machine learning & knowledge extraction

MDPI

Article

Application of Dueling Double Deep Q-Network for Dynamic Traffic Signal Optimization: A Case Study in Danang City, Vietnam

Tho Cao Phan [1], Viet Dinh Le [2] and Teron Nguyen [3,*]

[1] Department of Civil Engineering, The University of Danang—University of Technology and Education, Danang 550000, Vietnam; pctho@ute.udn.vn
[2] School of Architecture, Civil and Environmental Engineering, Kumoh National Institute of Technology, Gumi 39177, Gyeongbuk, Republic of Korea; vietld@kumoh.ac.kr
[3] Samwoh Innovation Centre, Samwoh Smart Hub, 12 Kranji Way, Singapore 739454, Singapore
* Correspondence: teron.nguyen@samwoh.com.sg

Abstract

This study investigates the application of the Dueling Double Deep Q-Network (3DQN) algorithm to optimize traffic signal control at a major urban intersection in Danang City, Vietnam. The objective is to enhance signal timing efficiency in response to mixed traffic flow and real-world traffic dynamics. A simulation environment was developed using the Simulation of Urban Mobility (SUMO) software version 1.11, incorporating both a fixed-time signal controller and two 3DQN models trained with 1 million (1M-Step) and 5 million (5M-Step) iterations. The models were evaluated using randomized traffic demand scenarios ranging from 50% to 150% of baseline traffic volumes. The results demonstrate that the 3DQN models outperform the fixed-time controller, significantly reducing vehicle delays, with the 5M-Step model achieving average waiting times of under five minutes. To further assess the model's responsiveness to real-time conditions, traffic flow data were collected using YOLOv8 for object detection and SORT for vehicle tracking from live camera feeds, and integrated into the SUMO-3DQN simulation. The findings highlight the robustness and adaptability of the 3DQN approach, particularly under peak traffic conditions, underscoring its potential for deployment in intelligent urban traffic management systems.

Keywords: deep Q-network; fixed-phase; traffic signal control; SUMO simulation; mixed traffic flow

check for updates

Received: 17 June 2025
Revised: 10 July 2025
Accepted: 10 July 2025
Published: 14 July 2025

Citation: Phan, T.C.; Le, V.D.; Nguyen, T. Application of Dueling Double Deep Q-Network for Dynamic Traffic Signal Optimization: A Case Study in Danang City, Vietnam. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 65. https://doi.org/10.3390/make7030065

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Traffic congestion presents a persistent and significant challenge in urban environments, resulting in adverse economic and social impacts, such as prolonged travel times, increased fuel consumption, and elevated air pollution levels. A primary contributor to this issue is the inefficiency of traffic signal control at urban intersections. In Vietnam, the problem is exacerbated by rapid urbanization and the continuous growth in private vehicle ownership, placing increasing pressure on transportation infrastructure. This trend is observed not only in major metropolitan areas, like Hanoi and Ho Chi Minh City, but also in smaller, well-planned cities, such as Danang. According to the General Statistics Office's Population and Housing Census 2019 [1], Danang ranked second nationwide in car ownership, with 10.7% of households owning a vehicle, second only to Hanoi. As one of the most dynamic and systematically planned cities in Vietnam, Danang is now

experiencing worsening traffic congestion, underscoring the urgent need for effective traffic signal control strategies.

Conventional traffic signal control methods are widely implemented across global urban networks, including in Vietnam. These methods typically rely on fixed-time schedules [2–4] or traffic-responsive control schemes [5,6]. Signal timings are often pre-configured based on historical traffic patterns or vehicle counts during peak hours. While these approaches are cost-effective, simple to implement, and adequate under stable traffic conditions, they lack the flexibility to respond to dynamic and unpredictable traffic fluctuations caused by incidents or surges in demand. Moreover, as urban traffic systems grow more complex, traditional control strategies often fall short in terms of scalability and optimization.

To address these limitations, Adaptive Traffic Signal Control (ATSC) systems have emerged as a promising alternative, receiving growing attention for their ability to reduce congestion and improve vehicle flow through real-time optimization [7–9]. A variety of computational intelligence techniques have been employed to enhance ATSC, including genetic algorithms, swarm intelligence, and, notably, reinforcement learning (RL) [10–12]. Among these, RL offers significant advantages due to its ability to learn from environmental feedback and make real-time adjustments to traffic signals. Several studies have shown that RL algorithms can effectively manage traffic flow at intersections by continuously updating decision policies based on real-world data [13–16]. RL is particularly well-suited for large-scale, interconnected traffic networks, offering the ability to coordinate multiple intersections. Moreover, it does not require prior assumptions about the traffic system's structure, making it highly adaptable. However, RL methods are often hindered by the "curse of dimensionality", which can degrade performance in complex or large-scale scenarios [17].

To overcome these challenges, Deep Reinforcement Learning (DRL)—which combines the representational power of deep learning with RL's decision-making capabilities—has been introduced. DRL leverages neural networks to approximate value functions and policy mappings in environments with large state and action spaces, making it increasingly prevalent in traffic signal control research [18–25]. Recent studies have focused on enhancing the adaptability of DRL models to dynamic traffic conditions. For example, Zhang et al. [26] developed RL models that effectively learn and adapt to changing traffic environments, while Wei et al. [27] proposed a deep Q-learning-based approach tested under realistic conditions. These methods typically employ deep neural networks (DNNs) to relate traffic states (e.g., queue length, average delay) to reward functions, enabling intelligent, real-time decision-making. Building on this foundation, Ducrocp and Farhi [28] advanced DQN-based traffic control systems by introducing partial detection capabilities, showing improved performance across various scenarios. Subsequently, Cardoso [29] conducted a comparative evaluation of DQN variants, including Double DQN and Dueling Double DQN (3DQN), demonstrating that the 3DQN model achieved the most stable and robust performance under complex traffic conditions in Simulation of Urban Mobility (SUMO)-based simulations.

One of the emerging areas of focus in traffic signal control is the development of demand-responsive systems that utilize real-time traffic parameters. With the advancement of technologies such as the Internet of Things (IoT), it is now feasible to integrate real-time data from GPS devices, RFID systems, Bluetooth sensors, and smartphone applications (e.g., Google Maps, Apple Maps) into traffic control strategies [30,31]. In Vietnam, government regulations mandate the installation of journey-tracking devices in vehicles, which collect data, such as GPS coordinates and speed [32]. As a result, the growing penetration of wireless communication technologies in vehicles presents new opportunities for imple-

menting demand-responsive control systems, marking a critical area for future research and development.

Building upon the work of Ducrocp and Farhi [28], this study applies the 3DQN algorithm to optimize mixed traffic flow (MTF) control at a key intersection in Danang City, a context that has received limited attention in prior research. Unlike studies focused on traffic networks in highly regulated environments, this research examines an emerging urban area with diverse vehicle types, variable traffic patterns, and rapidly evolving infrastructure. Danang serves as a representative case study to evaluate the practical effectiveness of 3DQN in managing complex urban traffic.

To enhance real-world applicability, this study incorporates advanced sensing and tracking technologies, including YOLOv8 for real-time object detection and the SORT algorithm for vehicle tracking. These tools are used to extract dynamic traffic flow data from intersection surveillance cameras, which are then fed into the SUMO simulation environment coupled with the 3DQN model. This integrated approach allows for a more accurate representation of real-world MTF conditions, addressing limitations in prior studies that often rely on static or synthetic datasets. By combining YOLOv8, SORT, and the 3DQN-SUMO framework, the study aims to assess not only the technical feasibility of reinforcement learning in MTF control but also its practical relevance in dynamic, data-driven urban contexts.

The remainder of this paper is organized as follows: Section 2 outlines the theoretical foundation of deep Q-learning and the architecture of the 3DQN model. Section 3 presents a case study focused on the application of 3DQN in a cluster of intersections in southern Danang City. Sections 4 and 5 detail the experimental results and discussion. Finally, Section 6 concludes with key findings and implications for future research and practice.

2. Fundamentals of Deep-Q Learning

The RL-based traffic signal control model operates on three main components: the agent, the environment, and the elements of state, action, and reward (see Figure 1). In this model, the agent represents the traffic signal control system, which determines the duration of green, red, and yellow lights based on the current traffic conditions at the intersection. The environment encompasses the surrounding traffic conditions that influence the agent, including the number of vehicles, traffic flow speed, and congestion levels. The state describes explicitly the traffic situation at a given moment, for example, the number of vehicles waiting in different lanes and the remaining time for each traffic light cycle. The action refers to the decision made by the agent, such as adjusting the traffic light duration for each lane, with options like extending the green light for lanes with heavier traffic. The reward reflects the performance of the action taken by the agent and is calculated based on factors, such as vehicle wait time and the level of traffic congestion.

Q-Deep Learning system is a combination of Q-learning and DNNs designed to map the states in the system into the Q-value, representing values related to behavior from changes in external environmental conditions. For traffic light controller problems, Q-learning can be expressed by the following formula:

$$Q_{t+1}(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1}) + \gamma \cdot \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t) \quad (1)$$

where $Q(s_t, a_t)$ is the Q-value, which is updated in a decreasing manner depending on the learning rate α and the action value at state s_t. Meanwhile, r_{t+1} is the reward value estimated after performing the action state step s_t. Furthermore, the $Q(s_{t+1}, a_t)$ is represented the Q-value in the next state, and s_{t+1} is the next state after finishing the current state s_t. In addition, $\max_A$ indicates that the action has the highest value among the potential actions

at the selected state s_{t+1}. Another parameter, γ, is the discount factor indicating that the future reward is less important than the present, and its distribution value ranges from 0 to 1. Following the application of Q-learning in traffic light controller research, Equation (1) can be simplified as follows:

$$Q(s_t, a_t) = \alpha(r_{t+1}) + \gamma \cdot \max_a Q'(s_{t+1}, a_{t+1}) \tag{2}$$

where $Q'(s_{t+1}, a_{t+1})$ is the Q-value corresponding to action a_{t+1} at state s_{t+1}. There is a rule that the updated Q-value of the current action is performed in state s_{t+1} with immediate reward as well as the decreasing Q-values of the actions in the future. Therefore, $Q'(s_{t+1}, a_{t+1})$ is represented by the Q-value of the action in the future, holding the largest reward value after performing state s_{t+1}. Similarly, $Q''(s_{t+2}, a_{t+2})$ and $Q'''(s_{t+3}, a_{t+3})$ hold the maximum reward value in the next states s_{t+1}. This is how the agent will choose an action based not only on the current reward but also on the expected future discounted rewards. This rule can be simplified as follows:

$$Q(s_t, a_t) = r_{t+1} + \gamma \cdot r_{t+2} + \gamma^2 \cdot r_{t+3} + \gamma^3 \cdot r_{t+4} + \ldots + \gamma^{y-1} \cdot r_{t+y} \tag{3}$$

where y is a random value indicating the last time step before the end of the episode. Also, since there are no more feasible actions in state $t + y$, the value of r_{t+y} is 0. To solve Equation (3), specialized computational software, such as MATLAB lastest version or ealier, can be used; however, it is necessary to limit the number of elements of the chain to easily control the complexity of the calculation. For practical problems, DNNs is used to solve Equation (3) more effectively through the machine learning training process.

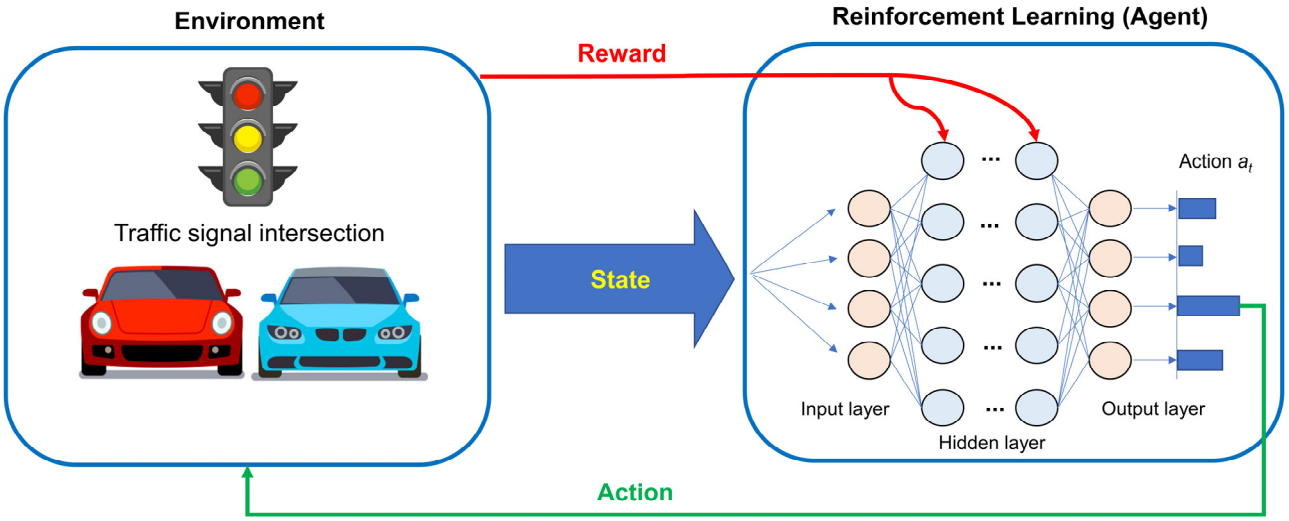


Figure 1. Schematic of reinforcement learning in traffic signal control.

