the traffic flow. Without the necessity of prior knowledge of the traffic system's model architecture, reinforcement learning (RL) is a promising tool to acquire the control policy through observing the transition of the traffic states. In this paper, we propose a novel data-driven traffic signal control method that leverages the latest in deep learning and reinforcement learning techniques. By incorporating a compressed representation of the traffic states, the proposed method overcomes the limitations of the existing methods in defining the action space to include more practical and flexible signal phases. The simulation results demonstrate the convergence and robust performance of the proposed method against several existing benchmark methods in terms of average vehicle speeds, queue length, wait time, and traffic density.

Keywords: traffic signal control; deep reinforcement learning; autoencoder neural network; representation learning

Citation: Shi, Y.; Wang, Z.; LaClair, T.; Wang, C.; Shao, Y.; Yuan, J. A Novel Deep Reinforcement Learning Approach to Traffic Signal Control with Connected Vehicles. *Appl. Sci.* **2023**, *13*, 2750. https://doi.org/10.3390/app13042750

Academic Editor: Xinlin Huang

Received: 3 January 2023
Revised: 16 February 2023
Accepted: 17 February 2023
Published: 20 February 2023

1. Introduction

Recent advances in communication technology, transportation infrastructure, and computational techniques, along with the application of artificial intelligence, will enable a potential to revolutionize the future of transportation systems. Many countries are making great efforts toward this transition [1]. Connected vehicles (CVs) are among the most promising emerging technologies, offering numerous benefits to road safety, traffic mobility, and energy efficiency. As the implementation of CVs and the transition to a connected transportation system evolve, the development of reliable and efficient control algorithms for both the infrastructure and in-vehicle components will be crucial.

The CV technologies create a dynamic and interconnected environment for drivers, vehicles, and traffic infrastructure. In this environment, connectivity plays a critical role. Through wireless communication, vehicles can communicate real-time data such as location, speed, and acceleration with other vehicles (V2V) and infrastructure (V2I). These real-time data enable traffic controllers to optimize signal phase and timing (SPaT) plans for enhanced road safety and sustainability. However, the complexity of SPaT optimization, considering

realistic driving behavior and multiple objectives, presents a significant challenge and remains an open research area, due to the NP-complete nature of the problem [2,3] and the "curse of dimensionality" associated with an increasing number of vehicles and traffic lights in the network.

With the large amounts of traffic data generated by CVs, it is possible to characterize the interactions between vehicles and traffic infrastructure components, thus enabling the development of data-driven traffic control strategies. Non-parametric learning approaches, particularly reinforcement learning (RL), are well suited for characterizing the stochastic and non-linear nature of traffic flow. These techniques allow the signal controller to learn policies by observing the transition of the traffic states, without the need for prior knowledge of the system's model structure [4]. In other words, RL-based signal control approaches eliminate the burden of building complex decision-making models for highly dynamic, nonlinear, stochastic traffic system.

However, constructing and training a learning-based controller directly from raw data presents a major challenge. Without proper design of the learning models and training algorithms, the learning-based controller may not be able to effectively learn a control strategy. To tackle the issue of the "curse of dimensionality", many researchers have to simplify the training model by restricting the action and state space, which reduces the practicability and optimality of the resulting controllers. In this article, we aim to overcome these limitations by defining the action space in a way that allows for more practical and flexible signal timings, and by restructuring the state space to improve the learning performance.

The main contributions of this paper are as follows:

1. A new traffic signal control framework using deep reinforcement learning (DRL) is proposed, by incorporating a novel convolutional autoencoder network to reduce the dimensionality of the input traffic states. As a result, a concise representation of comprehensive traffic information is obtained and utilized to facilitate the learning of effective SPaT plans.
2. The action space is extended by including both phase duration and cycle length, allowing for increased adaptability to dynamic traffic flow. With the combinatorial action space of multiple phase durations and cycle lengths, our method can effectively handle unbalanced traffic flow with varying traffic volumes.
3. To improve the learning efficiency of the proposed DRL algorithm, several state-of-the-art techniques, such as target network [5], dueling network [6], and experience replay [7] are implemented.
4. The effectiveness and performance of our method are demonstrated by comparing to several existing traffic signal control methods through simulations on the widely used Simulation of Urban MObility (SUMO) traffic simulator.

The structure of this paper is organized as follows: Section 2 presents a literature review of related work, including both optimization-based and learning-based signal control approaches. Section 3 describes the considered traffic scenario and details the development of the proposed methodology. Section 4 presents the simulation results and comparative analysis with other approaches to demonstrate the effectiveness and performance of the proposed method. Finally, the paper concludes with a summary of its findings and suggestions for future work in Section 5.

2. Related Work

For the past 20 years, numerous studies has been conducted to tackle traffic signal control issues in the presence of CVs. These studies primarily focus on improving the performance of isolated intersections, with the goal of scaling the solutions to larger networks and corridors. According to the mathematical models used, these methods can be broadly categorized into two groups: optimization-based and machine learning-based approaches [8].

2.1. Optimization-Based Approaches

Optimization-based approaches assume that the traffic model is known and the future traffic flow states could be predicted accordingly. Next, certain optimization problems are formulated and solved for the optimal SPaT control plans. The objectives of these optimization problems are usually to minimize traffic performance measures, such as traffic delay and queue length, which are estimated on the basis of predicted vehicle arrivals [8]. However, this approach requires accurate predictions of future traffic states, which can be challenging due to the complexity of the optimization problem that involves traffic flow models and couples with SPaT data. Therefore, the key challenges in CV-based traffic controls are to predict the future traffic states accurately, coordinate multiple intersections effectively by accounting for the conflicts of traffic flows, and efficiently solve the underlying large-scale optimization problem [8]. These challenges have led to the development of three different groups of optimization methods: centralized, decentralized, and hierarchical approaches.

In comparison to conventional signal control methods, such as adaptive control and coordinated control, the biggest challenge in implementing optimization-based methods is the high complexity of optimization models. To address this issue, centralized approaches reformulate the optimization problem by reducing the number of variables. For instance, in [9], individual vehicles were grouped into pseudo-platoons based on the headways between them, and a mixed-integer linear program (MILP) was utilized to determine the optimal signal phase sequence and phase initialization in real-time using platoon request data and traffic controller status. This work also introduced a dynamic arterial coordination strategy to promote traffic progression by taking into account platoon queue delay, signal delay in the current intersection, and possible delay at downstream intersections.

In [10], a real-time adaptive phase allocation algorithm was proposed that utilizes dynamic programming and optimization techniques to allocate signal phase sequences and duration based on predicted vehicle arrivals. Zhao et al. [11] adopted an interactive grid search method to solve an optimization problem, considering accumulated fuel consumption and travel time as the cost function, to determine the optimal traffic light timing of for each cycle at an intersection. Ma and Liu [12] proposed a new optimization method based on an improved genetic algorithm, which was compared with the Webster algorithm and the traditional genetic algorithm. The average vehicle delay was used as the control objective, and the green signal ratio and cycle time were used as the control variables.

Mohebifard and Hajbabaie [13] used a cell transmission model [14] to categorize the traffic network into cells and groups for higher-level representation and then formulated an MILP to maximize network throughput, which was solved using the Benders decomposition technique [15]. Bin AI Islam et al. [16] formulated an optimization problem to minimize network-level traffic delay, considering the energy consumption as a constraint, and solved the resulting non-convex problem using a stochastic gradient approximation algorithm. In Hong et al. [17], a linear dynamic traffic system model was built for a large-scale traffic network and a linear-quadratic regulator was applied to minimize both traffic delay and control-input changes, allowing for an online update of the traffic model to be adaptive to signal control outcomes.

Decentralized approaches aim to simplify the model and lower the computational cost of the traffic control problem by utilizing distributed control and optimization techniques. These methods optimize objective functions for each intersection individually and disregard coordination among neighboring intersections, leading to sub-optimal, local solutions instead of globally optimal solutions. These approaches typically predict only the traffic states, often just the arrivals, of the current intersection over a certain time horizon. To address this challenge, Li and Ban [18] transformed the problem into a dynamic programming model by dividing the timing decisions into stages with one stage for each phase, and minimizing the accumulated fuel consumption and travel time by calculating the objective function for each phase. Goodall et al. [19] proposed a predictive microscopic simulation algorithm to estimate future traffic conditions and objectives over a rolling hori-

zon of 15 s, assuming vehicles maintain heading and speed during this horizon. To account for the impact of queue spillbacks, [20] presented a decentralized method to maximize global network throughput by maximizing the effective outflow rate of each intersection locally and independently. This approach determines the minimum saturated green time of all possible phases based on queue lengths, arrival flows, and downstream queue lengths at each intersection to facilitate vehicle discharge at full capacity.

In [3], a distributed, coordinated approach was developed to tackle the network control problem through dividing it into a series of local controllers that can exchange traffic data with each other. At each decision time step, each controller collects data on queue lengths and incoming vehicle numbers from neighboring intersections, and decides whether to end or maintain the existing signal phase for local signal timing till the next step. Moreover, Islam et al. [21] expanded upon this work by taking into account unconnected vehicles. Specifically, they developed two algorithms to estimate the traffic states of unconnected vehicles relying on the traffic information from fuse loop detectors and CVs using car-following concepts. In [22], both connected and identified non-connected vehicles were grouped into platoons, resulting in the generation of all possible platoon departure sequences. Rather than solving for optimal signal timing directly, the platoon departure sequence that minimizes total vehicle delay was found by enumerating all possible departure sequences. The optimal SPaT was then calculated as the time needed to discharge all of the vehicles in a platoon.

Hierarchical approaches address the complexities of the traffic network optimization problem by breaking it down into multi-level optimization problems with different objectives for each level. The key step of these approaches is to establish macroscopic and microscopic models for each level of the control problem. For instance, [23] proposed a two-level adaptive signal control method for corridor coordination. Two optimization problems with distinct objectives were formulated at the intersection and corridor levels. At the corridor level, an MILP was developed to optimize the offsets along the corridor while minimizing the platoon delay based on the movement of vehicle platoons. The optimized offsets were then sent to the intersection level as the coordination constraints. At the intersection level, individual vehicle movements were computed using a dynamic programming method to minimize individual vehicle delay and handle phase allocation for both coordinated and non-coordinated phases.

In Qiao et al. [24], a three-level multi-agent signal control system was proposed for an urban traffic network, including an intersection agent, a regional agent, and a central agent. Three corresponding objective functions were designed to minimize total delay time, reduce the total green ratio-related delay, and find the optimal signal cycle. The fireworks algorithm was employed by Tan et al. [25] to solved the optimization problems, resulting in optimal cycle length, offset, and green ratio that minimize the total delay time of all intersections.

2.2. Data-Driven Approaches

The increasing availability of data and computation power has made machine learning-based approaches more and more popular in traffic control. These approaches offer the advantage of being model-free, eliminating the need to build complex mathematical models to describe the traffic states and solve nonlinear optimization problems. For instance, Ma et al. [26] proposed the use of real-time high-precision vehicle trajectory and traffic flow data to better understand and control congestion, and suggested using deep learning algorithms and model predictive control theory to construct a congestion recognition and control optimization model for the urban roadway network.

