IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 1, February 2025, pp. 500~515
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i1.pp500-515

Enhancing traffic flow through multi-agent reinforcement learning for adaptive traffic light duration control

Nada Faqir[1], Jaouad Boumhidi[1], Chakir Loqman[1], Youness Oubenalla[2]
[1]Department of Computer Science, LISAC Laboratory, Faculty of of Sciences Dhar El Mehraz, Sidi Mohammed Ben Abdellah University, Fes, Morocco
[2]Department of Computer Science, Faculty of Sciences and Technologies, Sidi Mohammed Ben Abdellah University, Fes, Morocco

Article Info

Article history:

Received Mar 15, 2024
Revised Oct 11, 2024
Accepted Oct 18, 2024

Keywords:

Deep Q-network
Deep reinforcement learning
Multi-agent
Q-learning
Traffic signal control

ABSTRACT

This study addresses urban traffic congestion through deep learning for traffic signal control (TSC). In contrast to previous research on single traffic light controllers, our approach is tailored to the TSC challenge within a network of two intersections. Employing convolutional neural networks (CNN) in a deep Q-network (DQN) model, our method adopts centralized training and distributed execution (CTDE) within a multi-agent reinforcement learning (MARL) framework. The primary aim is to optimize traffic flow in a two-intersection setting, comparing outcomes with baseline strategies. Overcoming scalability and partial observability challenges, our approach demonstrates the efficacy of the CTDE-based MARL framework. Experiments using urban mobility simulation (SUMO) exhibit a 68% performance enhancement over basic traffic light control systems, validating our solution across diverse scenarios. While the study focuses on two intersections, it hints at broader applications in complex settings, presenting a promising avenue for mitigating urban traffic congestion. The research underscores the importance of collaboration within MARL frameworks, contributing significantly to the advancement of adaptive traffic signal control (ATSC) in urban environments for sustainable transportation solutions.

Corresponding Author:

Nada Faqir
Department of Computer Science, Faculty of Sciences Dhar El Mehraz
Sidi Mohammed Ben Abdellah University of Fes
Fes 30000, Morocco
Email: nada.faqir@usmba.ac.ma

1. INTRODUCTION

Traffic congestion is a pervasive issue in urban areas worldwide, causing significant economic losses due to the total time lost by commuters. These economic losses are estimated to reach billions of dollars annually in countries suffering from traffic congestion. As a result, addressing the economic impact of alleviating traffic congestion has become a crucial priority. Managing traffic congestion has emerged as a pressing and unsolved problem in numerous urban areas around the world.

The traditional static traffic light control system (TLCS) has long been the conventional solution for managing traffic flow. However, with the increasing complexity and scale of urban transportation networks, traditional traffic light control system is no longer adequate in effectively mitigating traffic congestion. To overcome these limitations, researchers have turned to deep reinforcement learning frameworks, which leverage the power of reinforcement learning algorithms and neural networks such as deep neural networks (DNN) and convolutional neural networks (CNN) [1]. Notable advancements in this field include systems like intellilight, which utilize deep reinforcement learning for adaptive traffic signal control (ATSC), demonstrating significant improvements in traffic efficiency [2]. In addressing urban traffic congestion, our focus lies in optimizing traffic flow efficiency within a two-intersection scenario. We emphasize the use of CNN in our deep Q-network (DQN) model and employ a centralized training and distributed execution (CTDE) approach within a multi-agent reinforcement learning (MARL) framework [3].

This article proposes a novel multi-agent perspective to enhance the performance of traffic light control systems, recognizing the importance of agent collaboration in supporting efficient traffic flows [4]. By adopting a MARL approach, our goal is to develop a system where multiple agents regulate traffic at adjacent intersections, collaborating to optimize traffic flow. We acknowledge the potential for broader applications in more complex settings, and our work aims to address scalability and partial observability challenges in large-scale multi-agent deep reinforcement learning systems [5]. To validate the effectiveness of our approach, we conducted several experiments using traffic simulations involving two intersecting roads. These experiments aimed to observe how agents achieve collaboration and improve traffic flow through independent deep Q network (IDQN) techniques.

By addressing the economic impact of traffic congestion and introducing a novel multi-agent perspective, our proposed approach offers a promising solution for optimizing traffic flow in complex urban environments. Through the integration of real-time data, scalability, and considerations for mixed traffic scenarios, we aim to improve overall traffic efficiency and create smarter, more sustainable cities. This research provides valuable insights into the challenges and potential solutions for managing traffic congestion using deep reinforcement learning techniques.

Numerous studies have demonstrated the effectiveness of reinforcement learning in adaptive traffic light control, with a particular focus on single traffic light controllers [6], [7]. However, when addressing multi-agent systems, the challenge arises in determining whether traffic control should be approached as a collaborative or competitive problem [8]. In such environments, the conflict between local and global optima, coupled with the interdependence of agent actions, adds complexity to the dynamics of transport networks. Additionally, factors such as driver behavior and co-evolution further complicate the challenges of traffic light control in urban environments.

Our work contributes to this landscape by delving into the complexities of traffic signal control (TSC) in networks with multiple intersections. Introducing a novel MARL framework, our approach emphasizes collaboration among agents to enhance traffic flow. It capitalizes on the advantages of independent traffic light control, ensuring stable operations even in the event of failures in other components. To enable efficient decision-making, we employ CNNs to model our agents and incorporate a comprehensive representation of the state of adjacent intersections. While existing studies [6], [7] have predominantly focused on single traffic light controllers, our work extends the scope to multi-intersection scenarios, presenting a promising approach for addressing traffic congestion in urban environments. The outlined comparisons aim to underscore the innovative aspects of our proposed system within the broader landscape of traffic light control methodologies.

In the domain of deep reinforcement learning, a DNN typically represents the Q function. Value-based methods, such as DQN [9], utilize DNN to extract state representations directly from the environment. The DQN outputs Q values for all possible actions, enabling the approximation of optimal strategies. To enhance the performance, stability, and effectiveness of the DQN agent, two commonly employed mechanisms, namely the target network and replay memory, are integrated during training.

Our exploration aligns with this approach, and we further contribute by examining various DQN-based models, including double deep Q-network (DDQN) [10], dueling DQN [11], DDQN with proportional prioritization based on DDQN [12], and dynamic frame skip deep Q-network (DFDQN) [13]. While artificial neural networks (ANN) have been widely used as model architectures [14], [15], there is a growing utilization of CNN in this context [16]–[19]. By integrating CNN-based models and examining their impact on the learning quality of agents, our paper contributes to advancing the understanding of deep reinforcement learning in traffic light control. The findings shed light on the strengths and limitations of different neural network architectures, providing insights into their effectiveness in optimizing traffic flow.

2. THE COMPREHENSIVE THEORETICAL BASIS AND THE PROPOSED METHOD

2.1. Single-agent reinforcement learning

Figure 1 illustrates how a single agent in reinforcement learning learns by interacting with a dynamic environment using a trial-and-error approach. The goal is to derive the best possible policy by maximizing the expected total of cumulative rewards. In a reinforcement learning problem, the agent iteratively observes its environment, and then it takes an action. Consequently, he receives from the environment a reward following his chosen action; his objective is to accumulate the rewards in the long term. Then according to its rules and the state transition probability, the environment moves to the next state. The foundation of reinforcement learning is the Markov decision process, a stochastic model described by the tuple (S, A, R, P). Where S is the states set and $s_t \in S$ the state of the environment at the time-step t, A the actions set $a_t \in A$ the chosen action at the time-step t by the agent, $r_t \in R$ the immediate reward value received by the agent in the state s_t performing action a_t and finally P the state transition probability.