The 3DQN algorithm is an advanced reinforcement learning method that builds upon the traditional DQN by addressing key limitations, such as overestimation bias and improved value decomposition. It begins by initializing two neural networks: the dueling Q-network $Q(s, a, \mathbf{W}_t)$, which estimates the action-value function with the random weights $\mathbf{W}_{t=0}$ and the target Q-network $Q_b(s, a, \mathbf{W}_t^-)$ initialized with weights $\mathbf{W}_{t=0}^- = \mathbf{W}_{t=0}$, which helps stabilize the learning process. The networks are designed with separate streams for

estimating the state-value function $V(s)$ and advantage function $A(s,a)$. As a result, the Q-value can be computed as:

$$Q(s,a,\mathbf{W}_t) = V(s,a) + \left[A(s,a) - \frac{1}{|A|}\sum_{a' \in A} A\big(s.a'\big)\right] \tag{4}$$

Following Equation (4), this decomposition allows the agent to efficiently learn both the value of being in a state and the advantage of each action.

In addition, a relay memory buffer D, which stores N last transitions, is also initialized and populated with $N_{\min}$ random transitions $A(s,a,s',a')$, which store the agent's past experiences. At the beginning of each episode, environment state s_t is observed. While the episode is ongoing and the terminal state is not reached, the agent must select action a_t. The action selection follows two parameters with probability ε and random action $a_t \in A$. With probability $1-\varepsilon$, the action that maximizes the Q-value is expressed as:

$$a_t = \underset{a}{\text{argmax}} Q(s_t, a, \mathbf{W}_t) \tag{5}$$

This policy ensures a balance between exploration and exploitation as the agent learns optimal actions. Upon selecting an action, the agent executes it in the environment and observes both the immediate reward r_{t+1} and the next state s_{t+1}. The transition $(s_t, a_t, s_{t+1}, r_{t+1})$ is stored in relay memory D. In order to update the network, a back-propagation is adapted with a batch $U(D)$ of M transitions (s_m, a_m, s'_m, r'_m) randomly sampled from D. For each sampled transition, temporal difference (TD) error δ_m is computed to update the Q-values of both online and target networks based on the forward propagation process. Therefore, it can be estimated as:

$$\delta_m = r_m + \gamma Q_b\big(s'_m, \text{argmax} Q\big(s'_m, a, \mathbf{W_t}\big), \mathbf{W_t^-}\big) \tag{6}$$

where γ is the discount factor, and Q_b is the target network's Q-value for the next state-action pair.

Following TD error δ_m, the Huber loss function is minimized to update the online Q-network, and it can be computed as:

$$\mathbf{L} = \frac{1}{2M}\sum_{m=1}^{M} \begin{cases} (\delta_m)^2 & \text{if } |\delta_m| < 1 \\ 2 \cdot |\delta_m| - 1 & \text{otherwise} \end{cases} \tag{7}$$

The Huber loss is minimized using the Adam optimizer, which adjusts weight $\mathbf{W_t}$ based on gradient $\nabla_{\mathbf{W}}\mathrm{L}$. Therefore, the weight matrix of online network Q can be updated as:

$$\mathbf{W_{t+1}} = \mathbf{W_t} - \eta \nabla_{\mathbf{W}} \mathbf{L} = \mathbf{W_t} - \eta\left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \ldots, \frac{\partial L}{\partial w_n}\right) \tag{8}$$

where η is the learning rate, which controls the epoch in training process.

In addition, the target Q-network is updated using a Polyak averaging update

$$\mathbf{W_t^-} = \chi \mathbf{W_t} - (1-\chi)\mathbf{W_t^-} \tag{9}$$

where χ is a small value which allows the target network to slowly monitor the online Q-network.

Finally, the exploration rate α is decayed exponentially after each time step to progressively shift the focus from exploration to exploitation as the agent learns the optimal policy:

$$\alpha_t = \alpha_0 \cdot \exp(-k \cdot t) \tag{10}$$

where k is a decay factor. This ensures that the agent explores sufficiently in the early stages of training while increasingly exploiting learned knowledge as training progresses. This iterative process of action selection, experience replay, Q-network updates, and exploration–exploitation balance is repeated for each episode, resulting in an agent that learns an optimal policy over time.

3. Case Study: Application of 3DQN

3.1. Study Location: Traffic Intersection

Danang City, as one of the country's five direct-controlled municipalities, falls under the administration of the central government. It is located in central Vietnam, with a strategically important position along Vietnam's North–South transportation corridor (see Figure 2). The city's central area is close to the East and Northeast coasts, while the western region is predominantly mountainous. Due to these geographical features, Danang's urban planning has predominantly developed towards the south, making the intersections in the southern part of the city crucial for regulating traffic flow. These intersections frequently experience congestion, especially during peak hours.

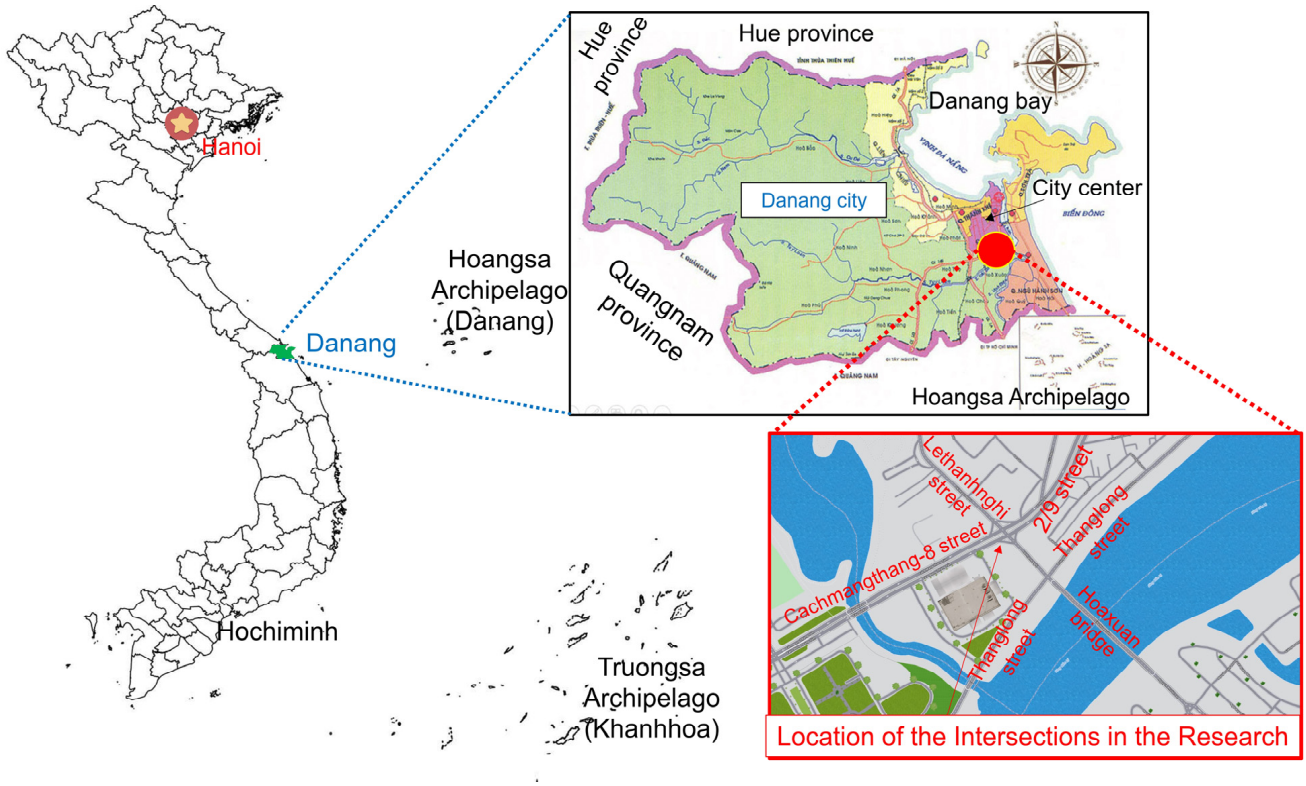


Figure 2. Location of the intersections.

In this study, we focus on the major traffic intersections at the junction between Nguyenthanhnghi Street and the main arteries Cachmangthang-8 and 2/9 Streets, located in the southern part of Danang City (see Figure 2). These intersections not only serve as key points for traffic leading into the city's gateway but also connect to Danang International Airport, as shown in Figure 3. Additionally, these intersections connect new urban areas, including the Hoaxuan urban area, further increasing their importance within the urban traffic system.

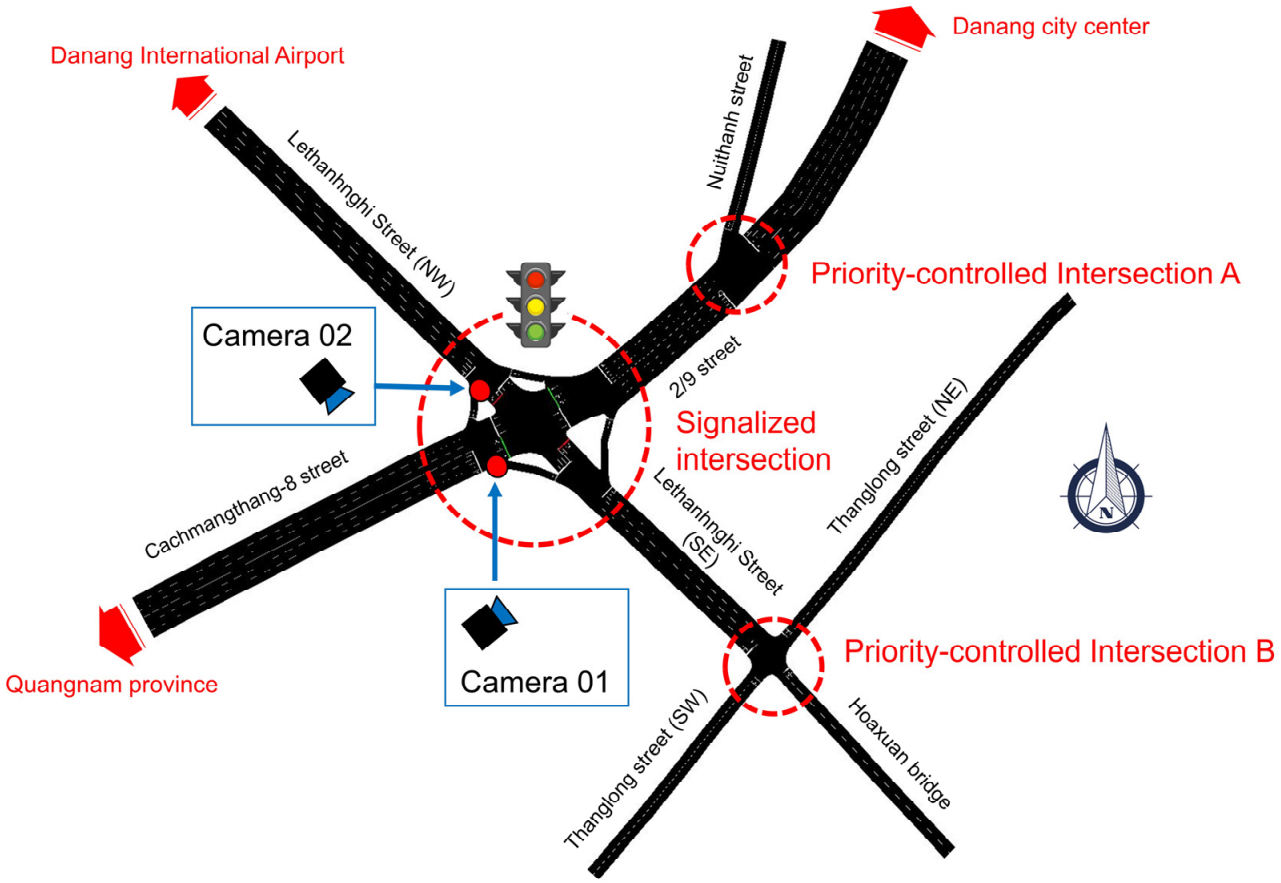


Figure 3. Layout of intersections.