Without a need for prior knowledge of the traffic system, machine learning-based approaches also reduce the likelihood of introducing errors to the estimation of traffic states. Additionally, machine learning approaches are less computationally intensive and have great potential for real-time applications, making them more practical than traditional model-based optimization methods. Moreover, machine learning-based controllers have the ability to continuously learn and adapt to changes in the traffic pattern, leading to

improved optimality. There have been efforts to use machine learning techniques to model the complicated relationship between the signal timing plans and traffic delays, as reported in [27]. Overall, machine learning-based approaches hold great potential for addressing the various challenges in the field of traffic signal control.

The critical elements in designing a machine learning-based approach for traffic control include (1) capturing the traffic state effectively, (2) selecting the appropriate learning algorithm, (3) defining clear learning objectives, and (4) designing an inclusive action space. For instance, Liang et al. [28] represented the traffic state of a single intersection as image-like grids, using an $n \times n \times 2$ matrix to denote the position and speed of vehicles within the grids. This matrix was then used as input to a convolutional neural network that calculated expected future rewards for each possible action from the action space, which was adjusting the current signal phase duration. The reward was defined as the reduction in cumulative waiting time between two successive signal cycles. By using the double Q-learning method to maximize the expected reward, the neural network could learn how to reduce the average waiting time of vehicles at an intersection. Ma et al. [29] employed a support vector machine (SVM) algorithm to model the relationship between the current traffic state and the optimal signal timings and an online learning algorithm to adjust the SVM model in real-time.

In [30], the traffic state of an intersection was represented using a set of normalized queue lengths in each lane, which were discretized through the application of the k-means clustering algorithm. To optimize energy consumption and mobility simultaneously, they utilized an RL algorithm with three different reward functions. At each decision time step, the signal controller agent made a decision to either end or continue the current signal phase. Similarly, [31] proposed a reward function with respect to a fixed-time controller. The agent receives positive rewards for better performance than the fixed-time controller, and negative rewards for under-performance. In [32], the traffic state was characterized by a 2-D matrix consisting of the number of stopped vehicles in each direction and the average speed measured in each section. The action space comprised two options: selecting signal phases and adjusting phase offsets. The reward function was a combination of the total volume that passed through the arterial network and the difference in queue lengths between the two different directions. To increase the adaptability of the signal control model, Yoon et al. [33] proposed a graph-based method that depicts the traffic state as graph-structured data, which were then input into a graph neural network to train the signal control policy. The study focused on an isolated intersection with only straight traffic flows and thus the SPaT was made up of two green-red phases and two yellow-red phases, and the action was defined as the ratio of the green time over a fixed signal cycle. To enhance the ability of the RL algorithm to generalize, Zeng et al. [34] incorporated prior traffic knowledge. They used a fully-connected network to classify the intersection's demand pattern, and combined the results with outputs from a convolutional network to produce joint Q-value approximations. The intersection state was represented by a discrete encoding matrix consisting of vehicle position, speed, and signal phase. The reward structure was a combination of the number of stopped vehicles and passing vehicles, phase changes, and total vehicle waiting time.

In the scenario of network-wide traffic control, the application of a centralized RL method faces many challenges, such as an exponential increase in the dimension of the action space as the number of traffic lights increases, making it difficult to find an effective joint control policy. To address this issue, some studies, such as [35,36], trained each intersection as an individual agent based on the local traffic states and information from neighboring intersections, while the overall system state and performance were determined by the joint actions of all intersections. However, the stationary reward distribution and environment dynamics are required by a Markov decision process. It is difficult for the agent to converge to a stationary policy when its rewards are influenced by neighboring intersections. To resolve this issue, [37] implemented a knowledge sharing mechanism to enhance cooperation and collaboration among traffic signals, where the "knowledge" was

a collective representation of the traffic environment collected by all agents and used for learning individual policies. Similarly, to improve the convergence of the multi-agent RL algorithm, [38] combined a group of traffic signals into a single agent through a *k*-nearest-neighbor-based joint state representation.

In summary, the studies reviewed above have highlighted the potential of data-driven approaches in developing signal control strategies that outperform conventional controllers such as fix-timing and actuated controllers. However, the scenarios considered were significantly simplified. Some studies limited the number of lanes and traffic directions, resulting in a smaller state space, while others limited the action space by either selecting a continuous space with fixed phase sequence and phase split action, or choosing a discrete action space with limited options for phases and duration or fixed cycle/phase length with phase switching as the only actions. These simplifications were made to reduce the complexity of the data-driven models, indicating that the existing methods face challenges in learning a practical and truly optimal signal control strategy.

3. Methodology

The deep reinforcement learning (DRL) approach has gained much attention due to its capability in learning high-level decision-making processes, as demonstrated in [39]. As a data-efficient method, it enables learning of decision-making by agents through interactions with the environment. A typical DRL-based control framework consists of three basic components: environment sensing, the action space of the agent, and a learning goal for the agent [4]. As demonstrated in previous studies, data-driven methods applied to traffic signal control problems outperform conventional methods in simulations involving large amounts of data.

The objective of this research is to improve the overall performance of data-driven signal control frameworks by enhancing training outcomes and yielding more adaptable SPaT results. As highlighted in the previous section, the existing DRL-based methods face various challenges in terms of learning efficiency and optimality. In particular, the learning structure needs to be revised to accommodate dynamic traffic conditions and improve learning efficiency. With this in mind, this paper presents a novel data-driven optimization framework for traffic signal control that leverages innovative DRL techniques. The structure of the proposed method is depicted in Figure 1.

3.1. Scenario and Simulation Environment

In this paper, we consider a traffic signal control problem for a typical four-way signalized intersection. As shown in Figure 2, the intersection has one left-turn lane, one through lane, and one right-turn lane in each direction. The isolated intersection and traffic are simulated by SUMO [40], a microscopic, space-continuous traffic simulation software, which allows to retrieve details of simulated objects and to adjust their parameters at every time step. The traffic signal at the intersection is managed by a DRL-based actor, which is also referred to as an agent. This agent continuously receives traffic states and a reward signal from the simulation environment and makes decisions based on the current traffic state.

Traffic States: The traffic state is represented by discrete encoding of position and speed information of vehicles around the intersection [41]. The simulated intersection is divided into squared mesh grids with equal length c, which can be represented by an $N \times N$ matrix. Each grid in the matrix has two values: one binary value that indicates the presence of a vehicle, and the other that stores the speed of the existing vehicle. An example of a 30×30 traffic state matrix is shown in Figure 3, where the yellow grids denote vehicles and the numbers show their speeds in m/s. Blank grids indicate the absence of at those positions. In real-world implementations, vehicle mobility information can be obtained through a vehicular network or other devices Jeong et al. [42].

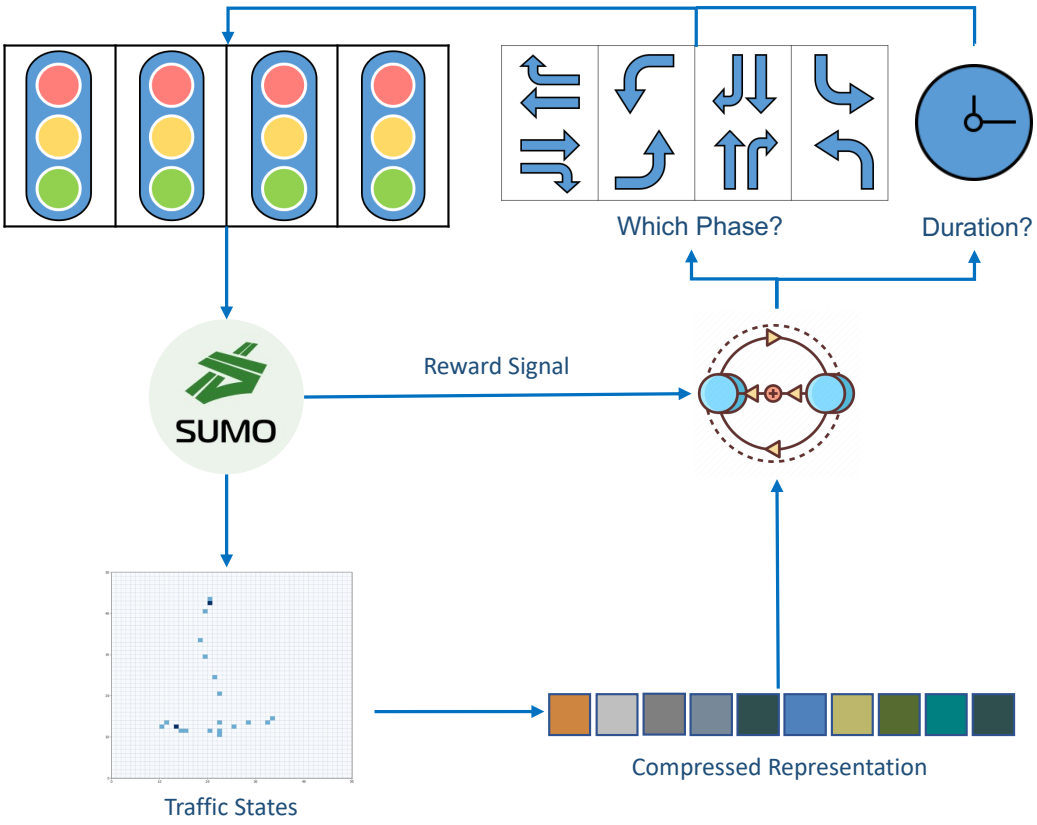


Figure 1. Proposed data-driven optimization framework for traffic signal control.

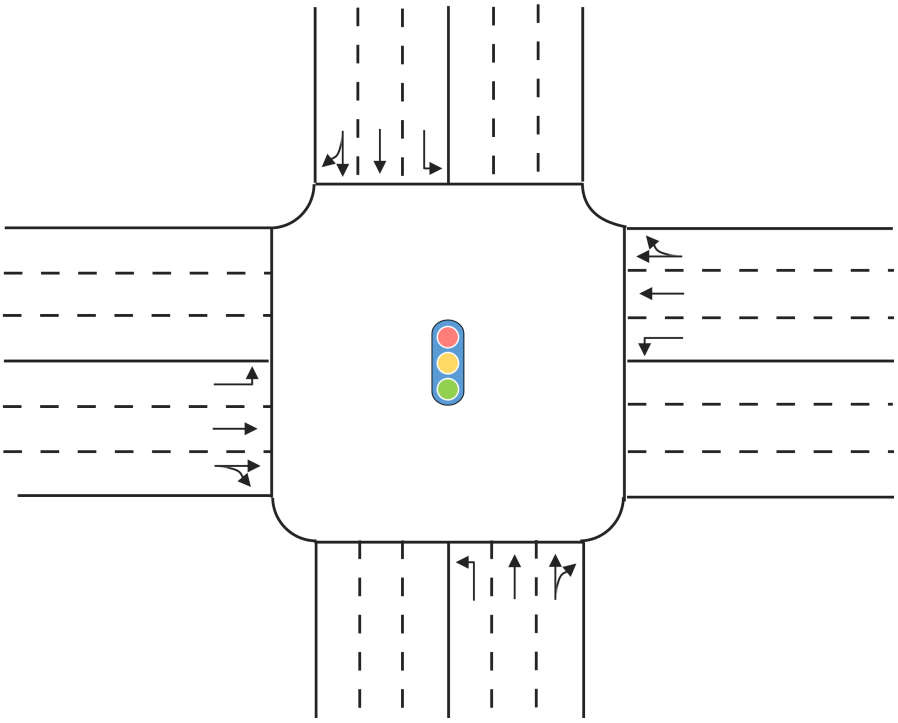


Figure 2. The simulated intersection.