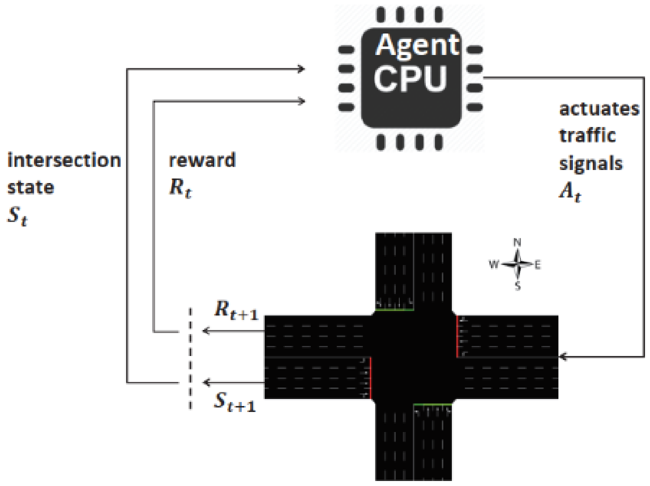

Figure 1. Single agent reinforcement learning for TSC

In traffic signal problems, the agent interacts with the intersection at discrete time steps t = 0, 1, 2, first observing the state of the intersection St at the start of the time step t, then selects an action At depending on a policy π. After the vehicle moves under the active traffic light, the state of the intersection changes to the new state of St+1. At the end of the time step t, the agent receives a reward Rt following its decision to choose a signal. Assuming that the reward at each time step t must be multiplied by the discount factor γ ϵ [0, 1] to determine the importance given to immediate and future rewards, the cumulative reward from time step t to the end of time T is expressed as follows: $R_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$. The value function Q_π(s,a) represents the action A_t taken in the current state S_t, followed by the application of policy π until the episode concludes. During this process, the cumulative rewards earned by the agents can be expressed as (1).

$$Q_\pi(s,a) = E[R_t | S_t = s, A_t = a] \tag{1}$$

The agent's objective is to identify the optimal policy for selecting actions, represented as $\pi*$ in (2), which maximizes the cumulative future reward or Q-values and enables the agent to accomplish its objective.

$$\pi^* = argmax_\pi\, Q_\pi(s,a)\ for\ all\ s \in S, a\ \in A \tag{2}$$

The Q-values that correspond to the optimal policy π^* are denoted as $Q^*(s,a) = Q_{\pi^*}(s,a)$ and can be approximated using linear function approximators. The primary objective is to identify the optimal policy π^* and subsequently, determine the optimal Q-values $Q^*(s,a)$. The recursive relation for the optimal Q-values $Q^*(s,a)$ is provided by Bellman's optimality equation.

$$Q^*(s,a) = E[R_t + \gamma Q^*(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{3}$$

Linear function approximators, as well as nonlinear function approximators such as DNN, can be employed to estimate action-value functions or policy.

2.2. Multi-agent reinforcement learning

MARL is built around a stochastic game that relies on a Markov decision process. This process is defined by the tuple $S, A_1, \dots, A_n, R_1, \dots, R_n, P$, where n represents the number of agents involved., as shown in Figure 2. The joint action space $A = A_1 \times \dots \times A_n$ and $R_1, \dots, R_n$ represent the rewards of each agent. In multi-agent environments, state transitions arise from the collective actions of all agents, and each agent's rewards are influenced by the joint policy denoted as $H: S \times A \rightarrow [0,1]$. The rewards for each agent is calculated as follows: $R_i^H = E\left[R_{t+1} \middle| S_t = s, A_{t,i} = a, H\right]$. Consequently, the Bellman equation can be expressed as (4).

$$Q_i^H(s,a) = E_i^H[R_t + \gamma Q_i^H(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \quad (4)$$

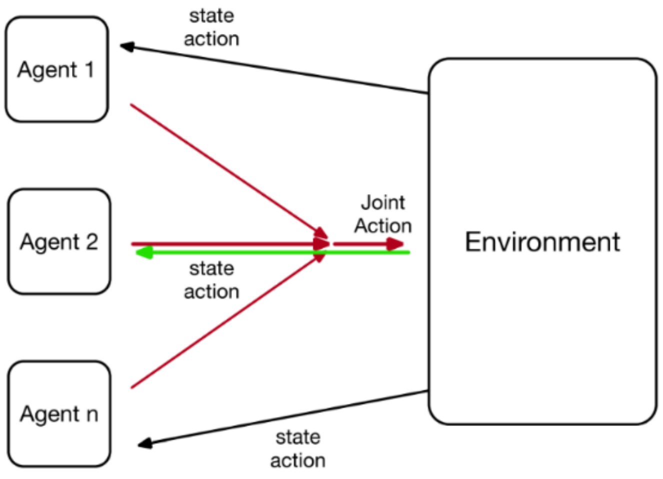

Figure 2. Agents' interaction with the environment

Stochastic games can be categorized into three types: fully cooperative, fully competitive, and mixed games. In fully cooperative games, all agents share the same reward function, aiming to maximize the total reward collectively. Cooperation among agents is essential to reach this objective. Approaches based on the Q-learning algorithm have been developed to tackle coordination challenges, focusing on a single optimal joint action [20]. Lauer's distributed Q-learning algorithm [21] addresses the issue while maintaining low computational complexity. However, this algorithm only applies to deterministic scenarios with non-negative reward functions.

When dealing with a perfectly competitive game, where there are two agents with opposing goals and R1 = -R2, most of the literature on reinforcement learning focuses on the two-agent scenario. The minimax principle is applied in such a situation, which aims to maximize an agent's profit while considering the worst-case scenario, where the opponent will always try to minimize it. To achieve this, adversary-independent algorithms are used, such as the minimax-Q algorithm [20], [22], which computes stage game strategies and values using the minimax principle and propagates values through state pairs-stock using a time-difference rule similar to Q learning.

Mixed stochastic games involve agents with different and correlated rewards, where no constraints are imposed on the agents' reward functions. This model is suitable for self-interested agents, but even cooperating agents may encounter conflicts of interest, such as when competing for resources. Game theory concepts such as equilibrium play a significant role in algorithms for mixed stochastic games. Single-agent reinforcement learning to MARL methods make no assumptions about the task and can be applied to mixed stochastic games, along with other techniques such as agent-independent, agent-tracking, and agent-aware methods. However, there is no guarantee of success in these scenarios.

Setting learning goals for agents can be challenging since it is difficult to define good general objectives. The two primary aspects of learning goals are stability and adaptability. Stability refers to an agent's ability to converge to a stable policy, while adaptability ensures that an agent's performance does not decline when other agents change their strategies [23], [24].

3. METHOD

In this study, we introduce an innovative approach to address the persistent challenge of urban traffic congestion using advanced artificial intelligence techniques. The primary objective is to optimize traffic flow at complex intersections by integrating CNN into a DQN model within a MARL framework. Our approach leverages a CTDE architecture, crucial for scalability and effective coordination among multiple agents controlling traffic signals.

To formulate the problem, the primary aim is to enhance traffic flow optimization in a scenario with each intersection's state is discretized into irregularly shaped cells using discrete traffic state encoding (DTSE). These cells represent different zones monitored by the agents, capturing vehicle presence and traffic dynamics. The agent's state representation, denoted as two intersections, and we benchmark our results against baseline approaches. Our innovative approach leverages data such as vehicle positions, queue lengths, and current traffic light states. This detailed situational awareness enables agents to make informed decisions to optimize traffic flow.