Specifically, traffic signal control is implemented at the main intersection to manage vehicle flow, with dedicated lane designs to protect right-turning traffic. Cachmangthang-8 Street, which connects with 2/9 Street, features a tree-planted island median with three lanes on each side. Notably, there is also a branch road at the intersection from the northeast, Nuithanh Street, just 110 m away, which complicates traffic regulations during peak hours. Moreover, Nguyenthanhnghi Street and the extended section towards the Hoaxuan Bridge have only two lanes on each side, without a hard median. The end of the Hoaxuan Bridge intersects with Thanglong Street, creating another critical intersection within the city's traffic network. Figure 3 provides a detailed layout of these intersections.

At this intersection, the traffic light system has three phases under a fixed-cycle traffic signal control system, with a total signal cycle of 103 s (see Figure 4). In this cycle, the traffic flow from the northwest (NW) to the southeast (SE) receives a green signal for 32 s, the flow from the northeast (NE) to the southwest (SW) receives 28 s, while the flow from the southwest (SW) to the northwest (NW) and from the northeast (NE) to the southeast (SE) receives only 22 s of green light. This time allocation is crucial for regulating and managing traffic flow in the area, aiming to reduce congestion and improve the operational efficiency of the urban traffic system. The traffic volumes for the signalized and self-regulated intersections are shown in Tables 1 and 2, respectively. The traffic volumes were collected by the Danang Department of Transport in 2022 [33].

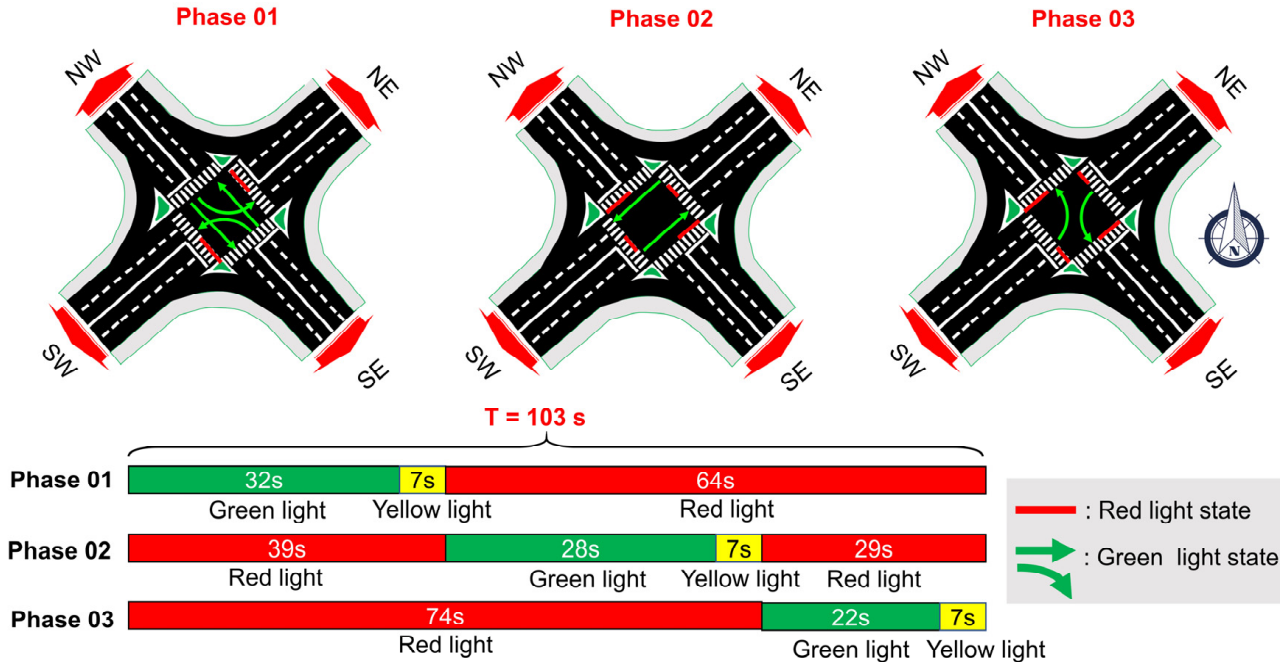


Figure 4. Fixed-phase traffic light control design for a signalized intersection.

Table 1. Traffic volumes for the signalized intersection [33].

ID	Lane Group	Direction	Denotation	Traffic Flow During Peak Hours (Passenger Car Equivalent/Hour)	Percentage (%)
1	Lethanhnghi (NW)	Go straight on Hoaxuan Bridge	NW-SE	696	16.58
		Turn right onto Cachmangthang8 Street	NW-SW	318	
		Turn left onto 2/9 Street	NW-NE	9	
2	Lethanhnghi (SE)	Go straight on Danang International Airport	SE-NW	369	20.86
		Turn right onto 2/9 Street	SE-NE	711	
		Turn left onto Cachmangthang8 Street	SE-SW	207	
3	2/9 Street (NE)	Go straight on Quangnam province	NE-SW	942	38.84
		Turn right onto Lethanhnghi (NW) to go to Danang International Airport	NE-NW	96	
		Turn left onto Lethanhnghi (SE) to go to Hoaxuan Bridge	NE-SE	1359	
4	Cachmangthang8 Street (SW)	Go straight on Danang City center	SW-NE	1059	23.72
		Turn right onto Lethanhnghi (SE) to go to Hoaxuan Bridge	SW-SE	300	
		Turn left onto Lethanhnghi (NW) to go to Danang International Airport	SW-NW	105	
		Total		6171	

Table 2. Traffic volumes for priority-controlled intersection B [33].

ID	Lane Group	Direction	Denotation	Traffic Flow During Peak Hours (Passenger Car Equivalent/Hour)	Percentage (%)
1	Lethanhnghi (SE)	Go straight on Hoaxuan Bridge	NW-SE	1545	47.84
		Turn right onto Thanglong Street (SW)	NW-SW	618	
		Turn right onto Thanglong Street (NE)	NW-NE	60	
2	Hoaxuan Bridge	Go straight on Danang International Airport	SE-NW	1107	30.86
		Turn right onto Thanglong Street (NE)	SE-NE	261	
		Turn left onto Thanglong Street (SW)	SE-SW	66	
3	Thanglong Street (NE)	Go straight on Quangnam province (NE-SW)	NE-SW	333	13.75
		Turn right onto Lethanhnghi Street to go to Danang International Airport	NE-NW	108	
		Turn left onto Hoaxuan Bridge	NE-SE	198	
4	Thanglong Street (SW)	Go straight on Danang city center	SW-NE	120	7.55
		Turn right onto Hoaxuan Bridge	SW-SE	75	
		Turn left onto Lethanhnghi Street to go to Danang International Airport	SW-NW	156	
		Total		4647	

3.2. Architecture of 3DQN Model

A Convolutional Neural Network (CNN) is a crucial component of the DQN model. It is a specialized neural network designed for image processing and pattern recognition. CNN operates in two main learning stages. The first stage is the convolutional module, consisting of multiple layers. These layers gradually detect more complex patterns in the images through 2D convolution, where filters (or kernels) slide across the input data in a windowed fashion, generating feature maps. The filters share weights across the entire datasets, optimizing computational efficiency. The convolution parameters include the number of input channels (C), filter size (K), and stride (S). Pooling layers are typically added between convolutional layers to reduce the spatial dimensions of the feature maps, thus reducing the computational load. The second stage is the fully connected module, where the output from the convolutional layer is flattened and passed through a Multi-Layer Perceptron (MLP) for classification or regression tasks.

In this study, the convolutional 3DQN model consists of five layers: the input layer, the CNN mapping layer, the fully connected layer, the stream layer, and the output layer (as shown in Figure 5). The input layer receives data in the form of matrix grids from the Dynamic Traffic State Estimation (DTSE), which represents the traffic flow states coming from different directions at the simulated intersections in SUMO software version 1.11. As presented in Ref. [28], DTSE encodes the traffic environment into a structured matrix

grid, where each cell contains information, such as vehicle presence, speed, and distance to the nearest vehicle ahead. These grids are continuously updated over time to reflect the dynamic changes in traffic conditions across different lanes and approaches. This matrix is defined similarly to an image with 3 channels, where each "pixel" corresponds to a grid in the DTSE, indicating the presence of vehicles at intersections (as shown in Figure 6). The convolutional 3DQN model processes the DTSE matrix, treating it as a multi-channel image to determine traffic signal phases via Q-value predictions. The CNN mapping layer consists of two convolutional layers; the first convolutional layer (CNN1) extracts basic features from the DTSE matrix, detecting general traffic patterns. The second convolutional layer (CNN2) further identifies more complex features, allowing the network to better understand the variations in traffic flow at intersections. The fully connected layers (FC1 and FC2) aggregate the information from the extracted features, preparing for the output calculation.

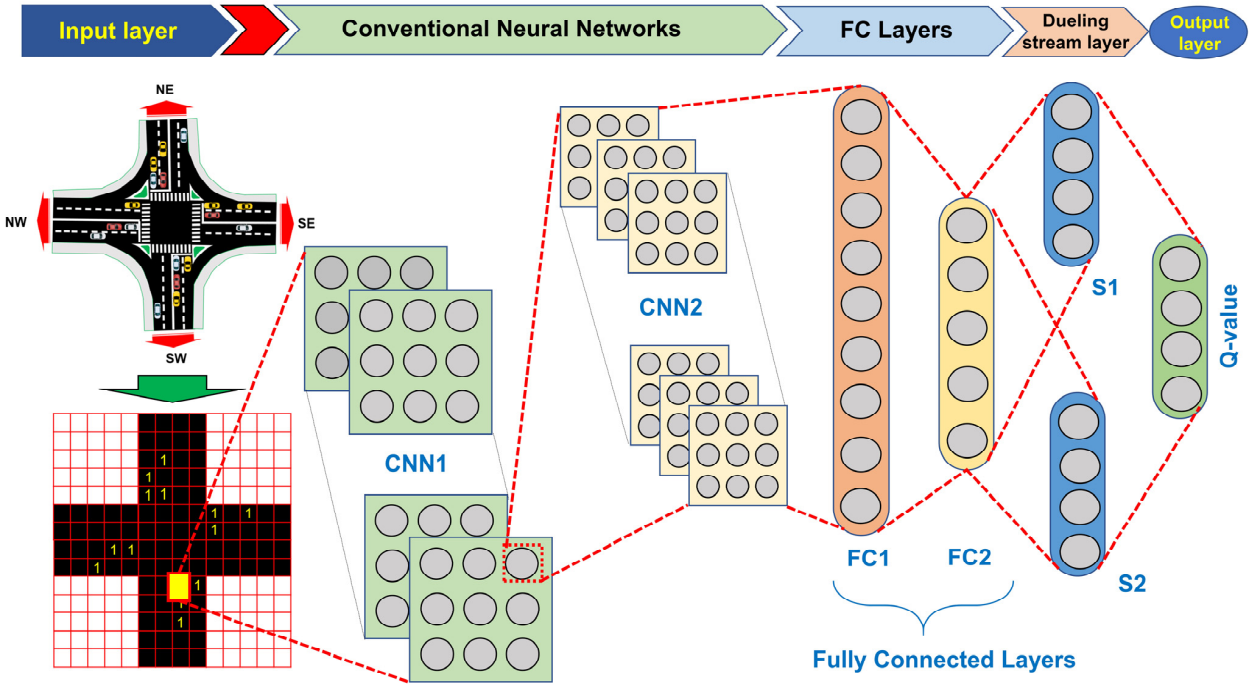


Figure 5. Architecture structure of 3DQN.

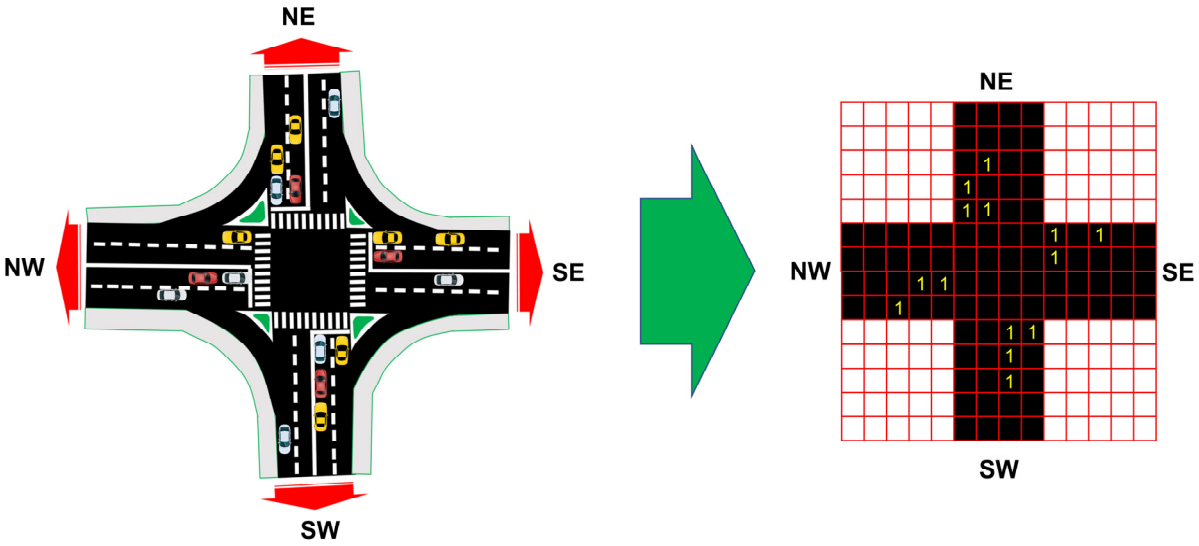


Figure 6. Schematic of dynamic traffic state estimation.