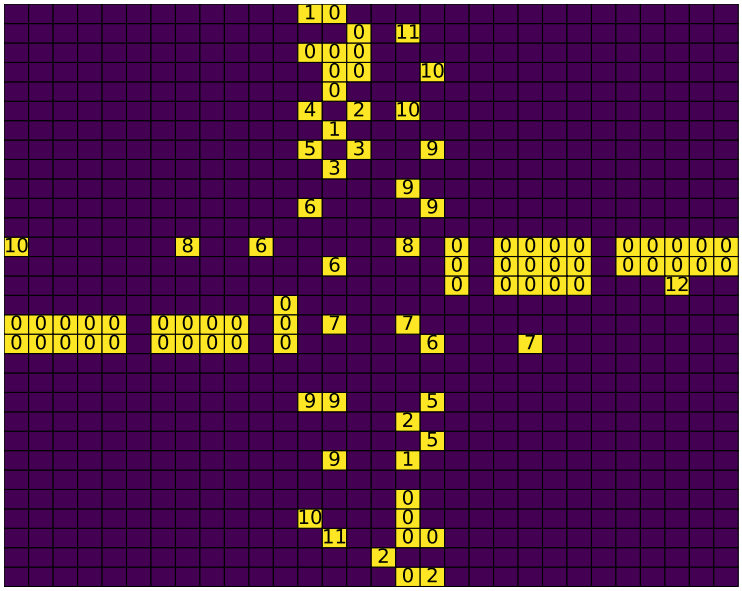


Figure 3. An example of traffic state matrix.

Action Space: This research focuses on two main types of discrete action spaces for RL-based traffic signal control. The first option involves selecting a signal phase from a fixed set of choices at predetermined time intervals, with the duration of each phase limited to a multiple of the time interval. The second option involves fixing the phase sequence of a signal cycle and adjusting the duration of each phase in the subsequent cycle at the end of the current cycle. As depicted in Figure 4, a typical four-phase signal cycle is considered as comprised of two straight and two left-turn phases. To discretize the action space, a combinatorial approach is used to choose the signal cycle length and phase splits.

The selection of the signal cycle length plays a crucial role in handling traffic volumes. A long signal cycle increases road capacity and prevents loss of green time that could occur with delayed response to the green light [43]. However, excessively long cycle lengths can result in increased congestion and long waiting queues. There is a trade-off between road capacity and traffic delay to consider when selecting the cycle length. To balance this, the selection of cycle length is limited to {10 s, 20 s, 30 s, 40 s, 50 s, 60 s} to prevent extremely long cycle lengths from slowing down the traffic flow. The available selections for phase duration range from 0 to the maximum cycle length, in increments of 5 s. This results in a total of 1035 possible actions. Additionally, during the last 3 s of each phase, the green lights turn yellow.

3.2. Compressed Representation of Traffic States

It is convenient to use the position and speed information of vehicles to build a traffic state matrix for input to the DRL training algorithm. However, the large space of the resulting traffic states makes it difficult for the DRL algorithm to identify a direct relationship between the traffic state and the signal control action. To address this issue, finding an appropriate traffic state representation is crucial. In this study, an autoencoder neural network is used to represent the complex traffic states of the whole intersection as a concise representation Baldi [44]. Having a state representation that contains rich information is vital for the control agent to make informed decisions, without being affected by the "curse of dimensionality". To achieve this, the dimension of state representation must be reduced while preserving as much information as possible. Additionally, the compressed state representation allows the underlying traffic pattern to be extracted as features by the autoencoder. Although it is hard to know the exact features learned by the autoencoder, feature extraction has been shown to be a useful approach in improving reinforcement learning in various applications [45].

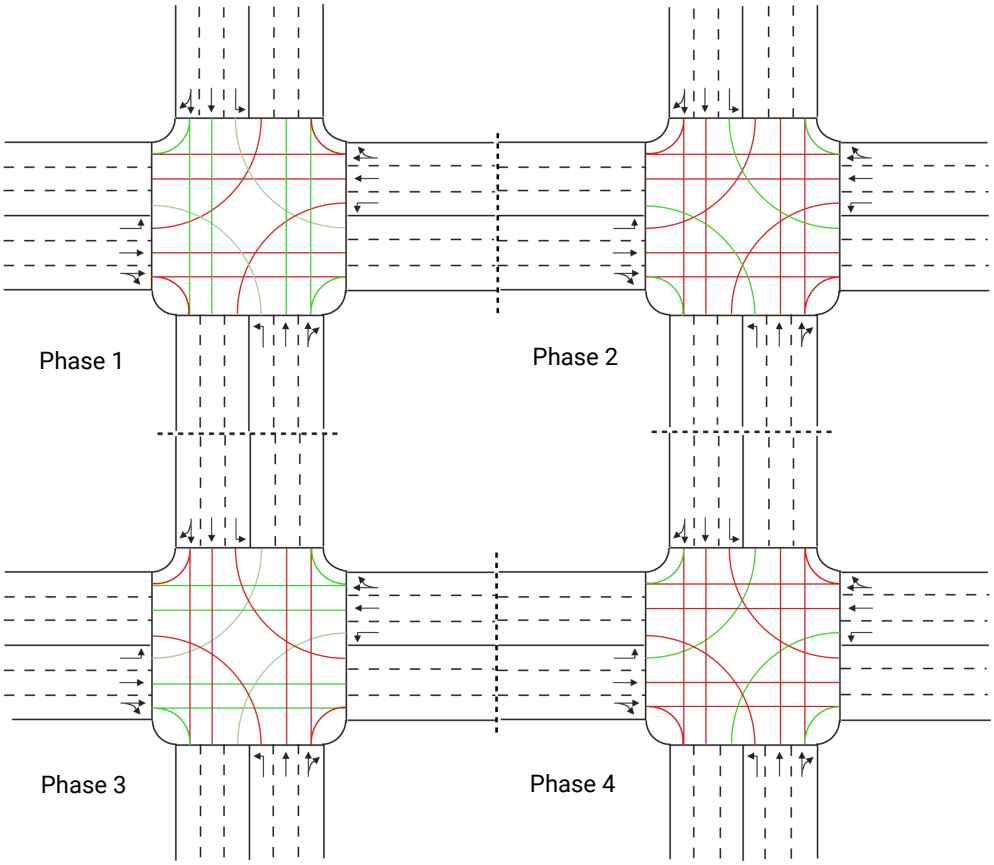


Figure 4. A typical signal cycle with four phases.

Autoencoder is a type of neural network that can learn a compact representation of the input data [46]. Convolutional neural networks, such as the Visual Geometry Group (VGG) neural networks [47], can extract inherent features from the spatial information in the input data. By organizing a convolutional neural network into an encoder–decoder architecture, the network can encode a static traffic state into a fixed-length vector, serving as input to the reinforcement learning model.

The proposed convolutional autoencoder (CAE) network, shown in Figure 5, has a mirror structure of two functional components: the encoder and decoder. The task of the CAE is to reconstruct the original input to its output through a designated bottleneck layer **h**. As illustrated in the figure, the input and output of the CAE are the traffic state matrices with a shape of $64 \times 64 \times 2$. The encoder part consists of two pairs of convolution-pooling layers followed by two fully-connected layers, while the decoder is the reverse of the encoder structure. The notation numbers define the dimensions of the outputs at each respective layer. The size of hidden layer **h** can be determined through training experiments. Upon completion of training, the encoder network will be used as the state representation compressor to generate the input vector to the DRL neural network.

The proposed CAE was implemented and trained using Tensorflow [48]. The optimization process employs the Adam algorithm [49] with mean squared error (MSE) as the cost function. The hyper-parameters, such as number of filters and neurons in the CAE, were determined through cross-validation training experiments. The size of the reduced representation vector of the traffic states was selected as 8, resulting in a compression ratio of 1024 to 1. In addition, the input state matrix was normalized by scaling the vehicle speed to the range of 0 to 1 based on the maximum allowable speed of the road.

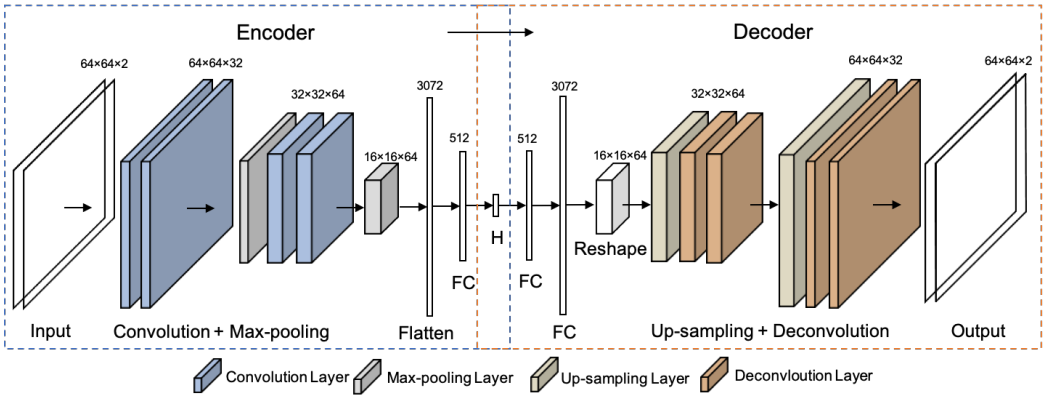


Figure 5. The structure of the proposed convolutional autoencoder (CAE) for traffic state representation.

The proposed method was evaluated using a validation set, which constituted 10% of the total available dataset. The simulation of the entire traffic signal control system was carried out in SUMO, and the training dataset was generated by continuously running simulations in SUMO. All vehicles were randomly initialized with a specified flow rate. An example training session is shown in Figure 6 using a dataset of one million samples. The minimum reconstruction errors in this example are 5.07×10^{-4} for the training error and 6.54×10^{-4} for the validation error. It is important to note that the purpose of using CAE is not only to reduce the dimension of the traffic state but also to extract inherent features within the traffic information. The original traffic state is image-like data, and convolutional neural networks are usually used to learn the hierarchical representations of these visual data. Based on convolutional operations, CAE is capable of learning inherent features associated with the geometric distribution of the vehicles around the intersection. Hence, CAE was selected in this work. However, the effectiveness of CAE should not be solely judged by its ability to perfectly reconstruct the input sample but validated through the RL control experiments. An increase in the size of the hidden layer **H** may lower the reconstruction loss; however, a large input size for the RL algorithm may have negative consequences.