The action space consists of four distinct actions per agent: adjusting traffic lights to prioritize north/south or east/west traffic, encompassing both straight and turning movements. Actions are selected based on the agent's observation of the current state aimed to minimize cumulative waiting times. The reward function across all vehicles at the intersection is defined as the reduction in cumulative waiting time from t-1 to t, emphasizing the agent's role in improving traffic efficiency over time.

The CTDE framework facilitates collaborative action selection among agents by sharing their last chosen actions, enhancing by sharing their last chosen actions, enhancing coordination and traffic synchronization between adjacent intersections. This sharing mechanism optimizes traffic flow across interconnected road networks, crucial for mitigating congestion hotspots and improving overall urban mobility. To evaluate the effectiveness of our approach, we simulate realistic traffic scenarios using the Weibull distribution: high-traffic (4,000 cars), low-traffic (600 cars), NS-traffic (2,000 cars, 90% north/south), and EW-traffic (2,000 cars, 90% east/west). These scenarios simulate diverse traffic conditions to robustly test the adaptability and performance of our CTDE-based MARL framework under varying load and directional traffic distributions. Through rigorous experimentation and simulations, we validate the efficacy of our approach we validate the effectiveness and efficiency of our CTDE-based MARL framework in enhancing traffic management strategies. Performance metrics such as average waiting times, intersection throughput, and overall traffic flow efficiency provide quantitative insights into the framework's capability to optimize urban traffic operations effectively.

3.1. Environment design

The test environment for evaluating our approach consists of two neighboring intersections, as illustrated in Figure 3. We intentionally opted for this minimalist simulation setup to ensure a straightforward implementation process and minimize unnecessary complexities. The simulation of urban mobility (SUMO) tool was utilized for modeling and simulating the urban traffic scenarios [25].

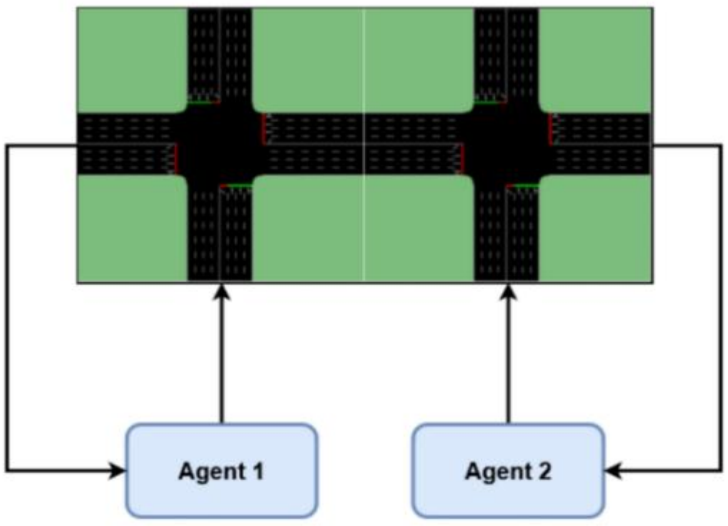

Figure 3. Environment of the simulation

Our environment is composed of an integrated traffic light system where each 4-arm intersection (shown in Figure 4) consists of eight lanes each (four inbound lanes and four outbound lanes). Each arm is 750 meters long from the vehicle access point to the intersection traffic lights. Each vehicle chooses its incoming lane according to its destination. The possible directions for each vehicle are explained as follows:

– Go straight using the two central lanes and the one on the far right.
– Turn right only using the rightmost lane.
– Turn left only using the leftmost lane.

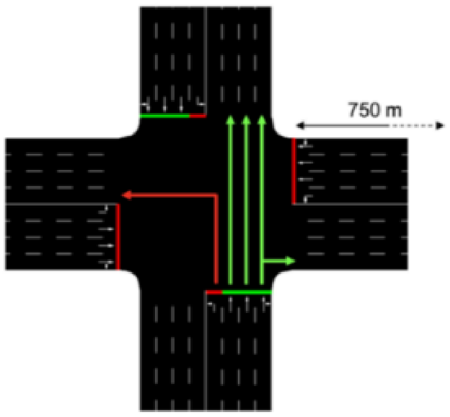

Figure 4. Details of the possible directions of each intersection

The traffic lights regulating each intersection arise from the following equation and shown in Figure 5. As in real life, the traffic lights are either red, yellow, or green and at each time step, the traffic light is either yellow or green in phase. The color phase transition is cyclical (red-green-yellow-red) and mandatory to clear the center of the intersection and to avoid probable accidents. The duration of each phase is fixed, 10 seconds for the green time and 4 seconds for the yellow time. The red phase, therefore, lasts the time elapsed since the previous phase change.

$$TL = \{TL_N, TL_{NL}, TL_S, TL_{SL}, TL_E, TL_{EL}, TL_W, TL_{WL}\} \tag{5}$$

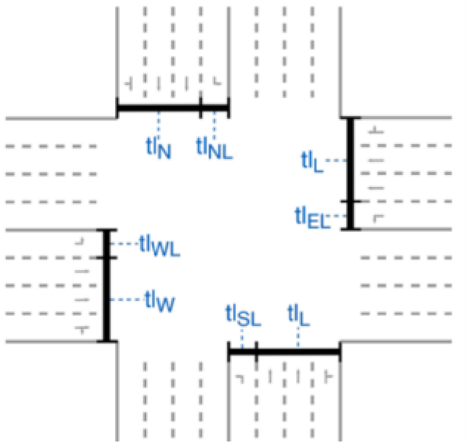

Figure 5. Traffic lights for each inbound lane of each intersection in the simulated environment

3.2. Agents

In the context of MARL, ATSC involves coordinating the actions of multiple intersection traffic light controllers to achieve network-wide traffic optimization. Each intersection traffic light controller acts as an agent within the MARL framework, perceiving traffic conditions at their respective intersections and making decisions on phase control signals accordingly. Through interaction and collaboration, agent controllers adapt their policies to changing traffic dynamics, leveraging collective network intelligence to optimize traffic flow across the entire system. This collaborative approach allows officers to learn from each other's experiences, improve decision-making processes, and ultimately improve the efficiency of the TSC system. By using MARL in ATSC, the network-wide control loop shown in Figure 6, facilitates the coordination and synchronization of agent actions, leading to a more efficient and adaptive traffic management solution. This approach not only considers local optimization at each intersection, but also considers the overall impact of signal control decisions, resulting in improved traffic flow, reduced congestion, and improved traffic overall performance of the transport network.