A key aspect of 3DQN is that after passing through the fully connected layers, the model splits into two streams: the state-value stream assesses the overall traffic state, while the advantage stream calculates the advantage of each traffic light phase over others. The final output layer combines both state-value and advantage to produce the Q-value, representing the optimal traffic signal control action. This structure enables the model to learn and make precise traffic control decisions and optimize traffic light performance based on real-time traffic conditions.

To ensure the efficiency of the 3DQN model, the architecture of the CNN layers and the number of neurons in the fully connected layers are crucial. According to the study by Ducrocp and Farhi [28], the optimal architecture for CNNs and fully connected layers has been identified. Specifically, the CNN1 uses 16 filters with a kernel size of 4 and a stride of 2. The CNN2 uses 32 filters with a smaller kernel size of 2 and a stride of 1. The fully connected layers consist of 128 neurons in the FC1 and 64 neurons in the FC2. Additionally, the model's hyperparameters need to be fine-tuned before training the convolutional 3DQN model. Based on the research by Ducrocp and Farhi [28], the model was fine-tuned using a grid search across hyperparameters, with tuning runs set to 200,000 timesteps. The fine-tuned hyperparameters include a learning rate of 10^{-4} using the Adam optimizer and the Huber loss function. The discount factor gamma is set to 0.99, with an initial epsilon value of 0.001, and epsilon decays exponentially with a decay factor of 2 million steps. The experience replay memory has a maximum size of N = 1 million steps and a minimum size of N = 0.1 million steps. The soft Polyak update rate is 10^{-3}, ensuring gradual updates between the target and the main networks. No terminal state is defined for the agent, and training episodes are terminated early by an internal time limit in the SUMO simulation. This prevents simulations from running excessively long and focuses on the critical parts of the learning process.

The agent actions are an important factor in the 3DQN model. This factor involves selecting the next phase from the set of possible phases to apply for a minimum green interval time $T_{g,\min}$. If the predicted phase by 3DQN is the same is the same as the current one, the green light duration T_g is extended by $T_{g,\min}$. If a different phase is selected, then the model in SUMO shifts to that phase, including transition period $T_y + T_r$ for changing and clearing intervals, followed by a new green interval beginning with $T_{g,\min}$. Consequently, the agent's decision to select the next phase also controls how long the current phase lasts if it selects the same phase repeatedly. The agent action space is based on the number of possible phases in the traffic light program. At a four-way intersection, there are typically eight valid signal phases that avoid conflicts in vehicle movements. To reduce conflicts among simultaneous movements, the set of phases typically includes either two or four options. Figure 7 illustrates these two kinds of action spaces. In a two-phase program, there are two permissive green intervals per axis that allow left turns, straight-through, and right-turn movements. For the two-phase program, the action space can be expressed as:

$$A = \left\{ (NW \rightarrow NE - SE, SE \rightarrow SW - NW) \quad (SW \rightarrow NW - NE, NE \rightarrow NW - SW) \right\}^T \tag{11}$$

In a four-phase program, the action space expands to four protected green intervals, with two per axis, allowing protected left turns or straight-through and right turns only. The action in this case can be estimated as:

$$A = \begin{Bmatrix} NW \rightarrow SE, SE \rightarrow NW \\ NW \rightarrow NE, SE \rightarrow SW \\ NE \rightarrow SW, SW \rightarrow NE \\ SW \rightarrow NW, NE \rightarrow SE \end{Bmatrix} \tag{12}$$

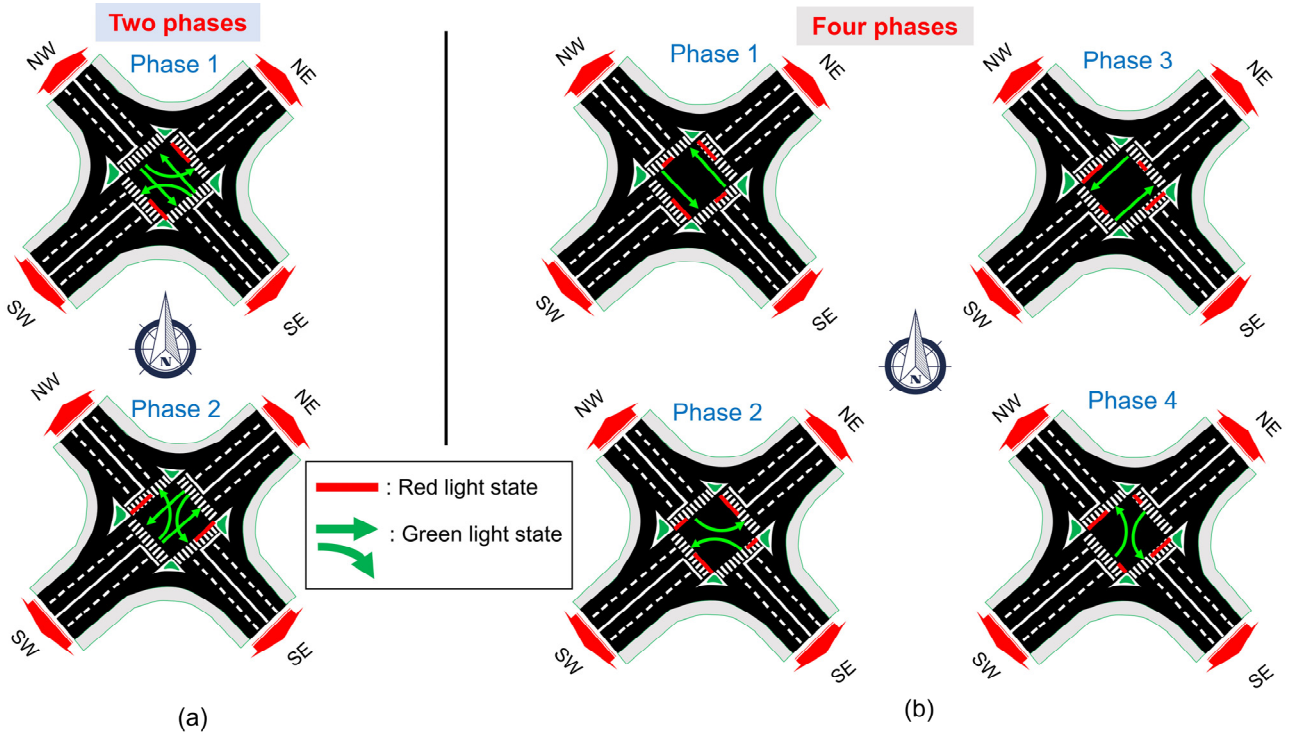


Figure 7. Description of agent's space for Dueling Double DQN: (**a**) 2-phases; (**b**) 4-phases program.

4. Evaluation of 3DQN

This study applied the 3DQN model to optimize traffic signal control at intersections and compare its performance with a fixed three-phase signal control method. Figure 8 illustrates this optimization process through a flowchart of the 3DQN agent. Before training, a simulation model was constructed using simulation in SUMO software version 1.11 [34], with the geometric design parameters of the intersection configured according to design standards. Traffic flow within the model was defined based on actual peak-hour traffic flow data.

The Traffic Control Interface (TraCI) API is an application programming interface that enables external scripts to interact with the SUMO simulation model. Through the TraCI API, SUMO is integrated with Python 3.6, enabling real-time control within the simulated environment [12,28,29,35]. This intersection simulation model is subsequently used as the environment for training the 3DQN agent. SUMO provides the agent with environmental states and rewards, guiding the agent in learning an optimal control policy. The agent returns actions in phase timing for the traffic signals, which are then implemented back into the simulation. This interactive loop facilitates continuous learning, enabling the agent to adapt its strategy based on real-time feedback from the environment.

A dynamic traffic flow generation module, also connected to SUMO, further enhances the simulation flexibility by varying traffic flow randomly from 50% to 150% of the reference flow. This method provides variability in traffic conditions, ensuring that the 3DQN model can handle unstable traffic scenarios in real-world applications—an essential factor for future applications. In addition, demand-responsive traffic signal control is analyzed. For simplicity, a connected vehicle (CV) penetration rate factor is assumed to vary randomly between 0 and 1. This factor is calculated using the binary matrix of CV position P, where 1 indicates the presence of a CV and 0 indicates its absence. The matrix of CV speed V, normalized to the speed limits of the approaches, is also used. The penetration rate is combined with vehicle speeds to introduce variability into the SUMO simulation. Applying this factor will assist in evaluating the performance of traffic signal control using the 3DQN

model. For a more detailed explanation of this factor and reward function of DQN, refer to the study by Ducrocp and Farhi [28].

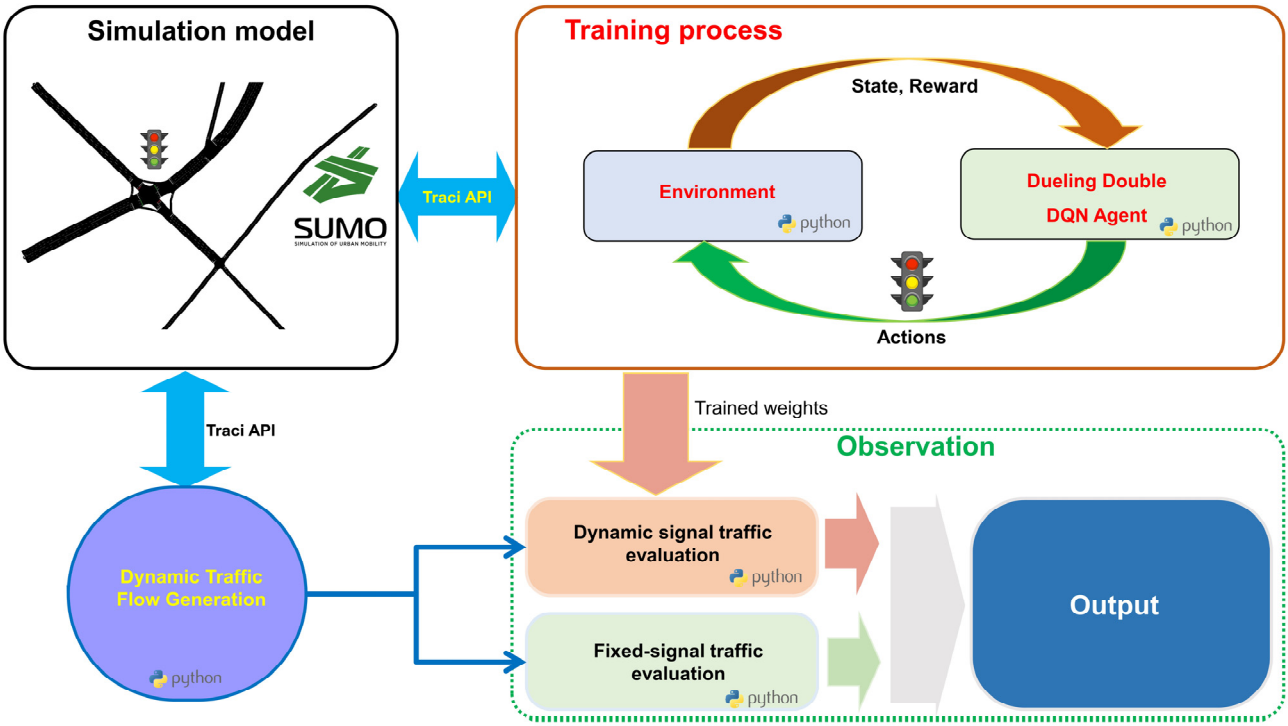


Figure 8. Flowchart of training and evaluation processes for 3DQN.

Figure 9 shows the variation in the reward value of the 3DQN model during training, reflecting the model performance improvement across training steps. Initially, the reward value exhibits large fluctuations, possibly because the model has not yet fully learned the system characteristics in an environment with fluctuating traffic flow. However, the reward value begins to gradually increase and stabilize from the 1.5 millionth training step, maintaining stability from the 3 millionth to the 5 millionth training step. The small fluctuations in the curve are a common phenomenon in reinforcement learning, as the model experiments with different strategies before stabilizing and finding the optimal strategy. To evaluate model performance, the weight matrices of the model at the 1 millionth (1M-Step) and 5 millionth (5M-Step) training steps were saved and compared during the model evaluation step to assess the impact of the number of training steps on the learning process and the effectiveness of the 3DQN model.