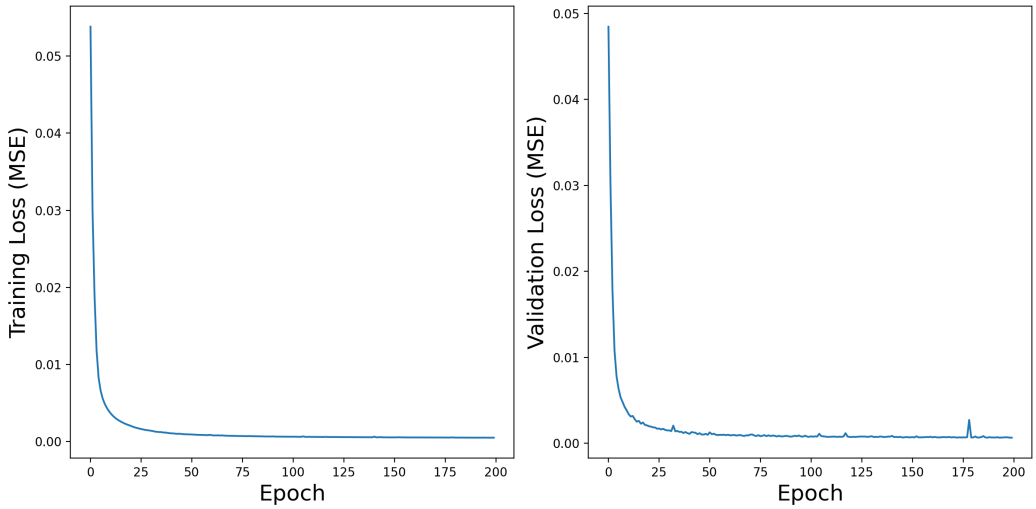


Figure 6. Training history of the proposed convolutional autoencoder.

3.3. Deep Reinforcement Learning Structure

The well-trained CAE is then combined with the DRL algorithm to form the final model. The overall structure of the model is shown in Figure 7. The CAE's encoder network generates a compressed representation of traffic states, which is then fed into the fully

connected neural network to approximate the Q-value function as described in [4]. At the end of each control cycle, the neural network computes the Q-values of all available actions for the given traffic state representation. The agent then selects the action with the highest Q-value, which is expected to result in the maximum reward. After executing the selected action, a new control cycle begins and the agent continues to learn how to maximize rewards through interaction with the environment. To improve learning efficiency and reduce possible over-estimations, the model incorporates techniques such as the target network [5], dueling network [6], and prioritized experience replay [7]. The proposed DRL training algorithm is detailed in Algorithm 1.

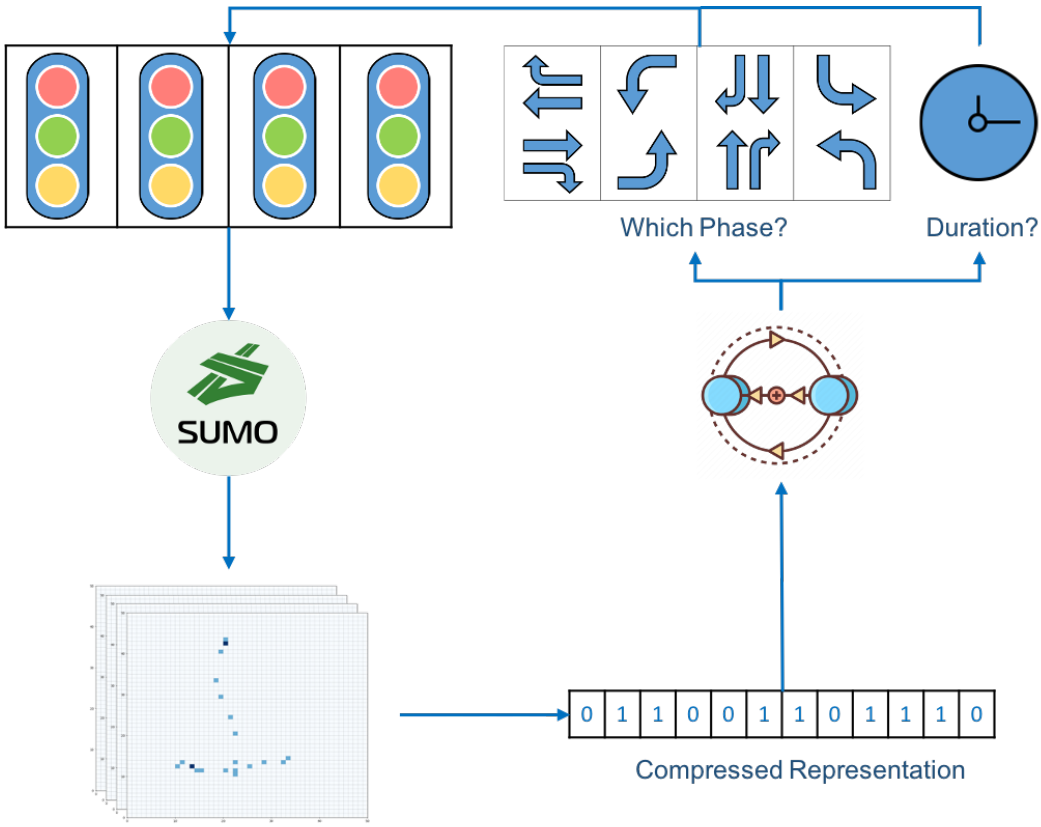


Figure 7. Proposed deep reinforcement learning model.

Reward Signal: The design of the reward signal is critical to the success of the RL-based traffic signal control. The reward signal serves as a guide for the agent to learn the objective of its control actions, which is to enhance the intersection's throughput and minimize vehicle waiting time. The reward provides feedback to the agent, evaluating its past actions. It's important to note that unlike other RL-based applications, the traffic signal control problem has no terminal state. This means that the reward signal must reflect the performance of each action taken by the agent, as there is no terminal reward to learn from.

There is no deterministic guideline for selecting the most appropriate performance index, but the selection is crucial, as it guides the agent in learning the desired control objectives. Commonly used performance indices in the field of traffic signal control include travel delay, queue length, and average vehicle speed (as seen in [8,17,50]). Some approaches focus on reducing road congestion and only use average waiting time or queue length as the reward signal. However, this could result in the agent learning a strategy that frequently changes the traffic signal, leading to shorter queue time but lower speed. On the other hand, using a shorter cycle length reduces average vehicle speed, leading to vehicles spending more time idling at intersections, which not only increases fuel consumption but also contributes to the reduction in road capacity.

Algorithm 1: Pseudo-code for training algorithm of DRL-based agent

Input: mini batch size B, pre-train step t_p, training episode length N, learning rate α, greedy ϵ, discount factor γ, target network update rate τ, target network update frequency K

Initialize primary network Q_θ, target network Q_{θ^-}, replay memory D with capacity M

Loop for each episode:
- Initialize simulator environment
- Initialize time step $t = 0$
- Observe current state S_t
- **while** time step $t < N$:
  - With probability ϵ select action A_t randomly
  - otherwise select $A_t \longleftarrow argmax Q_\theta(S_t, a)$
  - Execute action then observe next state S_{t+1} and reward R_t
  - Store (S_t, A_t, R_t, S_{t+1}) in replay memory D
  - $S_t \longleftarrow S_{t+1}$
  - **if** current step $t >$ pre-training step t_p:
    - Sample a minibatch of B experience tuples (S_t, A_t, R_t, S_{t+1}) from D
    - Compute target Q values for each experience:

$Q^*(S_t, A_t) \approx R_t + \gamma Q_{\theta^-}(S_{t+1}, argmax_{a'} Q_\theta(S_{t+1}, a'))$

    - Perform a gradient descent step with loss:

$\frac{1}{B}\|Q^*(S_t, A_t) - Q_\theta(S_t, A_t)\|^2$

  - Update target network θ^- every K steps:

$\theta^- \longleftarrow \tau\theta + (1-\tau)\theta^-$

  - $t \longleftarrow t + 1$

A better approach is to use average vehicle speed as the reward signal. This allows the agent to improve the overall traffic mobility and reduce average travel delay, while also promoting fuel efficiency, which is mainly related to the vehicle speed and idle time. Therefore, we use the average vehicle speed $\bar{V}$ of the entire intersection, calculated at the end of each control cycle, as the reward signal R, which is defined below:

$$R = \bar{V} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} v_i, \tag{1}$$

where v_i is the velocity of vehicle i, T is the length of the current signal cycle, N is the total number of vehicles in the control zone, and t is the time-step of the simulation.

4. Simulation Results

In this section, we present the simulated experiments to examine the effectiveness of the proposed methodology.

4.1. Simulation Parameters

The simulation took place in a SUMO environment where a 320 m $\times$ 320 m intersection was considered and established (as shown in Figure 2). The detailed parameters of the simulated intersection and vehicles are listed in Table 1. The vehicles were randomly initialized with a 10% probability per second. The Krauss car-following model [51] was employed to ensure that vehicles move as fast as possible while maintaining perfect safety requirements. The simulations assume a 100% CV penetration rate. Further evaluation of other penetration levels will be conducted in future work.

Table 1. Adopted parameters of simulation environment.

Parameter	Value
Lane length	160 m
Vehicle length	5 m
Time step	1 s
Maximum vehicle speed	20 (m/s)
Maximum vehicle acceleration	3 (m/s^2)
Maximum vehicle deceleration	4.5 (m/s^2)
Minimum gap between vehicles	2 m
Car following model	Krauss Following Model [51]
Duration of yellow phase	3 s
Traffic volume	480 vehicles per lane and per hour
Left turning vehicles ratio	25% of total
Right turning vehicles ratio	25% of total

4.2. Hyper-Parameter of Deep Reinforcement Learning Network

The implementation of the DRL network was carried out using Tensorflow [48] and integrated with the SUMO simulation environment through a Python interface. The training was conducted in episodes, with each episode consisting of 3600 time steps of 1 s each, totaling one hour per episode. The random seed for the vehicle simulation was changed in every episode. The critical hyper-parameters are listed in Table 2, with the values determined through a process of trial and error.

Table 2. Hyper-parameters of deep reinforcement learning network.

Parameter	Value
Simulated time steps for each episode	3600
Replay memory size	20,000
Minibatch size	64
Pre-train steps	2000
Target network update interval	64 control cycles
Target network update rate	0.001
Discount factor	0.99
Optimizer	Adam [49]
Learning rate	1×10^{-4}
Initial probability of exploration	1
Final probability of exploration	0.01
Ending step for exploration probability	40,000

4.3. Convergence of DRL-Based Signal Controller Training

The convergence of the proposed DRL training algorithm was demonstrated by evaluating the accumulated rewards for each episode. As shown in Figure 8, the rewards increased rapidly at the beginning and then leveled off as the training progressed. The average vehicle speed and average waiting time in each episode are also plotted to show the improvement and convergence of traffic measurements. As previously noted, the average waiting time is not part of the optimization objective due to its conflicting relationship with the average vehicle speed in signal control policy. As a result, the average waiting time increases slightly at the end of the training process.