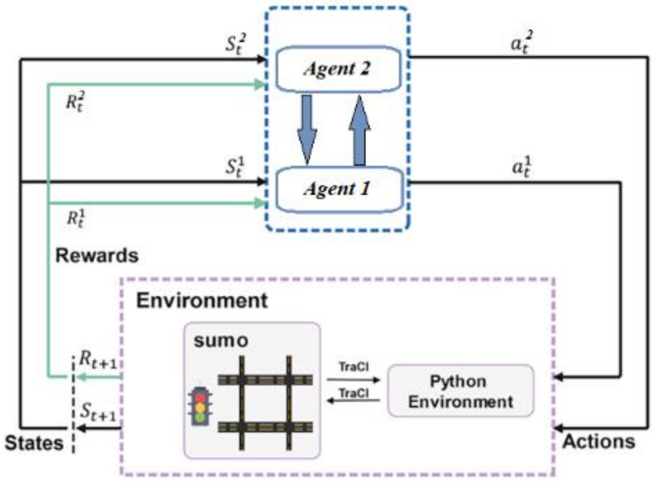

Figure 6. Network-wide MARL control loop: Policy optimization in ATSC as an interactive MDP process with 2 agents

Using deep Q-Learning, each agent in our proposed method retrieves the discretized representation of the environment state, denoted as St = $(P_t, C_t^c, \Delta C_t^c, V_t^c)$ (detailed in the next section). As our approach is based on the CTDE design of a multi-agent environment, knowledge sharing between the agents of the two intersections is crucial in a partially observable environment. To enable this knowledge sharing, at each time step t, each agent shares the last executed action at time step t-1 with the other agent. This information is provided as inputs to the CNN network, which estimates the Q-values $Q(S_t, a, \theta)$ for all possible actions in the action space, as shown in Figure 7.

Furthermore, using the replay memeory mechanism, each agent records the observed interaction experience $E_t = (S_t, A_t, R_t, S_{t+1})$ into a replay memory M $= \{E_1, E_2, \cdots, E_t\}$. To address the issue of oscillation during the training phase, we employ the target network mechanism, which utilizes the same architecture as the CNN network depicted in Figure 7. In addition to updating the CNN parameters, the agent also updates the target network parameters using a method called soft updating, as explained in section 3.6. The agent system structure of our proposed method and his training process are depicted in Figure 7, illustrating the integration of these components and the overall flow of information and updates within the system.

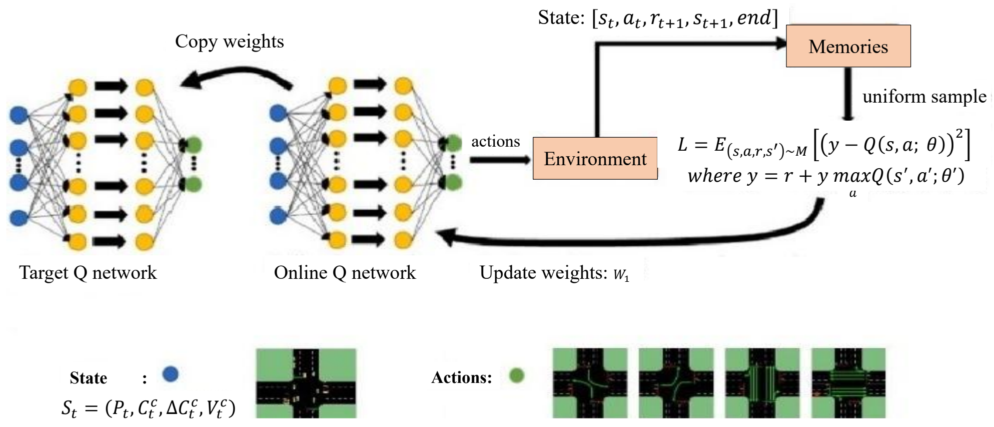
State: $[s_t, a_t, r_{t+1}, s_{t+1}, end]$
Copy weights
Memories
uniform sample
actions
Environment
$L = E_{(s,a,r,s')\sim M}\left[\left(y - Q(s,a;\theta)\right)^2\right]$
$where\ y = r + y\ \underset{a}{max}Q(s', a'; \theta')$
Target Q network
Online Q network
Update weights: w_1
State :
$S_t = (P_t, C_t^c, \Delta C_t^c, V_t^c)$
Actions:

Figure 7. Agent system and training process

3.3. State representation

To apply reinforcement learning, we need to formulate the problem as a Markov process, so we first need to define the state representation used by the agent. The state is the agent's perception of the environment at time step t and is commonly denoted by St. Choosing the right situational representation is important for the agent to learn how to optimize traffic effectively. This choice is also motivated by the quantity and quality of information provided to the agent.

Vidali *et al.* [19] used a discretized representation of each arm of a junction where each cell is irregular based on formula DTSE [14], except that less information is encoded. Not all cells are the same size, as shown in Figure 8. At each intersection, there are 20 cells per arm, with 10 cells allocated to the left lane and the remaining 10 cells covering the other three lanes, resulting in 80 cells per intersection. These cells are not of the same size and their length is determined by their distance from the stop line. It is important to strike a balance between cell length and computation complexity. If the cell is too long, cars may not be detected, while if it is too short, the computation required covering the path length increases. The paper proposes a cell length that is 2 meters longer than the car's length for the shortest cell closest to the stop line. Whenever the agent observes the environment, it receives a set of cells that indicate the presence of vehicles in each approach road zone.

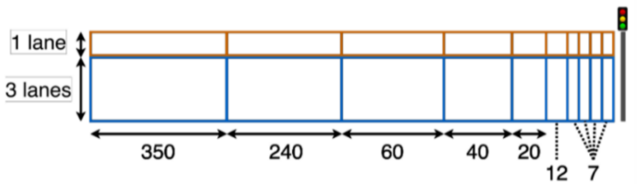

Figure 8. State representation design of an intersection arm with cell lengths

In our approach and for each intersection we retrieve the state from sumo, in the form of four metrics as follows:
- The phase P_t of the traffic signal at time step t;
- The congestion level C_t^c of each cell c, at time step t;
- The variation in congestion level ΔC_t^c for each cell c during a single step;
- The mean speed V_t^c of the vehicles of each cell c, at time step t.

Where congestion level C_t^c must be in the range of [0, 1]; A value of 1 is assigned when a vehicle is present, up to the maximum occupancy of the cell., is in cell c of each incoming lane of the intersection and 0 when no vehicle is in this cell. We define the congestion level as (6).

$$C_t^c = \frac{N_{veh}^c}{L^c(S_{veh}+g_{min})} \tag{6}$$

Where N_{veh}^c the number of vehicles in cell c, the length of the cell c L^c, S_{veh} the size of the vehicle veh and g_{min}. This refers to the minimum distance maintained between vehicles. (in SUMO S_{veh} = 5 meters and g_{min} = 2.5).

To identify congested lanes, the congestion level is employed, and if a lane has high congestion levels, it should shift to the traffic signal phase that allows traffic to proceed and reduces congestion. The status of neighboring traffic signals is determined using the change in congestion denoted as ΔC_t^c. If the congestion levels at the lanes of adjacent intersections increase, the signaling phase that permits traffic to enter the intersection. must be adopted. Each traffic signal can potentially achieve cooperative control with neighboring signals by considering the change in congestion level as a state. ΔC_t^c at time step t can be calculated by taking the difference between C_t^c and C_{t-1}^c then ΔC_t^c can be in the range [-1,1]. We normalize ΔC_t^c as (7).

$$\Delta C_t^c = \frac{1+C_t^c - C_{t-1}^c}{2} \tag{7}$$

The speed V_t^c is determined by normalizing the speed of each vehicle against the maximum speed allowed for the edge. In summary, the traffic signal agent in this environment establishes the state at a specific discrete time step t as St=($P_t, C_t^c, \Delta C_t^c, V_t^c$).

Our approach is based on the CTDE design of a multi-agent environment where the sharing of knowledge between the agents of the two intersections in a partially observable environment implies that each agent at time step t shares with the other the last action executed at time step t-1. The detail of the state of the environment that we have adopted also takes the partially observable nature of this environment. This is justified by the change congestion level metric ΔC_t^c added to the representation of the state.