Upon completion of training, the 3DQN model is evaluated by comparing its performance with that of the fixed-signal control method. Two separate evaluation modules are employed, with the 3DQN evaluation module utilizing the trained weight matrix of the agent. In contrast, the fixed-signal control evaluation module simply uses TraCI API to interact with the SUMO simulation model, employing pre-established fixed three-phase signals (Figure 4). The dynamic traffic flow generation module generates the input traffic flow for both modules, ensuring that both methods are under the same traffic conditions. The results are then compiled in the output block, enabling a detailed comparison of the performance between dynamic and fixed-signal control. These findings offer valuable insights for recommending the practical application of 3DQN to improve traffic efficiency under various conditions.

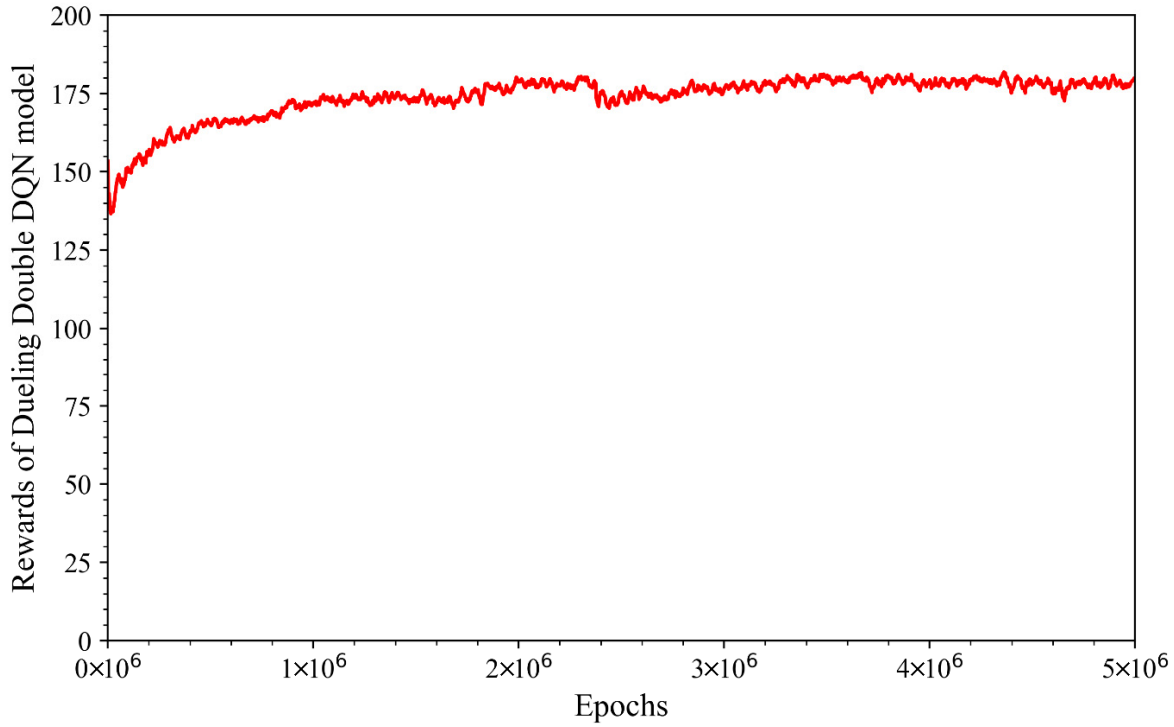


Figure 9. Evolution of reward versus training steps of 3DQN.

Figure 10 presents the frequency distribution of vehicle delays under two traffic signal control methods—3DQN and fixed-signal control—based on a 1000-h simulation using the SUMO platform. The 3DQN method was tested using two pre-trained models: one trained for 1M-Step and the other for 5M-Step. The x-axis represents vehicle delay intervals (e.g., 0–5 min, 5–10 min), while the y-axis indicates the frequency of each delay interval.

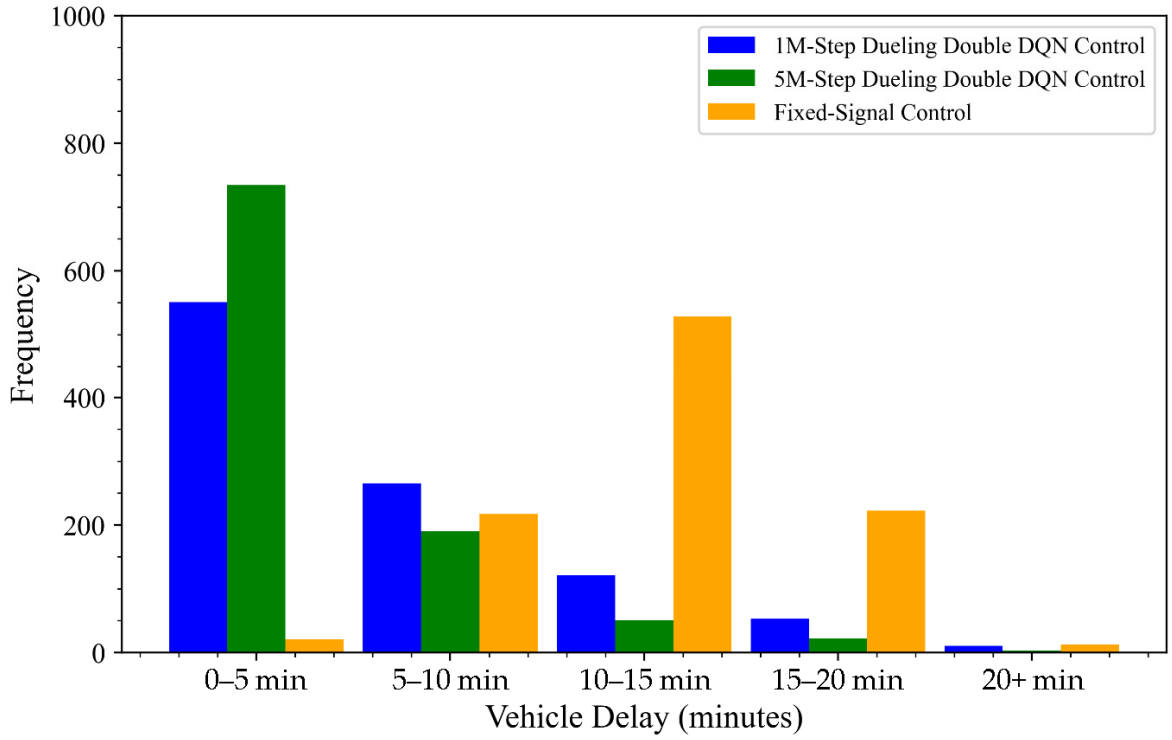


Figure 10. Comparison of vehicle delay frequencies: 3DQN vs. fixed-signal control.

The results clearly differentiate the performance of the two approaches. Both 3DQN models (1M-Step and 5M-Step) significantly reduce the frequency of long waiting times (over 5 min) compared to the fixed-signal control method. This highlights 3DQN's ability to dynamically adapt to changing traffic conditions, thus improving traffic flow and mitigating congestion. In contrast, while the fixed-signal control method exhibits a higher frequency of short delays (0–5 min), it also shows a substantial rise in delays exceeding 10 min, revealing its limited capacity to respond to variable traffic patterns.

When comparing the two 3DQN models, the 5M-Step model outperforms the 1M-Step model by achieving 750 instances of vehicle delays under 5 min, compared to 550 instances for the 1M-Step model. Furthermore, the 5M-Step model records a significantly lower frequency of delays over 5 min. These findings suggest that the number of training steps has a substantial impact on the model's performance and stability in real-world conditions. Hence, selecting an appropriate training duration is critical to ensuring that the 3DQN model reaches its full potential and effectively adapts to dynamic traffic environments.

To assess the impact of the CV penetration rate coefficient, vehicle delays were surveyed and analyzed when applying both traffic control methods: 3DQN and fixed-signal control. Figure 11 presents a chart showing the distribution of vehicle delay and the CV penetration rate for the two control methods: fixed-signal control and 3DQN control. When applying the fixed-signal control method, vehicle delays are randomly distributed with a wide amplitude, and the average waiting time remains almost unchanged as the CV penetration rate increases. This indicates that the traditional method lacks the ability to connect traffic flow information, leading to inflexibility in adapting to traffic fluctuations. Thus, the vehicle delay remains unchanged as the CV penetration rate increases.

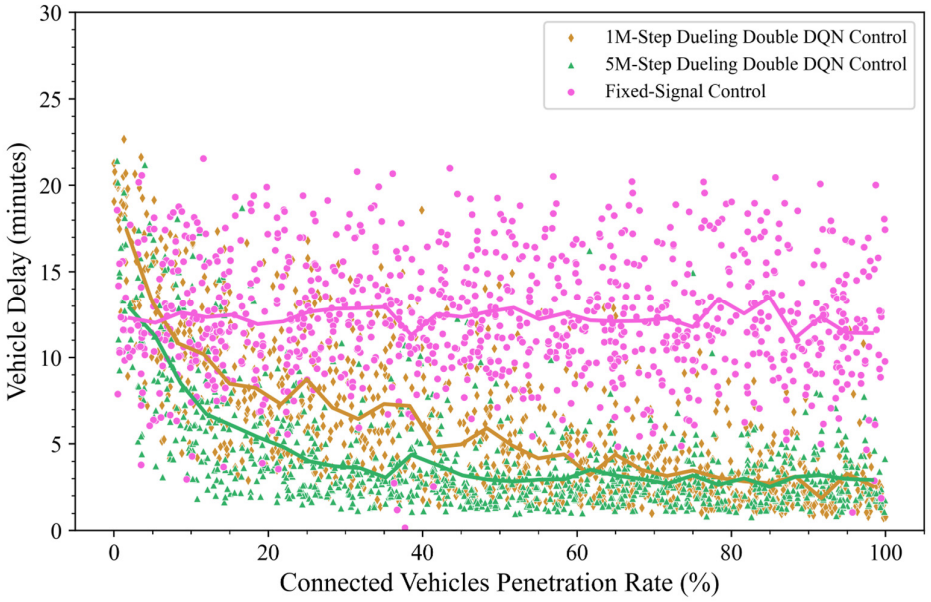


Figure 11. Comparison of vehicle delay versus CV penetration rate: 3DQN vs. Fixed-signal control.

In contrast, the 3DQN control method shows a tighter distribution of vehicle delay, which tends to decrease as the CV penetration rate increases. This demonstrates that the 3DQN model can optimize waiting times more effectively when sufficient traffic flow information, such as vehicle locations and speeds, is available, even if the vehicle-connected penetration rate may not be fully complete. Comparing the 1M-Step and 5M-Step Dueling Double DQN models, the results show that the 5M-Step model is more stable and can reduce the average waiting time to under 5 min when the CV penetration rate reaches 20%, while the 1M-Step DQN model requires a penetration rate of over 40%. It is worth

noting that both the 1M-Step and 5M-Step models begin to stabilize and optimize vehicle waiting times to 3–4 min when the CV penetration rate reaches 70% or higher. Therefore, the 5M-Step model proves to be more efficient, requiring only about 20% of the information from traffic flow. In comparison, the 1M-Step model needs 40% of the vehicles in the flow to be connected to the traffic control system. This difference indicates that the number of training steps in the model plays a key role in the system's flexibility to adapt when full vehicle-connected information is unavailable.

5. 3DQN Performance in Mixed Traffic Patterns

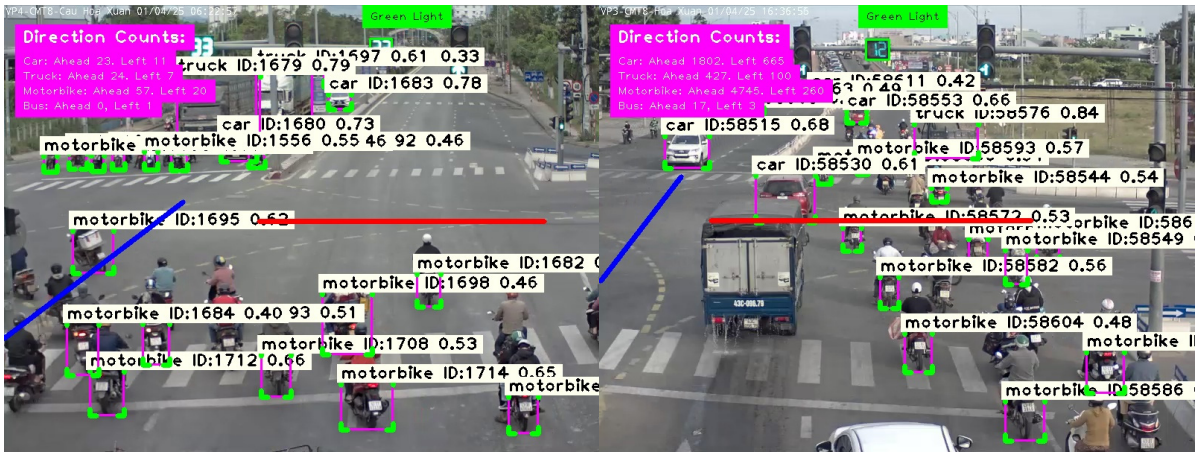
The 5M-Step 3DQN model has demonstrated superior performance in optimizing vehicle delays at intersections. The traffic flow data presented in Section 4 is based on random traffic simulated over a period of 1000 h, using static traffic values corresponding to peak hours. However, this data does not fully represent the complex fluctuations of real traffic flow, which vary over time and depend on many factors, such as peak hours, weather conditions, or unusual traffic events. To accurately evaluate the impact of MTF on the performance of the 5M-Step 3DQN model, this section presents the process of collecting and processing MTF data, as well as the method for evaluating the model's performance under real conditions.