4.4. Comparison with Baselines

The proposed method was compared with existing methods by implementing the DRL-based traffic signal controller in [28] under the same simulation conditions. The DRL training algorithm used is similar to the one described in Algorithm 1. It is important

to note that the reward signal in [28] is the average waiting time and their signal control strategy involves adding or subtracting 5 s from the duration of one of the current phases. As shown in Figure 9, the performance of both average waiting time and average vehicle speed improve as the training process progresses, but its convergence is slower and more fluctuated compared to our proposed method in this paper.

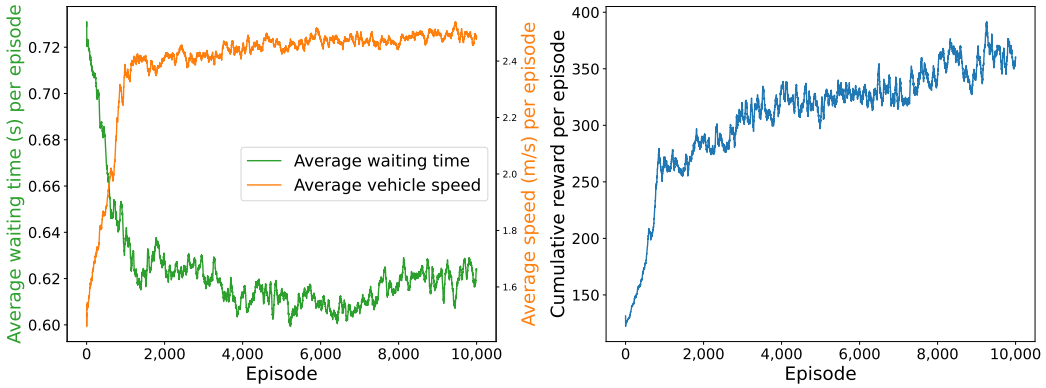


Figure 8. Convergence of the proposed DRL network.

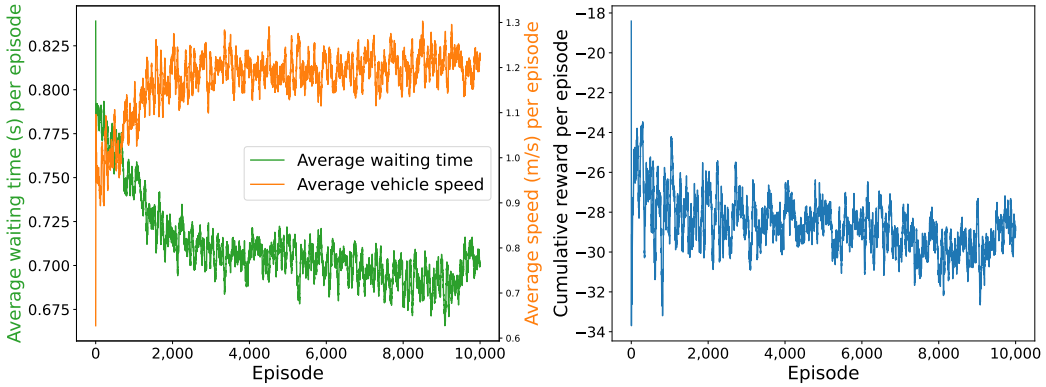


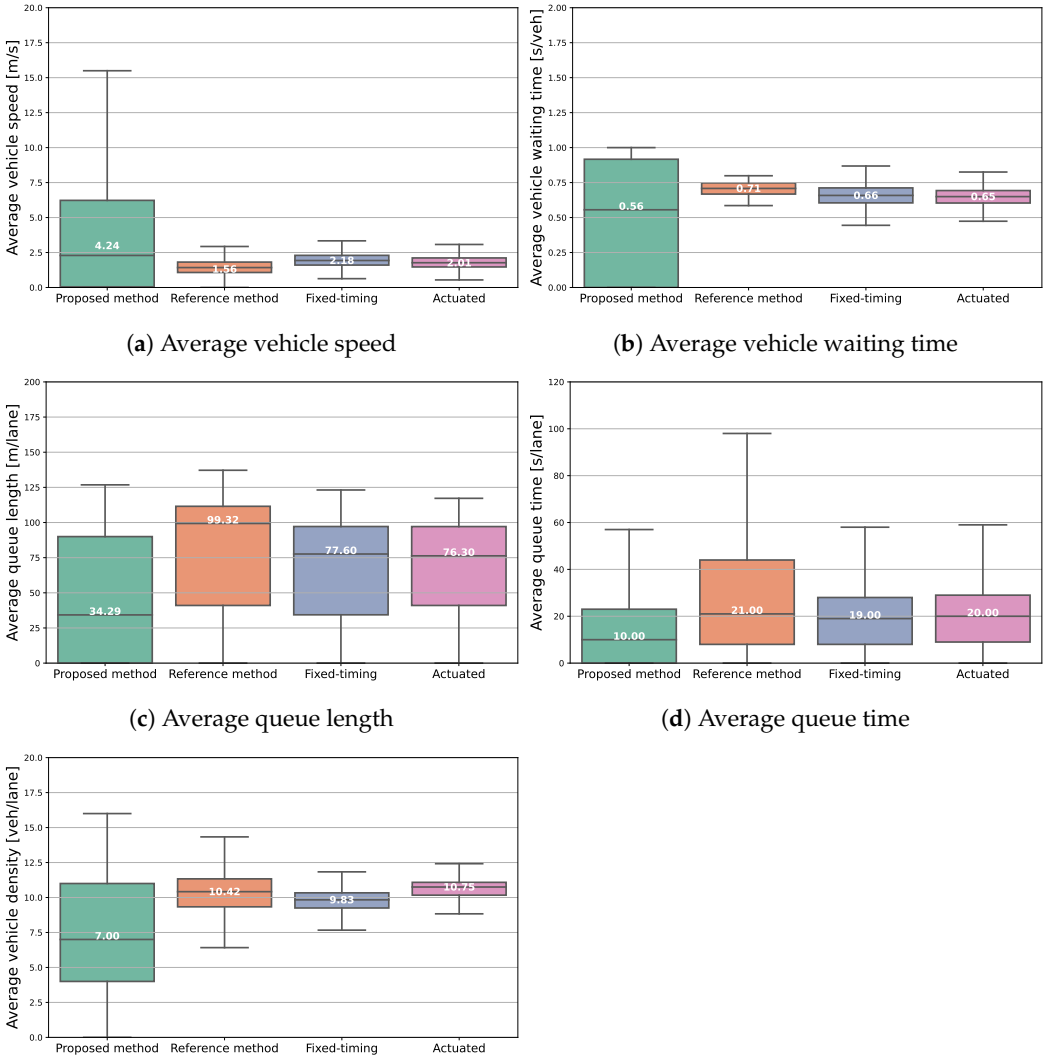
Figure 9. Training history of a reference DRL-based traffic signal controller.

In addition to the proposed DRL-based traffic signal controller, two baseline methods have been implemented for comparison. These include a fixed-timing signal controller and an actuated signal controller. To determine the best fixed-timing strategy, various combinations of cycle lengths and phase duration were explored and the one with the best performance was selected. The selected cycle length was 60 s, and the four phase duration were 20 s, 10 s, 20 s, and 10 s. The actuated signal controller operates on a time-gap basis, allowing the green phase to be extended if there is a continuous flow of traffic. When the time gap between successive vehicles satisfies a predefined criterion, the signal switches to the next phase. Actuated controllers are known to perform better than fixed-timing controllers in dynamic traffic conditions. However, the traffic flow was steady and thus the actuated signal controller performed similarly to the fix-timing controller. In addition, the maximum phase duration was set to match the fix-timing controller, while the minimum phase duration was set at 5 s.

To evaluate the performance of the proposed DRL-based control method, five common traffic mobility metrics were selected: (1) average vehicle speed, which reflects the overall mobility of the intersection, in the present moment and over a certain period of time, and represents the average travel time of all vehicles to complete the trip when computed over an episode; (2) average waiting time, which is calculated by dividing the total waiting time of vehicles by the number of vehicles present at each time step, providing an insight into travel delays from the perspective of road users; (3) average queue length, which is the average of the total number of lanes and measures the congestion level of the intersection

at each time step; (4) average queue time, which is calculated by dividing the total waiting time of vehicles caused by a queue by the total number of lanes at each time step and offers a similar perspective as average waiting time, but with a focus on traffic congestion; (5) average vehicle density, which is the average number of vehicles in each lane at each time step and represents the density of vehicles approaching the intersection.

The performance comparisons of our proposed method with the baseline methods is shown in Figure 10. These metrics were obtained through 100 repetitions of statistical testing using the same vehicle initialization file. Each data point represents the traffic state at a single time step of the simulations and the median values are indicated by the white labels in the middle of the boxes. As can be seen from this figure, the wide distribution of the average vehicle speed and the average vehicle waiting time is due to the stochastic nature of the traffic flow, while the lower median values indicate that our DRL-based controller can acquire the signal control policies that may prioritize the overall traffic performance over the performance of individual vehicles, which is also verified in Figure 13 that will be discussed later. However, there is room for further improvement of the control policies by tuning the learning algorithm, such as by adding a maximum constraint for the vehicle's waiting time, which will be explored in future research.