3.4. Action space

In our case, we have four different actions that can be taken by each agent as shown in Figure 9.

- A1: Activate green lights for vehicles in the north and south arms going straight or turning right.
- A2: Activate green lights exclusively for vehicles coming from the north and south who want to turn left.
- A3: Activate green lights for vehicles that are in the east and west arm and wish to continue straight or turn right.
- A4: Activate green lights exclusively for vehicles in the east and west arm turning left.

When the action chosen at time step t differs from the one selected at time step t+1, a transition phase must be executed. This transition consists of activating a yellow light phase lasting 4 seconds, but only if the action at time step t is not the same as the previous one. In this case, there are 10 simulation steps separating two identical actions, as each simulation step in SUMO represents 1 second. If consecutive actions are different, the yellow light phase adds 4 more simulation steps, resulting in 14 steps between the two actions.

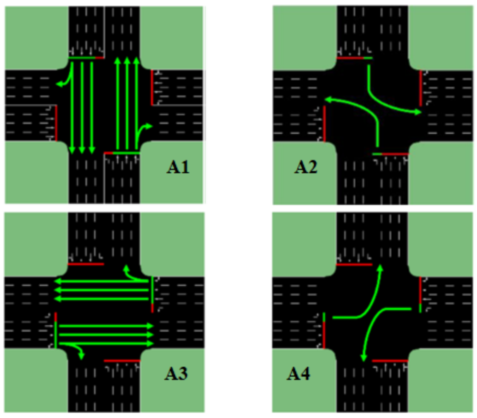

Figure 9. Possible actions for each agent

3.5. Reward definition

At the core of reinforcement learning lies the fundamental concept where an agent undertakes an action and subsequently garners feedback from the environment in the form of a reward, denoted as Rt. In our approach, we direct our attention specifically to negative rewards, emanating from suboptimal actions, with a primary objective of mitigating their impact. Our overarching aim is to curtail negative rewards, thereby enhancing the fluidity of traffic flow within the intersection as time progresses.

While numerous traffic parameters such as throughput, average delay, and travel time offer diverse metrics, our deliberate choice centers on the metric of total waiting time. This strategic decision is rooted in the metric's capacity to furnish a comprehensive evaluation of TSC strategies in urban environments. Unlike singular parameters, total waiting time encapsulates the cumulative influence on the entire traffic system, presenting a holistic perspective on congestion levels. Computed by summing the waiting times of all vehicles in the environment at time step t, a vehicle's waiting time is incremented when its speed falls below 0.1 m/s, as defined by (8).

$$TotalWT_t = \sum wt_{i,t} \tag{8}$$

Where $wt_{i,t}$ present the waiting time of the vehicle i at timestep t. then the reward can be defined as (9).

$$reward_t = \ TotalWT_{t-1} - TotalWT_t \tag{9}$$

The waiting time of a vehicle is the time (in seconds) after the appearance of the vehicle in the environment where its speed is less than 0.1 m/s at the step of the agent t. As a metric, each agent considers the total cumulative waiting time for step t of each vehicle ($Cumul\_TotalWT_t$). If the vehicle manages to move forward but fails to pass the intersection, this metric does not reset the value of the total cumulative waiting time (10). This saves us from misleading statistics during agent training.

$$Cumul\_TotalWT_t = \sum cumul\_wt_{i,t} \tag{10}$$

Where $cumul\_wt_{i,t}$ is the accumulated waiting time of the vehicle i at timestep t. Then our reward function is the accumulated total waiting time of all the cars in the intersection captured respectively at agent step t and t – 1 presented as (11).

$$r_t = Cumul\_TotalWT_{t-1} - Cumul\_TotalWT_t \qquad (11)$$

In (9) and (11) produce negative values. As more vehicles accumulate in the queues at the agent's step t, the reward becomes increasingly negative, prompting the agent to assess its action more carefully. Consequently, in our results, we focus on visualizing the negative reward at each time step to evaluate the behavior of each agent.

3.6. Training algorithm

Algorithm 1 summarizes the complete training process of each agent. Using deep Q-learning, each agent first retrieves the environment state St = $(P_t, C_t^c, \Delta C_t^c, V_t^c)$ and provides it as inputs to the CNN network, which estimates the Q-values $Q(S_t, a, \theta)$ as outputs for all possible actions in the action space. As illustrated in Figure 7. Each agent records observed interaction experience $E_t = (S_t, A_t, R_t, S_{t+1})$ into a replay memory M $= \{E_1, E_2, \cdots, E_t\}$. As part of our adopted CTDE approach, agents engage in recognition-sharing, wherein each agent shares its last taken action at time t-1 with others, and vice versa. This collaborative strategy allows us to harness the advantages of centralized training while addressing challenges related to partial observability and minimizing dimensionality concerns associated with centralized execution.

Algorithm 1: DQN with experience replay and target network for ATSC
```
1:  Initialize CNN network with random weights $\theta_0$
2:  Initialize target network with weights $\theta'_0 = \theta_0$
3:  Initialize hyperparameters $\epsilon$, $\gamma$, $\alpha$, and the number of episodes N
4:  for each episode episode = 1,2,...,N do
5:      Initialize action $A_0$
6:      Start a new time step
7:      for each time step time = 1,2,...,T do
8:          if a new time step t begins then
9:              The agent observes the current intersection state $S_t$ ;
10:             The agent selects an action A = $\text{argmax}_a Q(S_t,a;\theta)$ with probability 1 - $\epsilon$,
                    and randomly selects a random action $A_t$ with probability $\epsilon$,
11:             if A == $A_{t-1}$ then
12                  No transition phase for traffic signals;
13:             else
14:                 Apply a transition phase for traffic signals;
15:             end if
16:         end if
17:         Execute the selected action $A_t$;
18:         Vehicles move according to the current traffic signals;
19:         Increment time time = time + 1;
20:         if the time step t ends then
21:             The agent observes the reward R and the current intersection state St+1;
22:             Store the observed experience $(S_t,A_t,R_t,S_{t+1})$ in replay memory M;
23:             Randomly sample a minibatch of 100 experiences $(S_i,A_i,R_i,S_{i+1})$ from M;
24:             Form the training data: input set X and targets y;
25:             Update $\theta$ with the training data;
26:             Update the target network $\theta'$ according to equation (12);
27:         end if
28:     end for
29: end for
```

The replay memory M has a finite capacity of 50,000 samples, and once full, the oldest data is discarded. To learn CNN parameters that best approximate $Q^*(\mathrm{s}, \mathrm{a})$, the agent requires training data: input data set X $= \{(S_t, A_t)$: t $\geq$ 1$\}$ and the corresponding targets y $= \{Q^*(S_t, A_t)$: t $\geq$ 1$\}$ can be retrieved from replay memory M for the input data set. However, the target $Q^*(S_t, A_t)$ is unknown. We use its estimated value $R_t + \gamma\, max_{a'}\, (Q(S_{t+1}, a';\, \theta')$ as the target instead, as in [16], [26]–[28], where $Q(S_{t+1}, a';\, \theta')$ is the output of a separate target network with parameters θ'. These parameters are softly updated using the equation (soft update for weights), and the target network's input is the corresponding S_{t+1} from interaction experience $E_t = (S_t, A_t, R_t, S_{t+1})$. The target network employs the same architecture as the CNN network depicted in Figure 7. Thus, targets y $= \{R_t + \gamma\, max_{a'}\, (Q(S_{t+1}, a';\, \theta')$: t $\geq$ 1$\}$.