MTF data at the signalized intersection is collected through the installed camera system, using the YOLOv8 model [36,37] to detect and classify vehicles, combined with the SORT (Simple Online and Real-time Tracking) algorithm [38] to track and determine the direction of movement of each vehicle type. The classified vehicle types include motorbikes, cars, trucks, and buses. To ensure consistency with the training data of the 5M-Step 3DQN model, the dynamic traffic volume of each vehicle type is converted to the equivalent passenger car unit, a common standardized unit in traffic research. The pre-trained 5M-Step 3DQN model will be integrated with the SUMO to evaluate the performance under dynamic traffic flow conditions, more accurately reflecting real-life traffic scenarios that change over time.

MTF data were collected at the signalized intersection on April 1, 2025 (from 6:00 to 18:00) to evaluate the effectiveness of the 5M-Step 3DQN model. Two cameras were installed at the intersection: Camera 01, recording the direction from Cachmangthang-8 Street to the center of Danang City, and Camera 02, recording the direction from the international airport approach road (Lethanhnghi) to Hoaxuan Bridge. Figure 3 illustrates in detail the installation locations of the two cameras at the signalized intersection, ensuring coverage of the main traffic flows. However, due to the limitation of the number of cameras, the dynamic traffic flow in directions not directly recorded is estimated based on the traffic ratio of the two main directions recorded by the cameras. This ratio is determined through the analysis of peak hour traffic volume, ensuring the representatives of actual traffic conditions.

To perform the analysis, a custom-built program was developed based on the pre-trained YOLOv8 model and the SORT library in Python. This program allows for accurate determination of the traffic volume and direction of movement of various types of vehicles, including motorcycles, cars, trucks, and buses. At each camera, two lines are set up to record the traffic volume in two directions: straight and left turn, as illustrated in detail in Figure 12. In this figure, the red line is used to count the number of vehicles going straight, while the blue line is used to count the number of vehicles turning left. Since the cameras are oriented towards the intersection, an additional algorithm was developed to determine the color of the traffic light to ensure that vehicle counting is only performed when the traffic light is green, allowing vehicles to move in the straight or left turn direction. In addition, each vehicle recognized by the YOLOv8 model is assigned a confidence score,

which evaluates the accuracy of the prediction process. This score plays an important role in evaluating the performance of the recognition model. The SORT algorithm is then used to track and determine the direction of each vehicle based on its trajectory. When a vehicle passes the direction line (whether going straight or turning left), the corresponding vehicle type count is recorded and accumulated in real-time. This result provides detailed data on dynamic traffic flow by vehicle type and direction, creating a basis for evaluating the performance of the 5M-Step 3DQN model in optimizing waiting time and traffic management at the intersection.



(**a**) (**b**)

Figure 12. Classification and counting resutls of vehicle types at the intersection: (**a**) Camera 01 captured from Cachmangthang-8 Street to Danang City center (SW-NE); (**b**) Camera 02 captured from Lethanhnghi Street to Hoaxuan Bridge (NW-SE).

Figures 13 and 14 show the cumulative traffic volume of vehicle types and the confidence scores of the YOLOv8 model from the image data of Camera 01 in two directions: straight and left turn, respectively. The overall analysis shows that the traffic volume of straight vehicles, as recorded from Camera 01, accounts for the majority compared to the left turn direction. In the straight direction, from Cachmangthang-8 Street to the center of Danang City, motorbikes are the type of vehicle with the highest traffic volume, followed by cars and trucks. Notably, in the left turn direction, from Cachmangthang-8 Street to Lethanhnghi Street towards Danang International Airport, the recorded traffic volumes of motorbikes and cars are similar. Regarding the prediction performance of the YOLOv8 model, the average confidence score in the first stage shows fluctuations, possibly due to the limited number of recognized vehicles. However, when the traffic volume increases, the model shows stability with the confidence score remaining high and fluctuating little over time. Specifically, in the straight direction (Figure 13), large vehicles, such as cars, trucks, and buses, have high confidence scores, ranging from 0.75 to 0.9, while motorcycles have lower confidence scores, ranging from 0.55 to 0.6. In the left turn direction (Figure 14), the confidence score of motorcycles is similar to that of trucks and buses but still lower than that of cars. The reason may come from the diversity in color, size, and purpose of use of motorcycles, as well as factors, such as the driver's clothing or the goods carried, making it difficult for the YOLOv8 model to accurately recognize them.

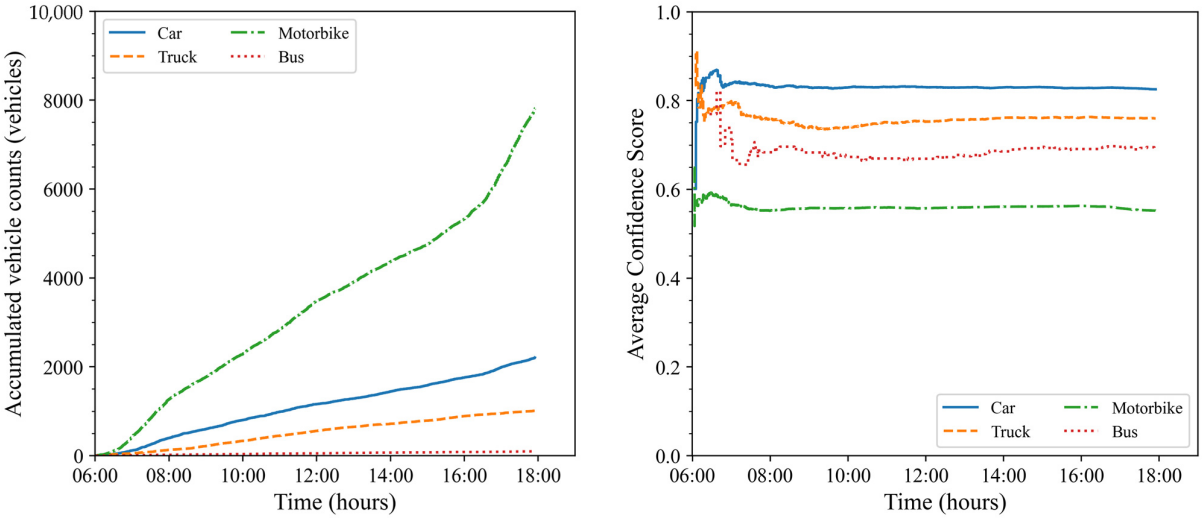


Figure 13. Vehicle traffic flow and YOLOv8 model confidence scores from Camera 01, straight direction entering the intersection: (**a**) Cumulative vehicle flow; (**b**) Confidence scores for vehicle detection.

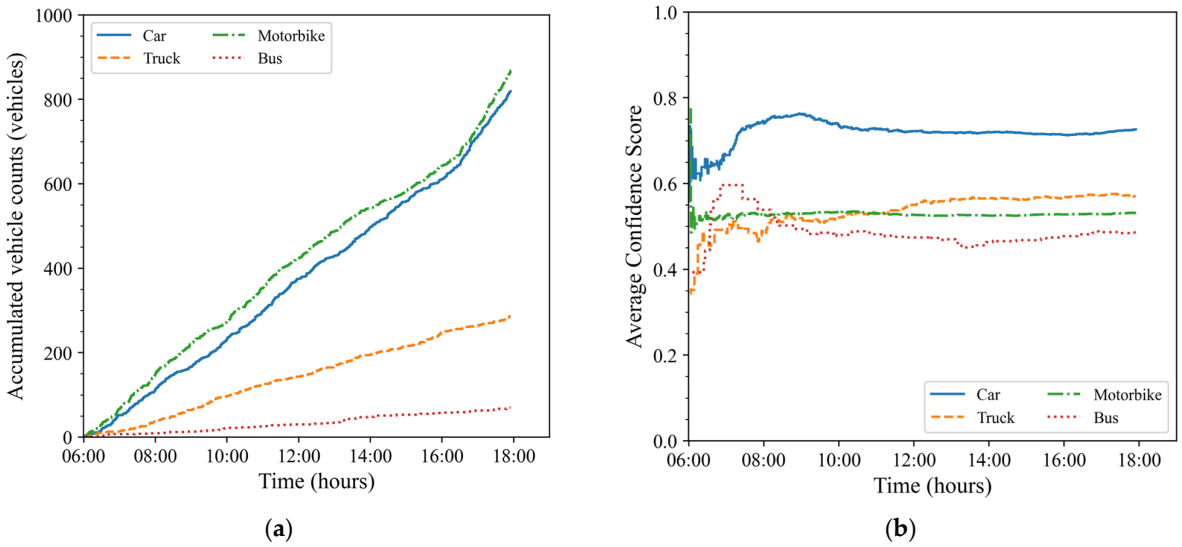


Figure 14. Vehicle traffic flow and YOLOv8 model confidence scores from Camera 01 in the left-turn direction: (**a**) Cumulative vehicle flow; (**b**) Confidence scores for vehicle detection.

Figures 15 and 16 show the cumulative traffic volume of vehicle types and the predicted confidence score of the YOLOv8 model from the image data of Camera 02 in two directions: straight and left turn. The overall analysis shows that the traffic volume of vehicles going straight, as recorded by Camera 02, accounts for the majority compared to the left turn direction. Specifically, in the straight direction from Lethanhnghi Street to Hoaxuan urban area (via Hoaxuan Bridge), motorbikes are the type of vehicle with the highest traffic volume, followed by cars and trucks. Notably, in the left turn direction from Lethanhnghi Street to 2/9 Street towards the center of Danang City, cars account for the majority of traffic volume, far exceeding other types of vehicles. Regarding the prediction performance of the YOLOv8 model, the average confidence score shows a similar fluctuation to the data from Camera 01, especially in the early stages when traffic volume is still low. According to the data from Figures 13–16, the average confidence score of all vehicle types at both Camera 01 and Camera 02 locations ranges from 0.5 to 0.9. This confidence

score is considered acceptable for use in the performance analysis of the 5M-Step 3DQN model when integrated with the SUMO traffic simulation software.

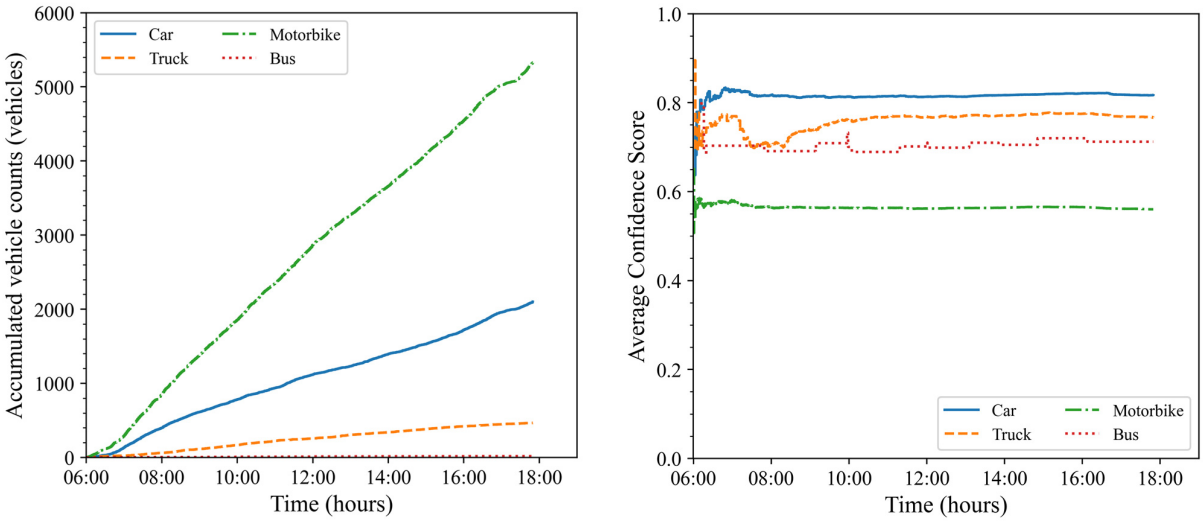


Figure 15. Vehicle traffic flow and YOLOv8 model confidence scores from Camera 02, straight direction entering the intersection: (**a**) Cumulative vehicle flow; (**b**) Confidence scores for vehicle detection.