(**a**) Average vehicle speed

(**b**) Average vehicle waiting time

(**c**) Average queue length

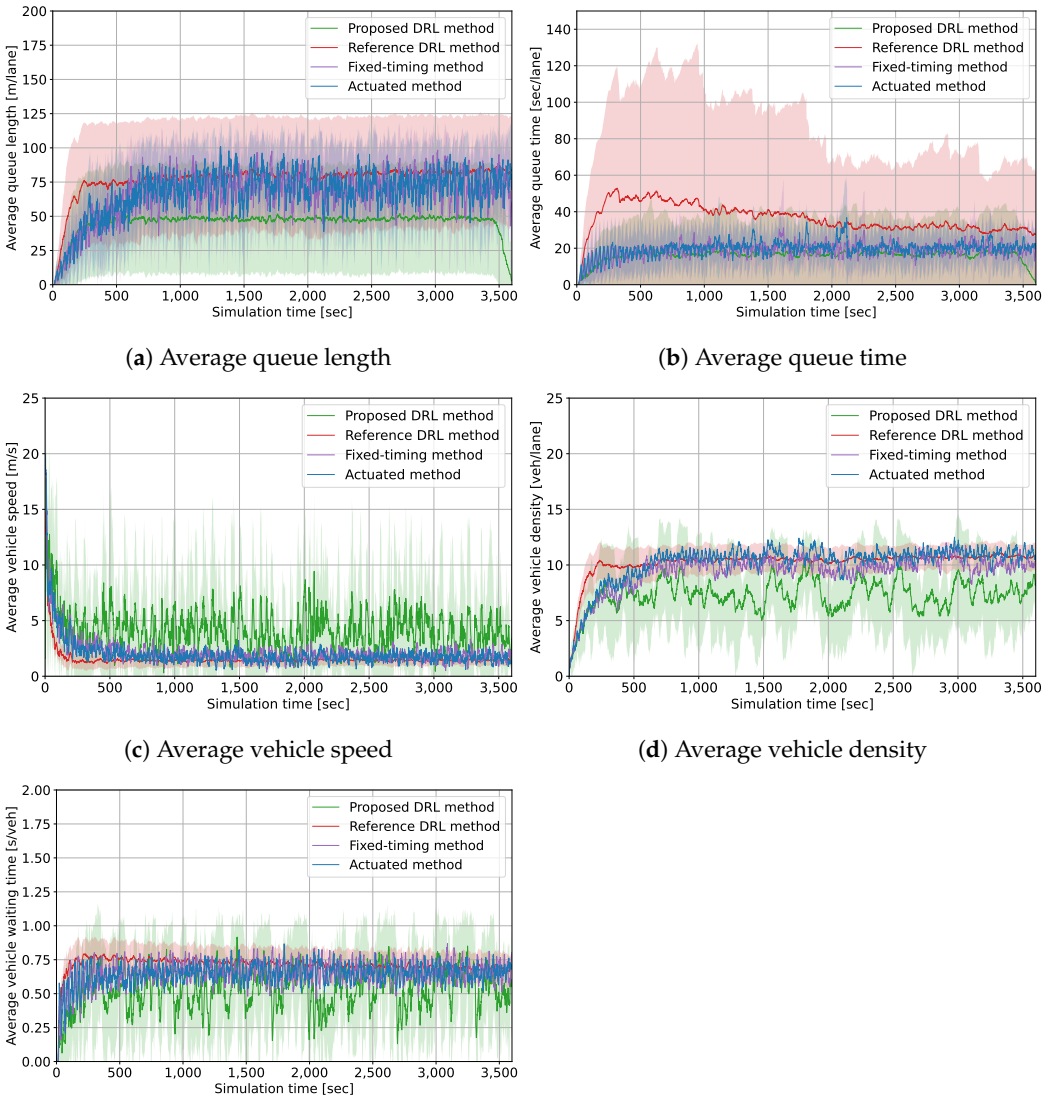
(**d**) Average queue time

(**e**) Average vehicle density

Figure 10. Performance comparisons with baselines.

The performance of the proposed DRL-based control method is also demonstrated in Figure 11 through the five common traffic mobility metrics considered, each with its

average value and confidence interval in a time series. Comparing Figures 11a–e, it can be seen that the curves have similar trends, implying that these metrics are interconnected. For example, longer queues lead to longer average queue time, slower average vehicle speed, and slower vehicle discharge. Specifically, Figure 11a shows that the proposed method had a much shorter average queue length than the three baselines. The curve of our proposed method increased at the beginning and then decreased sharply at the end, whereas the other methods did not follow this trend, demonstrating the effectiveness of the proposed controller in recovering from saturation and clearing the intersection quickly when the traffic volume decreases. By contrast, the other methods could not fully recover from congestion within the simulation time. Regarding the average queue time in Figure 11b, our proposed DRL method resulted in smaller queue times than the baseline DRL method while there seems to have been no obvious improvement over the fixed-timing and actuated methods. This is due to the acquired asymmetric policy, which sacrificed traffic flow in the north and south approaches to maximize the overall performance. If we continue to examine Figure 11c–e, we can see that our developed DRL method can lead to slightly better results (e.g., higher average vehicle speed, lower average vehicle density, and shorter average vehicle waiting time) than the three baseline methods under the simulation scenario considered in this work.



(**a**) Average queue length

(**b**) Average queue time

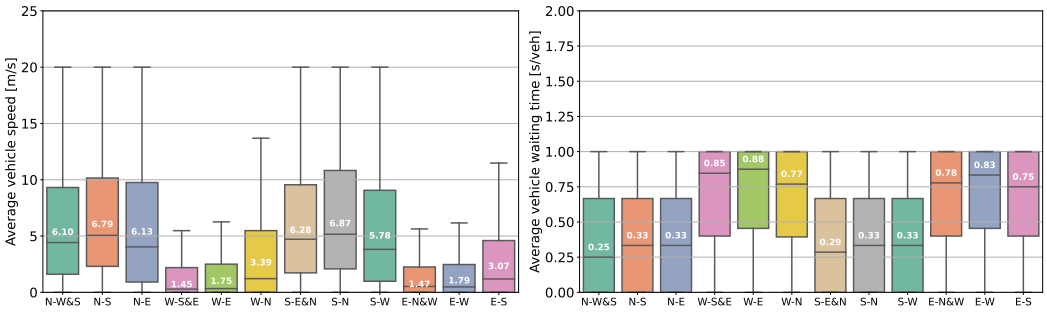
(**c**) Average vehicle speed

(**d**) Average vehicle density

(**e**) Average vehicle waiting time

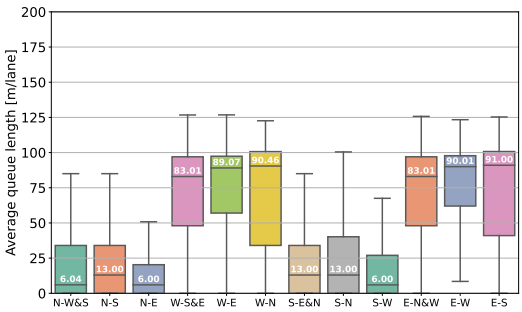
Figure 11. Simulation comparisons with baselines.

To better understand the reason behind the wide distribution of the mobility metrics simulated by the proposed method, the lane-wise simulation data of our proposed DRL method is plotted in Figures 12 and 13. The comparison of the average queue length and queue time of each lane reveals that the lanes in the north and south approaches have similar curves with lower values than the west and east lanes. This suggests that our DRL-based controller has adopted an asymmetric traffic flow policy, even though the traffic volumes are balanced, to optimize the performance of the entire intersection. Figure 12 further confirms that the significant deviations observed in the green boxes of Figure 10 are primarily due to the asymmetric traffic control policy. Specifically, the north and south approaches always have queues, while the west and east approaches have smoother traffic flow. As a comparison, the traffic flows simulated by other considered methods are plotted in Figure 14. It can be seen that their traffic flows are balanced, as evidenced by the similar trends and values of the average queue length and queue time for each lane.

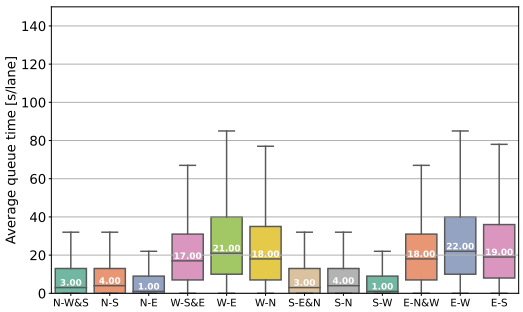


(**a**) Average vehicle speed

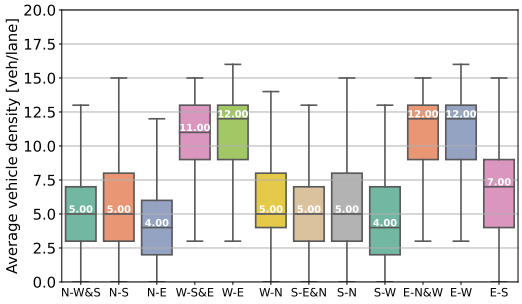
(**b**) Average vehicle waiting time



(**c**) Average queue length

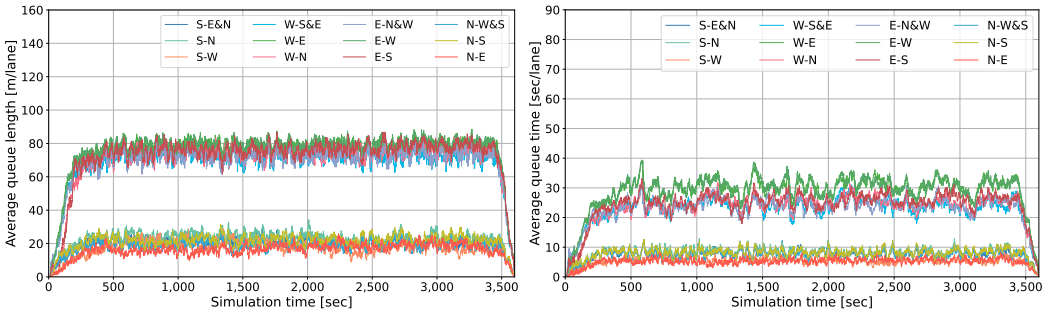


(**d**) Average queue time



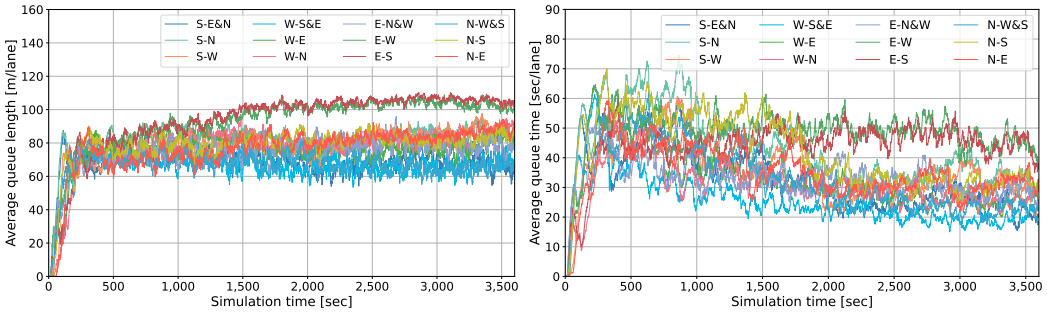
(**e**) Average vehicle density

Figure 12. Performance estimation of the proposed method in each lane.



(**a**) Average queue length (**b**) Average queue time

Figure 13. Traffic flow simulated by the proposed method in each lane.



(**a**) Average queue length (**b**) Average queue time

Figure 14. Traffic flow simulated by the referenced method in each lane.