We use the stochastic gradient descent algorithm root mean squared propagation (RMSProp) [29] with a minibatch size of 100 to reduce computational costs. During the training process of the CNN network, the agent utilizes the experience replay technique by randomly selecting 100 samples from the replay memory

M to create input data and target pairs. These pairs are then utilized to update the CNN parameters θ using the RMSProp algorithm. Along with updating the CNN parameters, the agent also needs to update the target network parameters′ using a method called soft updating as illustrated in (12) [27].

$$\theta' = \beta\theta + (1 - \beta)\theta' \tag{12}$$

Where β is update rate, β ≪ 1.

In [16], [26]-[28], we explain why experience replay and target network mechanisms can improve algorithm stability. To achieve optimal Q-values and learn the best action policy, the agent is trained on a limited number of intersection states, which may result in poorly estimated Q-values for inexperienced states or outdated estimates due to changing state spaces. This creates a trade-off for the agent, choosing between using previously learned but potentially inaccurate or out-of-date Q-values and selecting the highest Q-value action or exploring other options to improve Q-value estimation and action policy. To address this, the agent employs the ε-greedy approach, selecting either the action with the highest Q-value with probability 1-ε (exploitation) or a random action with probability ε (exploration). In other words, the current episode h has a probability of selecting an exploratory action or a probability of 1-ε for an exploitative action. The agent parameters are set as shown in Table 1.

$$\varepsilon_e = 1 - \frac{e}{Total\_episodes} \tag{13}$$

Table 1. Training parameters

Parameters	Values
Learning rate α	0.001
Β	0.001
Γ	0.75
Training epochs	800
Number of episodes	100
Number of steps in an episode	5400 (Each episode corresponds to a traffic of 1.5 hours)

4. RESULTS AND DISCUSSION

4.1. Traffic scenarios

Because a high level of realism is required, traffic generation is a critical component of the simulation. That is why the Weibull distribution with a shape of two was chosen. This approximation shows that the number of cars increases in the early stages until it reaches a peak. Following that, the number of incoming vehicles decreases. The source and destination of each car are determined at random using a seed that changes with each episode. Four different scenarios have been developed:

– High-traffic scenario: 4,000 cars are generated in this scenario. It represents the peak traffic volume. Even with good control, this scenario results in long queues. The goal is to see if it is feasible and how much it can be mitigated.
– Low-traffic scenario: 600 cars are generated in this scenario, which represents a typical flow with no peak time. However, a large number of cars can cause traffic jams if traffic control is inadequate.
– NS-traffic scenario: 2,000 cars produced, with 90% of traffic originating at the north and south junctions. The remaining 10% begin at the east and west intersections. This scenario depicts an unbalanced situation in which a traffic deviation may occur, forcing vehicles to drive on the east-west axis.
– EW-traffic scenario: 2,000 cars produced, with 90% of traffic originating at the east and west junctions. The motivation is similar to the one described in the preceding point.

4.2. Results

Our study distinguishes itself by addressing the more complex task of optimizing traffic light control in a network with multiple intersections. While earlier studies have explored fixed configurations of traffic lights and the use of ANN for traffic management, they have not explicitly addressed the influence of knowledge sharing between agents in a multi-intersection environment. We compare our results with two existing approaches in the literature. The first approach considers a fixed configuration of traffic lights, a method that fails to adapt to dynamic traffic conditions and thus often leads to suboptimal traffic flow. The second approach builds upon the work of [19], which models environmental agents using an ANN and employs a basic, straightforward representation of the environment's state. This state is discretized for each arm of an intersection, where each cell follows an irregular pattern based on the DTSE formula [14].

However, the work in [21] is limited to an environment based on an isolated intersection. This limitation prevents its extension to a more complex multi-agent environment with an extended discrete traffic state encoding (DTDE) design, where no knowledge sharing between the agents is feasible. As a result, these approaches fall short when applied to real-world scenarios involving multiple interconnected intersections, where coordinated decision-making is crucial for effective traffic management.

Our approach, guided by a CTDE framework, allows for knowledge sharing between agents, addressing scalability issues often associated with MARL approaches. Despite using only two intersections in this study, our CTDE design lays the groundwork for extending the model to larger networks, ensuring the relevance and generalization of our approach. Our study suggests that the use of CNNs and a detailed DTDE representation enhances traffic management efficiency without compromising scalability. Unlike previous approaches limited to isolated intersections, our CTDE framework facilitates effective coordination between multiple intersections, leading to significant improvements in traffic flow dynamics.

For example, research by Lauer and Riedmiller [21], which relies on ANN and a basic environmental state representation, is constrained by its application to isolated intersections. This lack of inter-agent communication hinders its scalability and efficiency in managing urban traffic. In contrast, our CTDE framework's ability to enable knowledge sharing among agents significantly improves traffic synchronization and overall network performance.

Our contribution demonstrates significant improvements in traffic flow while effectively minimizing the total waiting time for vehicles at intersections. This is evident as our agents were able to accumulate fewer negative rewards, reflecting remarkable robustness. Figure 10 illustrates that our approach outperforms the neural network method in terms of accumulated negative rewards.

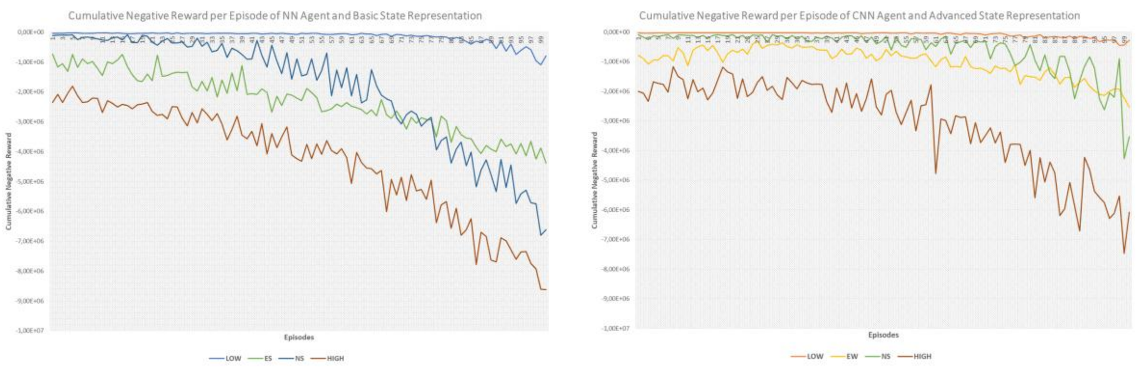

Figure 10. Cumulative negative reward over 100 episodes for both agents using CNN and neural network architectures, with basic and advanced state representations

In Table 2, we present a comparison of our results against the neural network approach and a basic representation of the environmental state. For the four traffic scenarios examined, the neural network-based agent only surpassed the classical traffic light management model in the east-west (E-W) scenario. In contrast, our approach, utilizing a CNN with an advanced and detailed representation of traffic intersection conditions, excelled across all traffic scenarios.

Figures 11 to 14 illustrate the performance of each agent at the two intersections, allowing us to visualize the impact of our contribution on agent behavior in various traffic situations. Figure 15 further complements this analysis by presenting the management of vehicle queue lengths at both intersections for both approaches across all proposed traffic scenarios. Notably, our method achieved optimal queue management, particularly in the HIGH scenario, leading to reductions in total waiting times of 7%, 70%, 58%, and 97% for the high, low, north-south (NS), and east-west (EW) traffic scenarios, respectively, compared to the fixed traffic light configuration. Conversely, the neural network approach, despite knowledge sharing between agents, showed only a marginal reduction in vehicle waiting times, primarily limited to the E-W scenario.