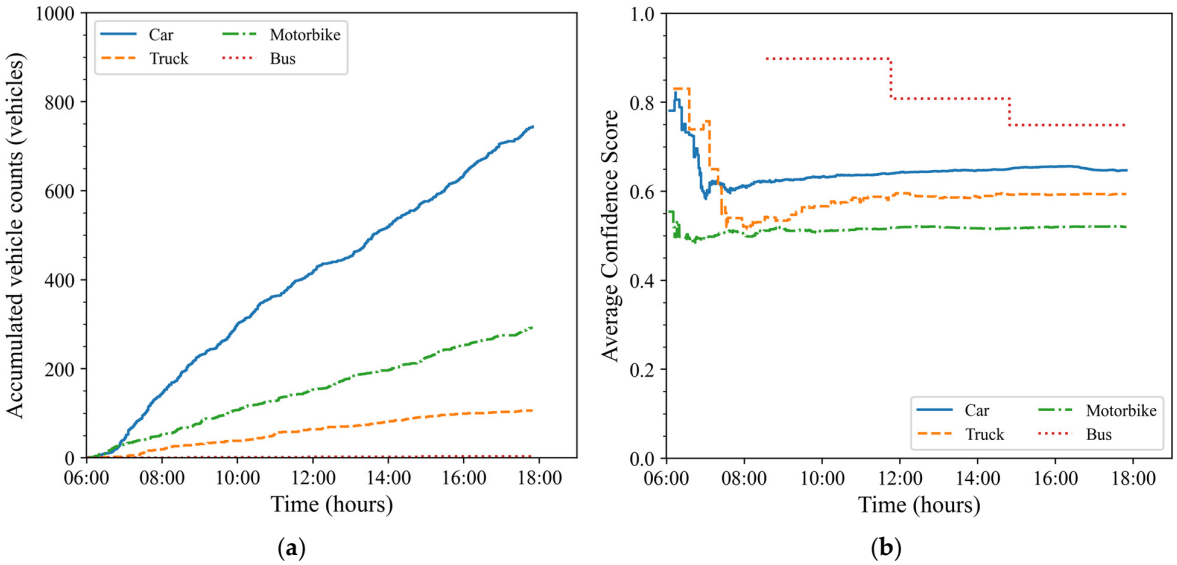


Figure 16. Vehicle traffic flow and YOLOv8 model confidence scores from Camera 02 in the left-turn direction: (**a**) Cumulative vehicle flow; (**b**) Confidence scores for vehicle detection.

Following the MTF database, which is collected cumulatively at two surveillance camera locations, the hourly average traffic volume is calculated to optimize data input into the SUMO model. These data are combined with the pre-trained datasets of 5M-Step 3DQN to improve the accuracy of traffic flow analysis and prediction. To ensure consistency and comparability between vehicle types, the hourly average traffic volume of each vehicle type is converted to the equivalent passenger car unit. The conversion factor applied to each vehicle type is as follows: motorbike is 0.3, car is 1.0, bus is 2, and truck is 2.5. These factors reflect the impact of each vehicle type on traffic flow, based on their size, weight, and operating characteristics.

The results of the average hourly traffic volume analysis for the travel directions recorded at Cameras 01 and 02 are illustrated in detail in Figures 17 and 18, respectively.

Specifically, Figure 17 presents data on the traffic volume of vehicle types and the corresponding converted car traffic volume at Camera 01. The data show significant fluctuations during peak hours, especially from 7:00 to 9:00 and from 16:00 to 18:00 for the straight travel direction. During this period, traffic volume increases sharply, reflecting the high vehicle density resulting from increased travel demand during morning and evening rush hours. For the left turn direction, traffic volume recorded two distinct peaks around 10:00 and 14:00, and there was a sudden increase during peak hours from 16:00 to 18:00. This variation shows different traffic behavior between travel directions, which may be related to factors such as work or study routes or socioeconomic activities in the area. Similarly, Figure 18 shows the average traffic volume data of converted vehicles and cars at Camera location 02.

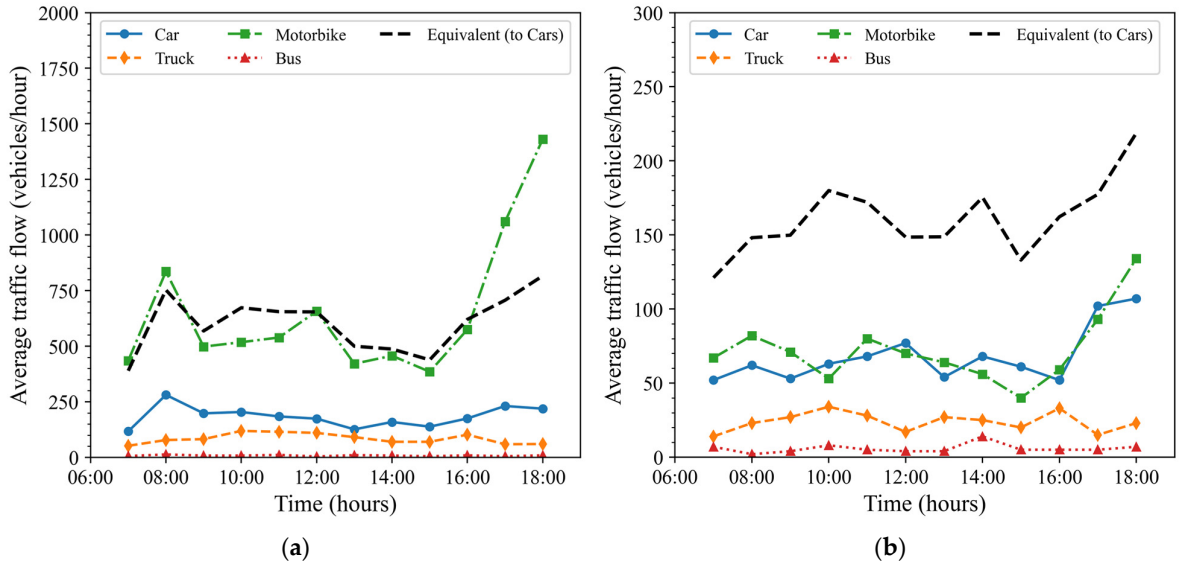


Figure 17. Hourly average traffic flow of passenger car units (PCUs) and other vehicle types from Camera 01 data: (**a**) Straight direction; (**b**) Left-turn direction.

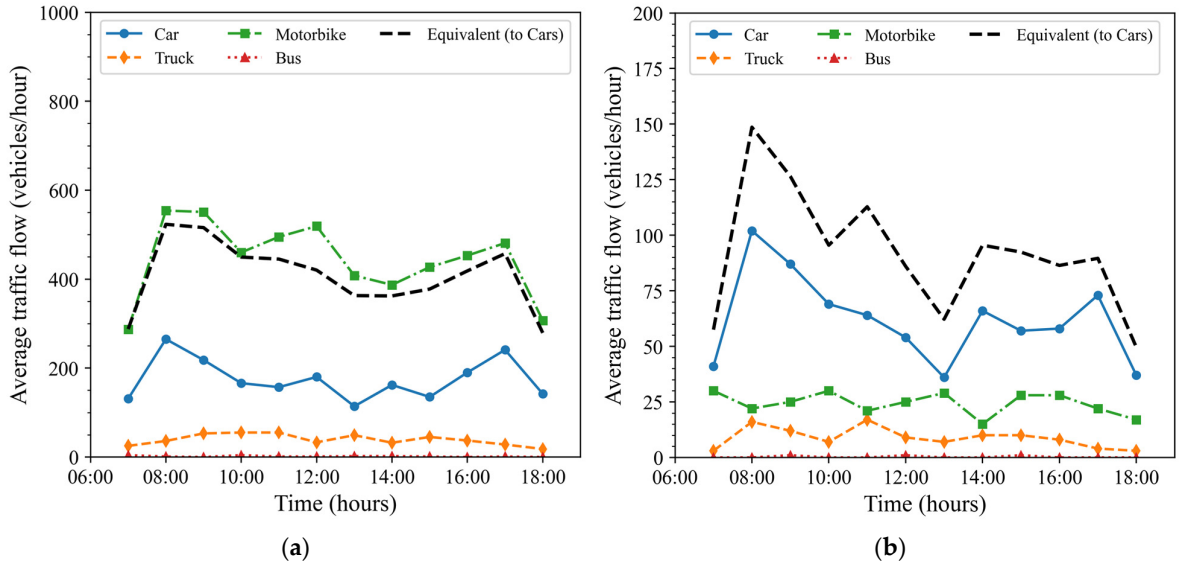


Figure 18. Hourly average traffic flow of passenger car units (PCUs) and other vehicle types from Camera 02 data: (**a**) Straight direction; (**b**) Left-turn direction.

Using the 12-h average traffic data from two camera locations, traffic volumes in other directions at the intersection were estimated as percentages based on the main recorded

flows. This dynamic data was input into the SUMO simulation, using the pre-trained 5M-Step 3DQN model to evaluate its performance under real traffic conditions. A fixed-signal control model was also simulated for comparison. For simplicity, both models assumed a CV penetration rate of 1.0.

Figure 19 compares vehicle delays between the two models. The graph illustrates a comparative analysis of average vehicle delay (in minutes) across various hours of the day between two traffic control methods: 3DQN control (in red with circles) and fixed-signal control (in blue with squares). The X-axis represents different time intervals throughout the day (though exact times are not labeled, the intervals are consistent). The Y-axis shows the average vehicle delay in minutes.

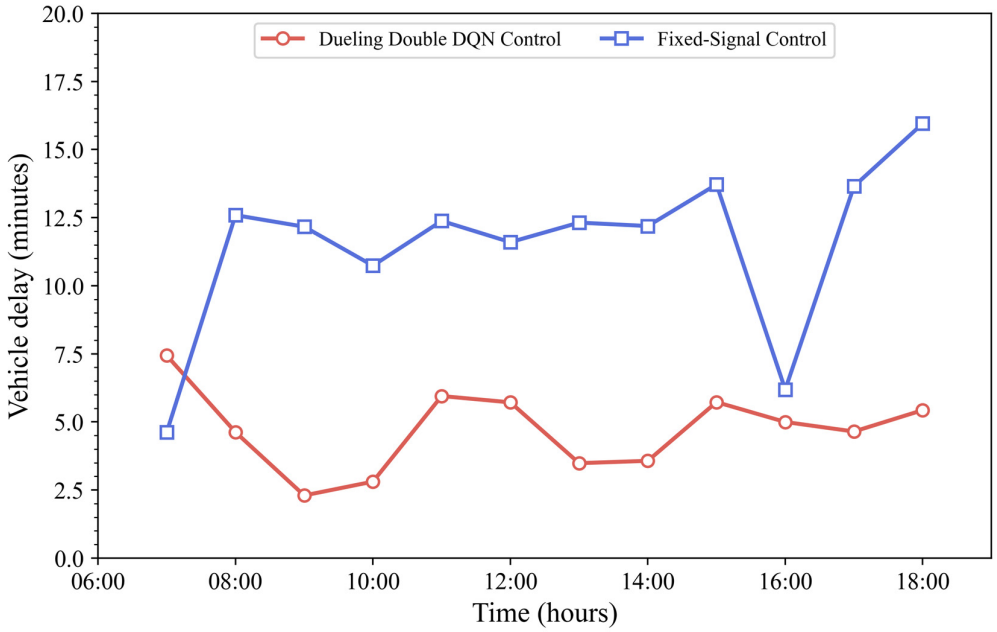


Figure 19. A comparison of average vehicle delay controlled by the 3DQN and fixed-signal control models using SUMO simulation.

The 3DQN control consistently achieves lower vehicle delay than the fixed-signal control at all time intervals. The fixed-signal control shows greater variability and higher peaks, with delays exceeding 12 min during some intervals. In contrast, the 3DQN-based model keeps delays mostly under 6 min, with relatively stable performance. The performance gap is especially pronounced during periods of peak congestion (e.g., rightmost section of the graph), where the fixed control experiences a sharp increase, while the 3DQN model maintains low delay.

At 06:00, the 3DQN model showed delays of 7–8 min, while fixed-signal control delays were under 5 min. This matched the rising morning traffic recorded by Camera 02. As traffic increased further, the 3DQN delay dropped to 2 min by 10:00, while the fixed-signal control delay spiked to 12.5 min, remaining high until 10:00. From 10:00 to 12:00, the 3DQN delay increased to 5 min and peaked again at 6 min between 14:00 and 15:00, likely due to midday traffic. The fixed-signal delay also rose, reaching 14 min. After 15:00, both models experienced a decline; however, 3DQN remained stable for approximately 5 min, while the fixed-signal control dropped to 6 min. During the evening peak (16:00–18:00), 3DQN delays remained steady at 4–5 min. In contrast, fixed-signal control delays increased sharply to over 15 min, consistent with the high traffic recorded by both cameras.