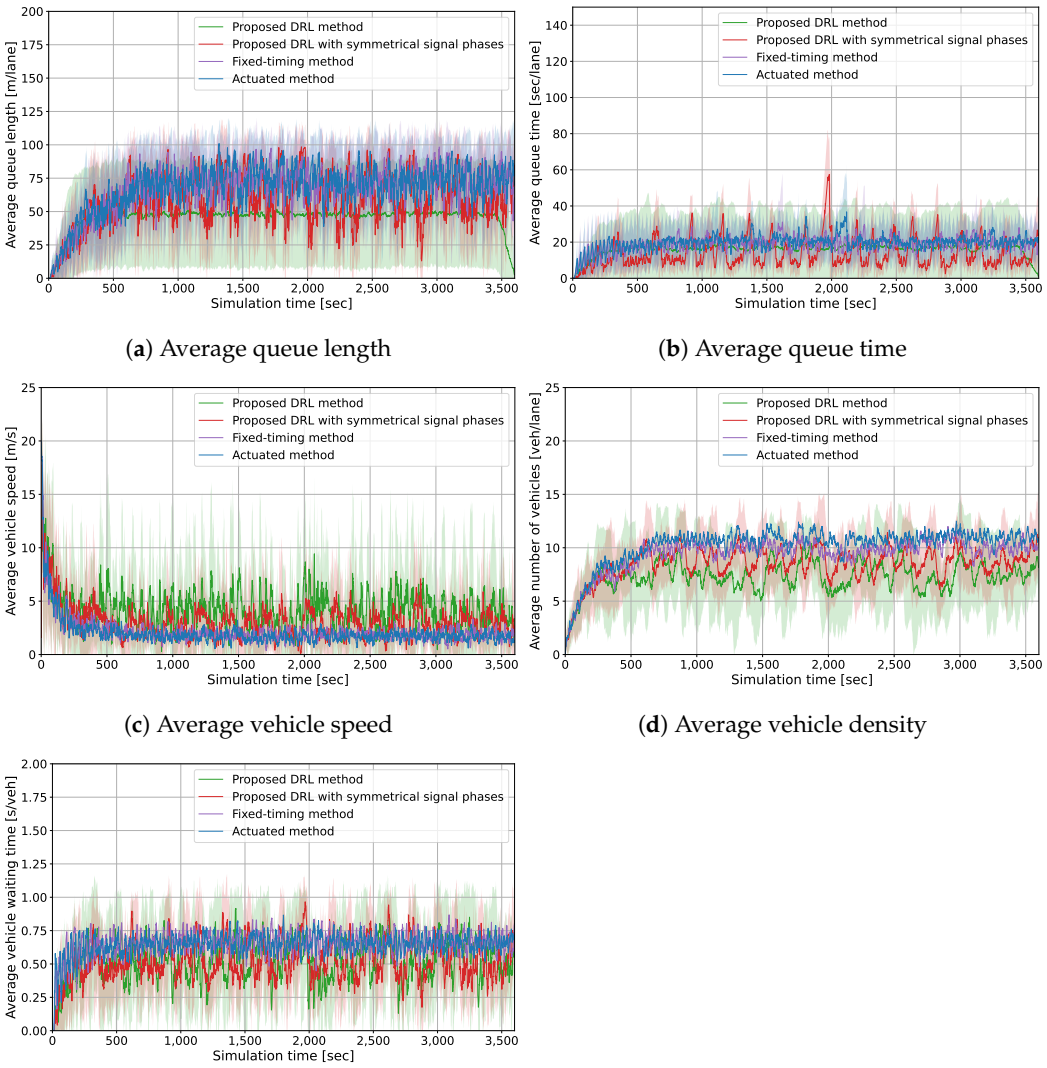
To further validate the control policy acquired by our proposed method, we trained a DRL controller with symmetric signal phases and timing. Specifically, each signal cycle had equal lengths for the first and third phases, while the second and fourth phases were identical. The cycle length was selected from the set {10 s, 20 s, 30 s, 40 s, 50 s, 60s}, resulting in 27 total actions. The purpose of this experiment was to see the control policy learned by the DRL algorithm with symmetric SPaT. As shown in Figure 15, the results for the symmetric policy have more noticeable periodic oscillations, indicating periodic traffic congestion. However, despite not being as good as the original DRL-based controller with flexible SPaT, it still outperformed the fix-timing controller. There is no significant difference in traffic flow between each direction for the symmetric SPaT policy; hence a direction-wise plot was not included. This comparison confirms our speculation that the original DRL controller had learned a policy with asymmetric traffic flows to optimize the traffic performance of the entire intersection.

4.5. Robustness Analysis

The challenge in data-driven control methods is ensuring robustness and reliability with discrete data points. On the one hand, to achieve the optimal control in a complex environment, the agent must visit each state–action pair enough times, which may result in over-fitting, i.e., poor control performance under the unseen traffic conditions. On the other hand, to perform robustly on unseen states, the solution is either to train the agent with a large dataset including as many state–action variations as possible or to simplify the state and action space, but both approaches may compromise convergence and optimality of the controller. As stated in previous sections, our solution tackles this challenge by using the CAE to reduce the dimensionality of the state space while increasing the action space to enhance the control performance.

The previous simulations were conducted with a constant traffic flow rate of 480 vehicles per hour per lane. To examine the robustness of the proposed DRL-based method, the controller was evaluated under different traffic volumes with varying traffic flow rates. The results of the previously trained controller with volumes of 400, 600, 720, and 800 vehicles per hour per lane are shown in the green box-plots in Figure 16. The orange box-

plots represent the performance of controllers specifically trained with the corresponding traffic flow rates. For instance, the orange box-plots of 400 vehicles per hour per lane show the simulation data generated by a DRL-based controller trained with a constant traffic flow of 400 vehicles per hour per lane. The fixed-time and actuated controllers are also included for comparison. The comparison shows that the proposed DRL-based controller performs well in unseen traffic scenarios. However, retraining the controller with specific volumes does result in improved performance, but at a higher training cost and without guaranteed results. The control policy depends on the traffic volume, but the agent cannot determine the traffic volume from a single control cycle's traffic state. To overcome this limitation, future investigation could focus on incorporating temporal information into the traffic state inputs to achieve optimal performance in various traffic scenarios.



(**a**) Average queue length

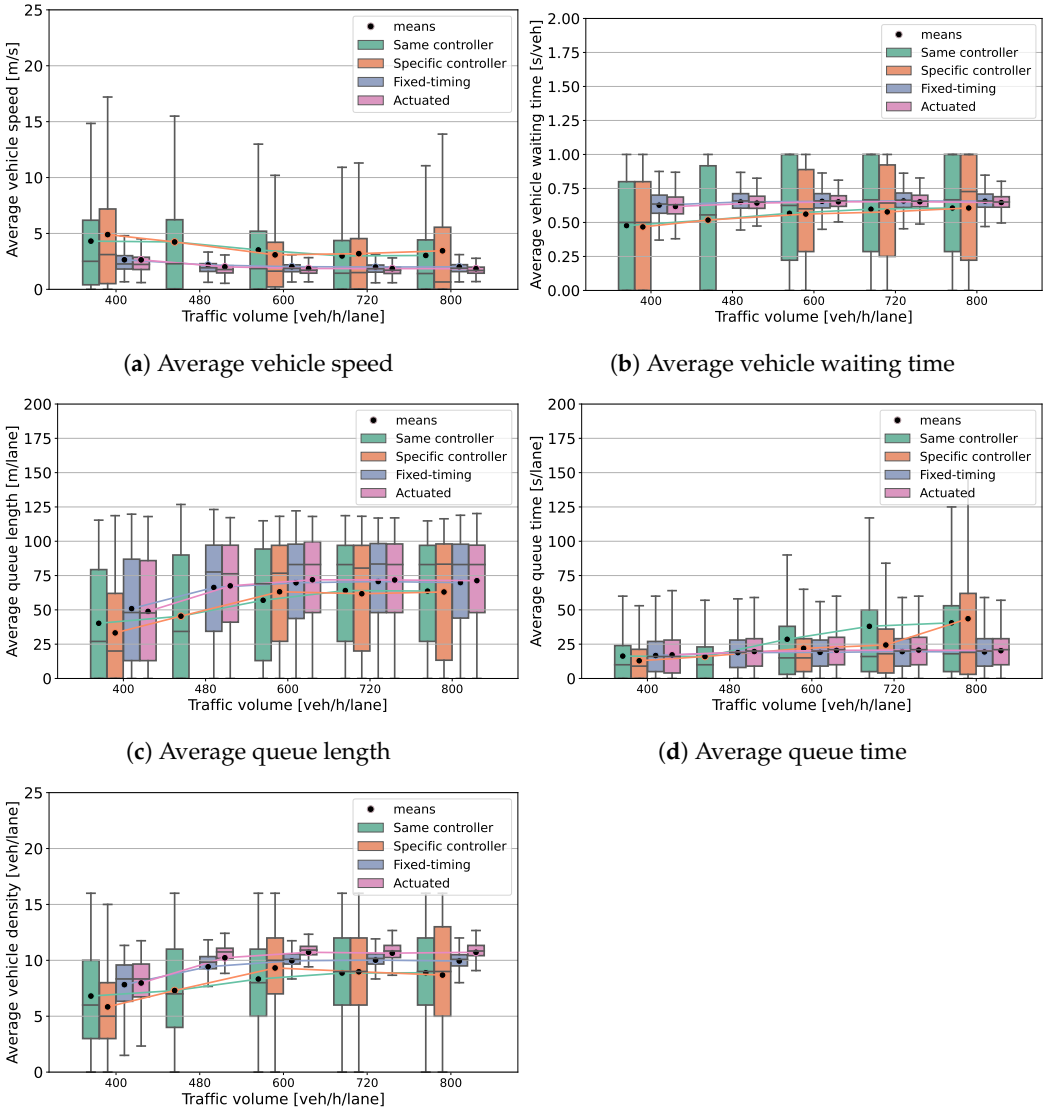
(**b**) Average queue time

(**c**) Average vehicle speed

(**d**) Average vehicle density

(**e**) Average vehicle waiting time

Figure 15. Simulation comparisons with the DRL-based controller trained using symmetric SPaT.



(**a**) Average vehicle speed

(**b**) Average vehicle waiting time

(**c**) Average queue length

(**d**) Average queue time

(**e**) Average vehicle density

Figure 16. Robustness analysis by comparing the performance of DRL-based controller.

5. Conclusions

This paper presents a novel DRL-based traffic signal controller for a typical four-way intersection. Different from the existing data-driven methods, this work takes advantage of CAE to capture traffic states into compressed representations. With a reduced dimensionality of the input traffic states, the proposed DRL-based controller enables more flexible design of the action space and increased responsiveness to dynamic traffic conditions. The simulation results demonstrate the effectiveness and performance of the proposed method through comparisons with three baseline methods across five commonly used performance metrics. The proposed DRL-based controller also exhibits more consistent training results compared to existing DRL methods. To further validate the control policy learned by the proposed DRL algorithm, the traffic flows with different SPaT plans are analyzed. In addition, the proposed DRL-based controller is tested for robustness against varying traffic volumes and compared with controllers retrained for specific traffic conditions. The results indicate that the proposed DRL agent is capable of handling unseen traffic scenarios effectively.