While our study focused on a limited scale with two intersections, future research should explore the scalability of our CTDE approach to larger networks. Additionally, further investigation is warranted to assess the adaptability of our model under varying environmental conditions and traffic dynamics. The current study, although promising, represents an initial step towards comprehensive urban traffic management.

Our findings highlight the potential for extending our CTDE framework to more complex urban networks. Future studies could explore adaptive learning mechanisms within the MARL framework to dynamically adjust traffic management strategies based on real-time traffic conditions and environmental changes. Moreover, investigating the integration of other advanced AI techniques, such as reinforcement learning algorithms with dynamic state representations, could further enhance the adaptability and efficiency of traffic management systems.

In conclusion, our study demonstrates the effectiveness of integrating CNNs and a detailed DTDE representation within a CTDE framework for optimizing traffic flow at multiple intersections. By significantly reducing vehicle waiting times across diverse traffic scenarios, our approach contributes to advancing intelligent traffic management systems and lays a foundation for scalable applications in urban environments. This work not only addresses current limitations in traffic management research but also opens avenues for future exploration and development in the field.

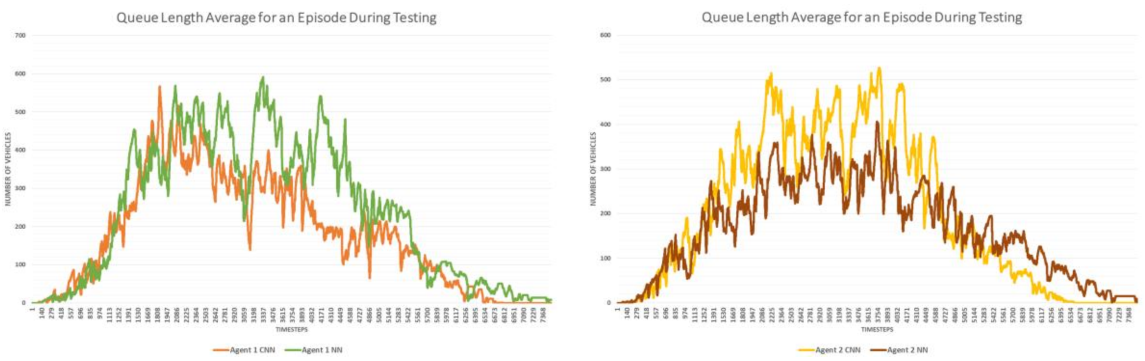

Figure 11. Agent 1 and Agent 2 queue management for HIGH traffic scenarios in both approaches

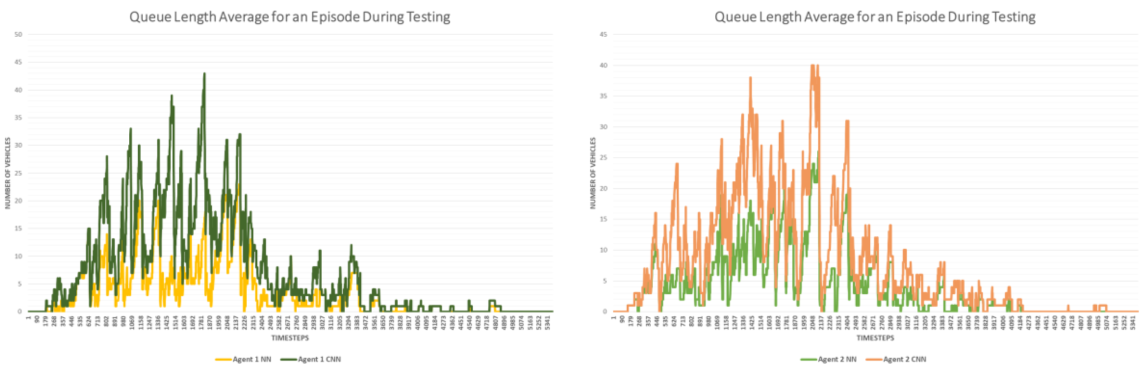

Figure 12. Agent 1 and Agent 2 queue management for LOW traffic scenarios in both approaches

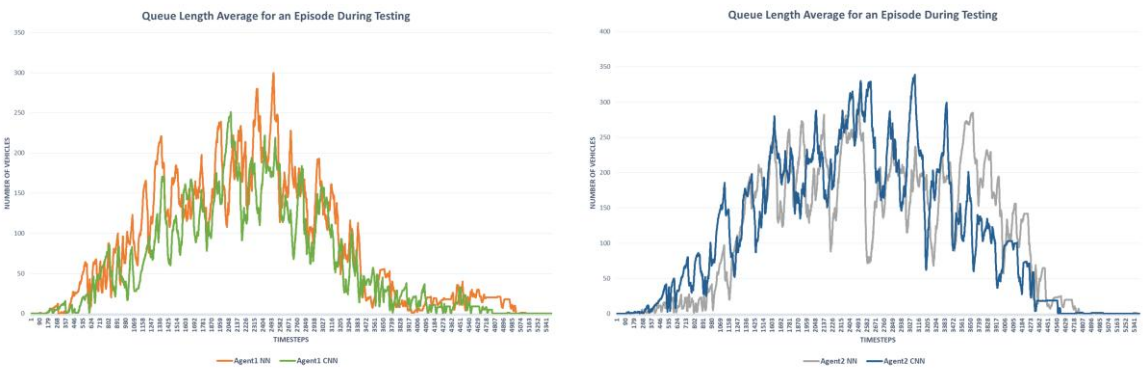

Figure 13. Agent 1 and Agent 2 queue management for EAST-WEST traffic scenarios in both approaches

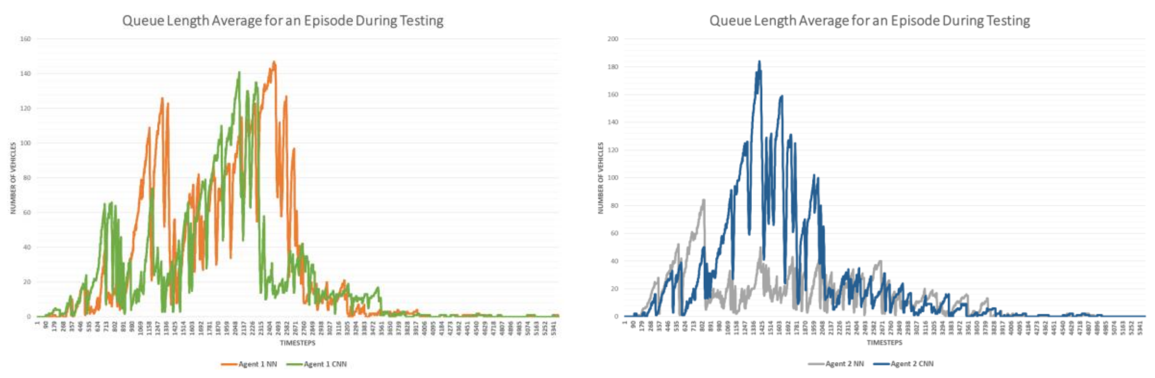

Figure 14. Agent 1 and Agent 2 queue management for NORTH-SOUTH traffic scenarios in both approaches