Overall, the 3DQN model consistently handled changing traffic better, keeping delays lower and more stable throughout the day, especially during peak periods. Its perfor-

mance reflects better adaptability to fluctuating traffic flows compared to the Fixed-signal control method.

The findings of this study are consistent with previous research [12,26–29], which demonstrated the potential and significant benefits of RL in improving traffic signal control under dynamic traffic conditions. A novel contribution of this study is the investigation of the effects of dynamic traffic flows on the performance of 3DQN, an aspect that has not been thoroughly addressed in prior studies. The input traffic data for the SUMO-based 3DQN model were collected from realistic camera footage using the YOLOv8 and SORT models for vehicle detection and tracking. As a result, the 3DQN model applied to a signalized intersection in Danang City, effectively reduced vehicle delays, particularly for longer waiting times, by dynamically adapting traffic signal control to real-time traffic flow.

However, the study has several limitations. The computational cost and training time required for the 3DQN model would be a significant hurdle when scaling the approach to a larger traffic network. In addition, the collected dynamic traffic data are limited, as it relies solely on data from two installed cameras, requiring assumptions about traffic flows in other directions. To overcome this issue, cameras covering all directions at signalized intersections need to be installed to provide comprehensive and accurate data, enabling the 3DQN model to make timely decisions and reduce vehicle delays during peak hours. For future research, exploring hierarchical RL or multi-agent reinforcement learning models could help overcome these challenges and improve coordination across multiple intersections. Furthermore, integrating real-time traffic data from connected vehicles and sensor networks could enhance the model's adaptability and efficiency in real-world applications. Future studies should also prioritize balancing model complexity with computational efficiency to facilitate the practical implementation of RL-based traffic signal control systems in urban environments.

6. Conclusions

This study employed the 3DQN algorithm in conjunction with the SUMO platform to optimize traffic signal control at real-world intersections in Danang City, Vietnam. By integrating dynamic and stochastic traffic flows derived from actual traffic data, the study evaluated the performance of the 3DQN model against a conventional fixed-time signal control approach. The key findings are summarized as follows:

1. Effectiveness in Reducing Vehicle Delay: The 3DQN model demonstrated superior performance in minimizing vehicle delays compared to the fixed-time signal control method. Both versions of the model—trained with 1 million (1M-Step) and 5 million (5M-Step) iterations—substantially decreased the frequency of prolonged waiting times (over 5 min). Notably, the 5M-Step model achieved a higher frequency of shorter delays (750 instances under 5 min) relative to the 1M-Step model (550 instances), underscoring the model's effectiveness in alleviating congestion.
2. Impact of Training Duration: The number of training steps critically influenced model stability and performance. The 5M-Step model consistently outperformed the 1M-Step model, particularly under fluctuating traffic conditions, indicating that extended training enhances the model's adaptability and reliability in complex environments.
3. Influence of CV Penetration: The CV penetration rate played a significant role in optimizing traffic signal control. The 5M-Step model successfully reduced average vehicle delays to below 5 min when the CV penetration rate reached just 20%, whereas the 1M-Step model required a penetration rate exceeding 40% to achieve similar results. Both models converged to optimal performance when the CV penetration rate exceeded 70%.

4. Flexibility Under Dynamic Traffic Conditions: Unlike fixed-time control methods, which are rigid and unable to adapt to real-time fluctuations, the 3DQN model demonstrated robust adaptability. This flexibility enabled more efficient traffic signal adjustments in response to variable traffic patterns, contributing to improved overall flow.
5. Integration of Real-Time Data: The incorporation of real-time mixed traffic flow data, obtained using the YOLOv8 object detection model and the SORT tracking algorithm, further enhanced the performance of the 3DQN model. This integration allowed the system to make timely and precise signal adjustments, yielding substantial reductions in vehicle delays during peak hours and better responsiveness to real-world traffic variability.

In conclusion, the 3DQN model presents a promising approach to intelligent traffic signal control, capable of reducing congestion and improving traffic management efficiency under dynamic, real-world conditions. The model's ability to integrate real-time data and adapt to variable traffic scenarios positions it as a valuable solution for next-generation urban traffic control systems. Future work may explore scaling the model across larger traffic networks and incorporating multi-agent reinforcement learning to further enhance system-wide coordination.

Author Contributions: Conceptualization, T.C.P. and V.D.L.; methodology, T.C.P. and V.D.L.; software, V.D.L. and T.N.; validation, T.C.P., V.D.L. and T.N.; formal analysis, T.C.P.; investigation, V.D.L.; resources, T.C.P.; data curation, T.N.; writing—original draft preparation, T.C.P.; writing—review and editing, T.N.; visualization, V.D.L.; supervision, T.C.P.; project administration, T.C.P.; funding acquisition, T.C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Office of National Science and Technology Research Programs grant funded by the Vietnamese Government, Ministry of Science and Technology (No. KC.01.02/21-30).

Data Availability Statement: Data available on request due to privacy/ethical restrictions.

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

References

1. General Statistics Office of Vietnam. *Population and Housing Census 2019: Summary Results*; Ministry of Planning and Investment: Hanoi, Vietnam, 2019.
2. Webster, F.V. Traffic Signal Settings. In *Road Research Technical Paper*; HMSO: London, UK, 1958.
3. Miller, A.J. Settings for Fixed-Cycle Traffic Signals. *Oper. Res. Soc.* **1963**, *14*, 373–386. [CrossRef]
4. Dion, F.; Rakha, H.; Kang, Y.-S. Comparison of Delay Estimates at Under-Saturated and Over-Saturated Pre-Timed Signalized Intersections. *Transp. Res. Part B* **2004**, *38*, 99–122. [CrossRef]
5. Porche, I.; Lafortune, S. Adaptive Look-Ahead Optimization of Traffic Signals. *J. Intell. Transp. Syst.* **1999**, *4*, 209–254.
6. Cools, S.-B.; Gershenson, C.; D'Hooghe, B. Self-Organizing Traffic Lights: A Realistic Simulation. In *Advances in Applied Self-Organizing Systems*; Springer: London, UK, 2013; pp. 45–55.
7. Feng, Y.; Wu, Y. Environmental Adaptive Urban Traffic Signal Control Based on Reinforcement Learning Algorithm. *J. Phys. Conf. Ser.* **2020**, *1650*, 032097. [CrossRef]
8. Pan, G.; Muresan, M.; Fu, L. Adaptive Traffic Signal Control Using Deep Q-Learning: Case Study on Optimal Implementations. *Can. J. Civ. Eng.* **2023**, *50*, 488–497. [CrossRef]
9. Kumar, R.; Sharma, N.V.K.; Chaurasiya, V.K. Adaptive Traffic Light Control Using Deep Reinforcement Learning Technique. *Multimed. Tools Appl.* **2024**, *83*, 13851–13872. [CrossRef]
10. Tian, D.; Wei, Y.; Zhou, J.; Zheng, K.; Duan, X.; Wang, Y.; Hui, R.; Guo, P. Swarm Intelligence Inspired Adaptive Traffic Control for Traffic Networks. In *Advances in Swarm Intelligence*; Springer: Cham, Switzerland, 2018; Volume 221, pp. 3–13.
11. Mohamed, M.N.; Essawy, Y.A.; Hosny, O. Traffic Signal Optimization Using Genetic Algorithms. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2024; Volume 496, pp. 101–113.
12. Pan, T. Traffic Light Control with Reinforcement Learning. *Adv. Educ. Technol.* **2024**, *43*, 26–43. [CrossRef]

13. Balaji, P.G.; German, X.; Srinivasan, D. Urban Traffic Signal Control Using Reinforcement Learning Agents. *IET Intell. Transp. Syst.* **2010**, *4*, 177. [CrossRef]
14. Al-Rawi, H.A.A.; Ng, M.A.; Yau, K.-L.A. Application of Reinforcement Learning to Routing in Distributed Wireless Networks: A Review. *Artif. Intell. Rev.* **2015**, *43*, 381–416. [CrossRef]
15. Mnih, V.; Hambly, K.B.; Xu, R.; Yang, H. Recent Advances in Reinforcement Learning in Finance. *Math. Financ.* **2023**, *33*, 437–503.
16. Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; Petersen, S.; et al. Human-Level Control Through Deep Reinforcement Learning. *Nature* **2015**, *518*, 529–533.
17. Park, B.; Lee, J.; Kim, T.; Har, D.S. Kick-Motion Training with DQN in AI Soccer Environment. *arXiv* **2022**, arXiv:2212.00389.
18. Li, L.; Lv, Y.; Wang, F. Traffic Signal Timing via Deep Reinforcement Learning. *IEEE/CAA J. Autom. Sin.* **2016**, *3*, 247–254. [CrossRef]
19. Chu, T.; Wang, J.; Codeca, L.; Li, Z. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1086–1095. [CrossRef]
20. Skuba, M.; Janota, A.; Kuchár, P.; Malobický, B. Deep Reinforcement Learning for Traffic Signal Control. *Transp. Res. Proc.* **2023**, *74*, 954–958. [CrossRef]
21. Wu, C.; Kim, I.; Ma, Z. Deep Reinforcement Learning Based Traffic Signal Control: A Comparative Analysis. *Procedia Comput. Sci.* **2023**, *220*, 275–282. [CrossRef]
22. Prajapati, M.; Upadhyay, A.K.; Patil, H.; Dongradive, D.J. A Review of Deep Reinforcement Learning for Traffic Signal Control. *Int. J. Multidiscip. Res.* **2024**, *6*, 11650.
23. Zheng, Q.; Xu, H.; Chen, J.; Zhang, K. Multi-Dimensional Double Deep Dynamic Q-Network with Aligned Q-Fusion for Dual-Ring Barrier Traffic Signal Control. *Appl. Sci.* **2025**, *15*, 1118. [CrossRef]
24. Chu, X.; Cao, X. Adaptive traffic signal control for road networks based on dueling double deep q-network. In Proceedings of the International Conference on Frontiers of Traffic and Transportation Engineering (FTTE 2024), Lanzhou, China, 22–24 November 2024; SPIE: Bellingham, WA, USA, 2025; Volume 13645.
25. Zhang, H.; Fang, Z.; Chen, Y.; Dai, H.; Jiang, Q.; Zeng, X. Traffic signal optimization control method based on attention mechanism updated weights double deep Q network. *Complex Intell. Syst.* **2025**, *11*, 217. [CrossRef]
26. Zhang, R.; Leteurtre, R.; Striner, B.; Alanazi, A.; Alghafis, A.; Tonguz, O. Partially Detected Intelligent Traffic Signal Control: Environmental Adaptation. *arXiv* **2019**, arXiv:1910.10808.
27. Wei, H.; Zheng, G.; Yao, H.; Li, Z. Intellilight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In Proceedings of the KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018.
28. Ducrocq, R.; Farhi, N. Deep Reinforcement Q-Learning for Intelligent Traffic Signal Control with Partial Detection. *Int. J. Intell. Transp. Syst. Res.* **2023**, *21*, 192–206. [CrossRef]
29. Cardoso, R.F.D. Intelligent Traffic Light Management for Automated Guided Vehicle Systems Using Deep Reinforcement Learning. Master's Thesis, Universidade do Porto, Porto, Portugal, 2023.
30. Nguyen, C.; Farhi, N. Estimation of Urban Traffic State with Probe Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2797–2808. [CrossRef]
31. Zhang, R.; Ishikawa, A.; Wang, W.; Striner, B.; Tonguz, O. Using Reinforcement Learning with Partial Vehicle Detection for Intelligent Traffic Signal Control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 404–415. [CrossRef]
32. Ministry of Transport of Vietnam. *Journey-Tracking Devices and Regulations on Their Installation, Management, and Data Usage for Commercial Transport Vehicles. Decree 10/2020/ND-CP and Circular 12/2020/TT-BGTVT*; Ministry of Transport: Hanoi, Vietnam, 2020.
33. People's Committee of Danang City. *Regulation, Tasks, and Competition Plan for the Planning and Architectural Design of the Intersection Cluster at Lethanhnghi—Cachmangthang 8—Thang Long—Hoaxuan Bridge Approach Road*; Da Nang City Government: Danang, Vietnam, 2024.
34. Krajzewicz, D. Traffic Simulation with SUMO Simulation of Urban Mobility. In *Advances in Intelligent and Soft Computing*; Springer: Berlin, Germany, 2010; Volume 145, pp. 269–293.
35. Kim, J.H.; Kwon, J.H. A Study on Traffic Signal System Improvement Using SUMO (Simulation of Urban Mobility). *J. Korean Soc. Geospat. Inf. Syst.* **2024**, *42*, 265–274. [CrossRef]
36. Explore Ultralytics YOLOv8. Available online: https://yolov8.com/ (accessed on 13 June 2025).
37. Bansal, A. Traffic Counting Program Using YOLOv8. Github. Available online: https://github.com/anujeshify/Traffic-Counting-Program-using-YOLOv8 (accessed on 13 June 2025).
38. Bewley, A.; Zongyuan, G.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.