The proposed method has a limitation in that it incurs a high training cost due to the expanded action space. Additionally, it has only been tested on a single four-phase intersection with 100% CV penetration rate. Future work will aim to extend the proposed method

to more complex scenarios and scale it up for corridor and network-level signal control with joint optimization of signal timing. This will present a greater challenge, as the traffic state and action space dimensions will increase exponentially. One potential solution to this challenge is to utilize the multi-agent reinforcement learning approach, which addresses the control problem of multiple autonomous, interactive agents in a common environment by distributing the global control to multiple local RL control agents [52]. The sharing of information among intersections can help individual signal controllers to learn and work together to optimize the overall performance of the traffic network. Additionally, we aim to expand the scenario to include varying traffic flows and heterogeneous vehicles, to further test the robustness and scalability of the data-driven methods. By incorporating these challenges, we hope to develop a data-driven approach that can learn a practical and truly optimal signal control strategy for real-world traffic systems.

Furthermore, the proposed method uses the position and speed information of CVs at the end of a control cycle to construct the traffic state matrix as input to the DRL controller, ignoring the temporal information. Utilizing recurrent neural networks, such as long short-term memory (LSTM), has the potential to capture the complex dynamics within the temporal information of input data. Integrating an LSTM-autoencoder into the encoder–decoder network architecture can help it learn a representation for time series sequence data, enabling the DRL controller to make more accurate traffic state estimations and improve control strategies. These are promising avenues for future research.

Author Contributions: Conceptualization, Y.S. (Yang Shi), C.W., Y.S. (Yunli Shao) and J.Y.; Funding acquisition, Z.W.; Investigation, Y.S. (Yang Shi); Methodology, Y.S. (Yang Shi); Project administration, Z.W. and T.J.L.; Supervision, Z.W.; Validation, Y.S. (Yang Shi); Writing—original draft, Y.S. (Yang Shi); Writing—review and editing, Z.W., T.J.L., C.W., Y.S. (Yunli Shao) and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Support for Affiliated Research Teams (StART) program at the University of Tennessee Knoxville and was completed through a partnership between the Department of Mechanical, Aerospace, and Biomedical Engineering at the University of Tennessee Knoxville and the Buildings and Transportation Science Division at the Oak Ridge National Laboratory. The APC was funded by the University of Tennessee Knoxville.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Aziz, H.A.; Wang, H.; Young, S.; Sperling, J.; Beck, J.M. *Synthesis Study on Transitions in Signal Infrastructure and Control Algorithms for Connected and Automated Transportation*; Technical Report; Oak Ridge National Laboratory Report, ORNL/TM-2017/280: Oak Ridge, TN, USA, 2017.
2. Wünsch, G. Coordination of Traffic Signals in Networks. Ph.D. Thesis, Technische Universität Berlin, Berlin, Germany, 2008.
3. Al Islam, S.B.; Hajbabaie, A. Distributed coordinated signal timing optimization in connected transportation networks. *Transp. Res. Part C Emerg. Technol.* **2017**, *80*, 272–285.
4. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
5. Van Hasselt, H.; Guez, A.; Silver, D. Deep reinforcement learning with double q-learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
6. Wang, Z.; Schaul, T.; Hessel, M.; Van Hasselt, H.; Lanctot, M.; De Freitas, N. Dueling network architectures for deep reinforcement learning. *arXiv* **2015**, arXiv:1511.06581.
7. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
8. Guo, Q.; Li, L.; Ban, X.J. Urban traffic signal control with connected and automated vehicles: A survey. *Transp. Res. Part C Emerg. Technol.* **2019**, *101*, 313–334.
9. He, Q.; Head, K.L.; Ding, J. PAMSCOD: Platoon-based arterial multi-modal signal control with online data. *Transp. Res. Part C Emerg. Technol.* **2012**, *20*, 164–184.

10. Feng, Y.; Head, K.L.; Khoshmagham, S.; Zamanipour, M. A real-time adaptive signal control in a connected vehicle environment. *Transp. Res. Part C Emerg. Technol.* **2015**, *55*, 460–473.
11. Zhao, J.; Li, W.; Wang, J.; Ban, X. Dynamic traffic signal timing optimization strategy incorporating various vehicle fuel consumption characteristics. *IEEE Trans. Veh. Technol.* **2015**, *65*, 3874–3887.
12. Ma, C.; Liu, P. Intersection signal timing optimization considering the travel safety of the elderly. *Adv. Mech. Eng.* **2019**, *11*, 1687814019897216. https://doi.org/10.1177/1687814019897216.
13. Mohebifard, R.; Hajbabaie, A. Optimal network-level traffic signal control: A benders decomposition-based solution algorithm. *Transp. Res. Part B Methodol.* **2019**, *121*, 252–274. https://doi.org/10.1016/j.trb.2019.01.012.
14. Daganzo, C.F. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transp. Res. Part B Methodol.* **1994**, *28*, 269–287. https://doi.org/10.1016/0191-2615(94)90002-7.
15. BnnoBRs, J. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **1962**, *4*, 238–252.
16. Bin Al Islam, S.M.A.; Abdul Aziz, H.M.; Hajbabaie, A. Stochastic Gradient-Based Optimal Signal Control with Energy Consumption Bounds. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3054–3067. https://doi.org/10.1109/TITS.2020.2979384.
17. Hong, W.; Tao, G.; Wang, H.; Wang, C. Traffic Signal Control With Adaptive Online-Learning Scheme Using Multiple-Model Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–13. https://doi.org/10.1109/TNNLS.2022.3146811.
18. Li, W.; Ban, X.J. Traffic signal timing optimization in connected vehicles environment. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1330–1335.
19. Goodall, N.J.; Smith, B.L.; Park, B. Traffic signal control with connected vehicles. *Transp. Res. Rec.* **2013**, *2381*, 65–72.
20. Noaeen, M.; Mohajerpoor, R.; H. Far, B.; Ramezani, M. Real-time decentralized traffic signal control for congested urban networks considering queue spillbacks. *Transp. Res. Part C Emerg. Technol.* **2021**, *133*, 103407. https://doi.org/10.1016/j.trc.2021.103407.
21. Islam, S.B.A.; Hajbabaie, A.; Aziz, H.A. A real-time network-level traffic signal control methodology with partial connected vehicle information. *Transp. Res. Part C Emerg. Technol.* **2020**, *121*, 102830. https://doi.org/10.1016/j.trc.2020.102830.
22. Liang, X.J.; Guler, S.I.; Gayah, V.V. An equitable traffic signal control scheme at isolated signalized intersections using Connected Vehicle technology. *Transp. Res. Part C Emerg. Technol.* **2020**, *110*, 81–97.
23. Beak, B.; Head, K.L.; Feng, Y. Adaptive coordination based on connected vehicle technology. *Transp. Res. Rec.* **2017**, *2619*, 1–12.
24. Qiao, Z.; Ke, L.; Wang, X.; Lu, X. Signal Control of Urban Traffic Network Based on Multi-Agent Architecture and Fireworks Algorithm. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 2199–2206.
25. Tan, Y.; Yu, C.; Zheng, S.; Ding, K. Introduction to fireworks algorithm. *Int. J. Swarm Intell. Res. (IJSIR)* **2013**, *4*, 39–70.
26. Ma, C.; Zhou, J.; Xu, X.D.; Xu, J. Evolution regularity mining and gating control method of urban recurrent traffic congestion: A literature review. *J. Adv. Transp.* **2020**, *2020*, 5261580.
27. Bala Subramaniyan, A.; Wang, C.; Shao, Y.; Li, W.; Wang, H.; Guohui, Z.; Ma, T. Hybrid Recurrent Neural Network Modeling for Traffic Delay Prediction at Signalized Intersections Along an Urban Arterial. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1384–1394.
28. Liang, X.; Du, X.; Wang, G.; Han, Z. A deep reinforcement learning network for traffic light cycle control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1243–1253.
29. Ma, C.; Zhao, Y.; Dai, G.; Xu, X.; Wong, S.C. A Novel STFSA-CNN-GRU Hybrid Model for Short-Term Traffic Speed Prediction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *2022*, 1–10. https://doi.org/10.1109/TITS.2021.3117835.
30. Al Islam, S.B.; Aziz, H.A.; Wang, H.; Young, S.E. Minimizing energy consumption from connected signalized intersections by reinforcement learning. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 1870–1875.
31. Du, Y.; ShangGuan, W.; Rong, D.; Chai, L. RA-TSC: Learning Adaptive Traffic Signal Control Strategy via Deep Reinforcement Learning. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3275–3280.
32. Chen, P.; Zhu, Z.; Lu, G. An Adaptive Control Method for Arterial Signal Coordination Based on Deep Reinforcement Learning. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3553–3558.
33. Yoon, J.; Ahn, K.; Park, J.; Yeo, H. Transferable traffic signal control: Reinforcement learning with graph centric state representation. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103321.
34. Zeng, J.; Hu, J.; Zhang, Y. Training Reinforcement Learning Agent for Traffic Signal Control under Different Traffic Conditions. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019, pp. 4248–4254.
35. Aslani, M.; Mesgari, M.S.; Wiering, M. Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transp. Res. Part C Emerg. Technol.* **2017**, *85*, 732–752. https://doi.org/10.1016/j.trc.2017.09.020.
36. Chu, T.; Wang, J.; Codecà, L.; Li, Z. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1086–1095.
37. Li, Z.; Yu, H.; Zhang, G.; Dong, S.; Xu, C.Z. Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103059.
38. Wang, T.; Cao, J.; Hussain, A. Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103046.

39. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354.
40. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; WieBner, E. Microscopic traffic simulation using sumo. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
41. Genders, W.; Razavi, S. Using a deep reinforcement learning agent for traffic signal control. *arXiv* **2016**, arXiv:1611.01142.
42. Jeong, J.; Shen, Y.; Oh, T.; Céspedes, S.; Benamar, N.; Wetterwald, M.; Härri, J. A comprehensive survey on vehicular networks for smart roads: A focus on IP-based approaches. *Veh. Commun.* **2021**, *29*, 100334. https://doi.org/10.1016/j.vehcom.2021.100334.
43. Transportation Research Board and National Academies of Sciences, Engineering, and Medicine. In *Signal Timing Manual*, 2nd ed.; The National Academies Press: Washington, DC, USA, 2015.
44. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. In *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning*; Guyon, I., Dror, G., Lemaire, V., Taylor, G., Silver, D., Eds.; Proceedings of Machine Learning Research; PMLR: Bellevue, DC, USA, 2012; Volume 27, pp. 37–49.
45. Hakenes, S.; Glasmachers, T. Boosting Reinforcement Learning with Unsupervised Feature Extraction. In *Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2019: Theoretical Neural Computation*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 555–566.
46. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
47. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
48. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Available online: tensorflow.org (accessed on 31 January 2022).
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Wang, H.; Zhu, M.; Hong, W.; Wang, C.; Tao, G.; Wang, Y. Optimizing Signal Timing Control for Large Urban Traffic Networks Using an Adaptive Linear Quadratic Regulator Control Strategy. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 333–343. https://doi.org/10.1109/TITS.2020.3010725.
51. Krauß, S. Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics. Master's Thesis, Universität zu Köln, Köln, Germany, 1998.
52. Bu, L.; Babu, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2008**, *38*, 156–172.