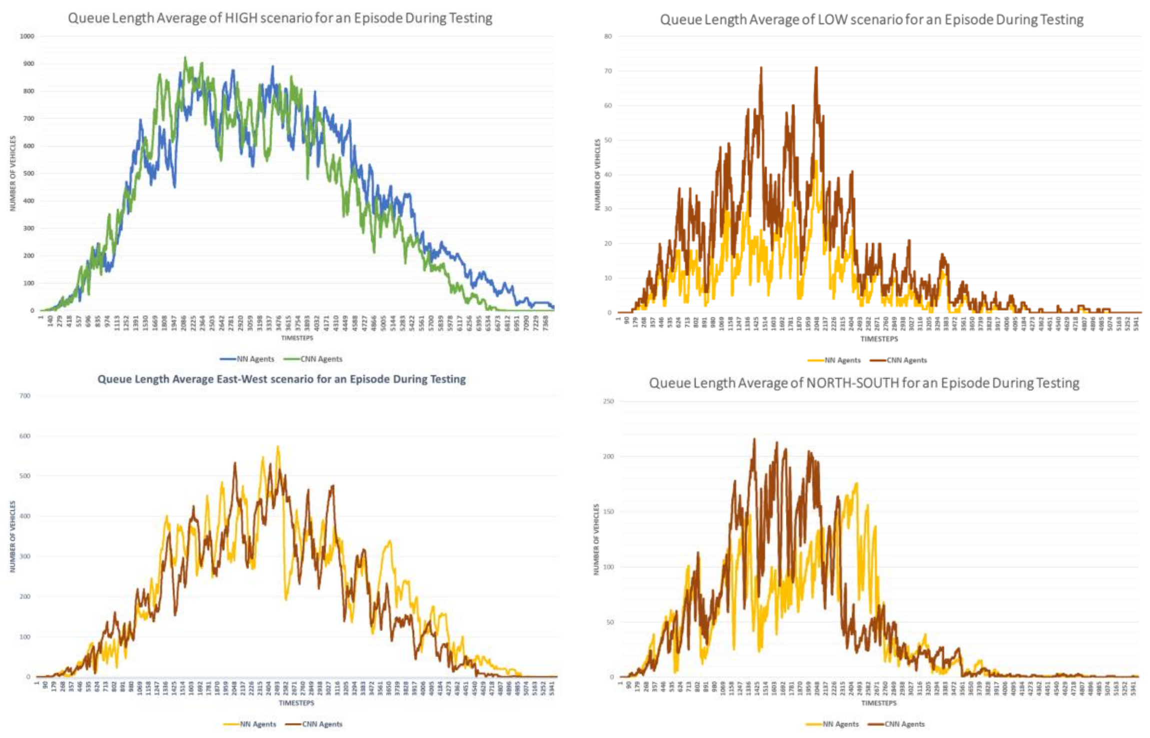

Figure 15. Vehicle queue length management at both intersections provided by both approaches across all proposed traffic scenarios

5. CONCLUSION

This paper explored the feasibility of utilizing reinforcement learning for adaptive traffic signal management. Using a validated traffic simulator, we established a robust framework for evaluating and training reinforcement learning agents in a controlled environment. Key investigations included detailed environmental state representation, knowledge sharing among agents in a partially observable setting, and the development of a refined reward mechanism focused solely on negative value accumulation to assess agent performance. Future research endeavors will build upon these foundations to enhance the outcomes achieved in this study. Specifically, we intend to further refine reinforcement learning algorithms and investigate their implementation across larger, more complex road networks. Additionally, we aim to explore the coordination potential among multiple reinforcement learning agents to achieve global traffic flow improvements rather than local optimizations alone. These analyses are pivotal in understanding both the potential advantages and unintended consequences associated with deploying self-adaptive systems in real-world settings. By continuing to refine our approaches and methodologies, we aim to contribute meaningfully to the ongoing evolution of intelligent transportation systems.

REFERENCES

[1] L. A. Prashanth and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 412–421, 2011, doi: 10.1109/TITS.2010.2091408.
[2] H. Wei, H. Yao, G. Zheng, and Z. Li, "IntelliLight: A reinforcement learning approach for intelligent traffic light control," *The ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2496–2505, 2018, doi: 10.1145/3219819.3220096.
[3] T. Wu *et al.*, "Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8243–8256, 2020, doi: 10.1109/TVT.2020.2997896.
[4] M. Wiering, "Multi-agent RL for traffic light control," *Physical Review D*, vol. 96, no. 1, 2017.
[5] K. J. Prabuchandran, A. N. Hemanth Kumar, and S. Bhatnagar, "Multi-agent reinforcement learning for traffic signal control," *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, pp. 2529–2534, 2014, doi: 10.1109/ITSC.2014.6958095.
[6] B. Abdulhai and L. Kattan, "Reinforcement learning: Introduction to theory and potential for transport applications," *Canadian Journal of Civil Engineering*, vol. 30, no. 6, pp. 981–991, 2003, doi: 10.1139/l03-014.
[7] D. D. Oliveira *et al.*, "Reinforcement learning-based control of traffic lights in non-stationary environments: A case study in a microscopic simulator," in *CEUR Workshop Proceedings*, 2006, vol. 223.
[8] A. L. C. Bazzan, "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 3, pp. 342–375, 2009, doi: 10.1007/s10458-008-9062-9.
[9] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv-Computer Science*, pp. 1-9, 2013.
[10] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100, 2016, doi: 10.1609/aaai.v30i1.10295.
[11] Z. Wang, T. Schaul, M. Hessel, H. V. Hasselt, M. Lanctot, and N. D. Frcitas, "Dueling network architectures for deep reinforcement learning," *33rd International Conference on Machine Learning, ICML 2016*, vol. 4, pp. 2939–2947, 2016.
[12] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 2016, pp. 1–21.
[13] A. S. Lakshminarayanan, S. Sharma, and B. Ravindran, "Dynamic action repetition for deep reinforcement learning," *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 2133–2139, 2017, doi: 10.1609/aaai.v31i1.10918.
[14] W. Genders and S. Razavi, "Using a deep reinforcement learning agent for traffic signal control," *arXiv-Computer Science*, pp. 1-9, 2016.
[15] W. Genders and S. Razavi, "Evaluating reinforcement learning state representations for adaptive traffic signal control," *Procedia Computer Science*, vol. 130, pp. 26–33, 2018, doi: 10.1016/j.procs.2018.04.008.
[16] N. Faqir, N. En-Nahnahi, and J. Boumhidi, "Deep Q-learning approach for congestion problem in smart cities," *4th International Conference on Intelligent Computing in Data Sciences, ICDS 2020*, 2020, doi: 10.1109/ICDS50568.2020.9268709.
[17] N. Faqir, C. Loqman, and J. Boumhidi, "Deep Q-learning approach based on CNN and XGBoost for traffic signal control," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 9, pp. 529–536, 2022, doi: 10.14569/IJACSA.2022.0130961.
[18] Y. Gong, M. Abdel-Aty, Q. Cai, and M. S. Rahman, "Decentralized network level adaptive signal control by multi-agent deep reinforcement learning," *Transportation Research Interdisciplinary Perspectives*, vol. 1, 2019, doi: 10.1016/j.trip.2019.100020.
[19] A. Vidali, L. Crociani, G. Vizzari, and S. Bandini, "A deep reinforcement learning approach to adaptive traffic lights management," in *CEUR Workshop Proceedings*, 2019, pp. 42–50.
[20] M. L. Littman, "Value-function reinforcement learning in Markov games," *Cognitive Systems Research*, vol. 2, no. 1, pp. 55–66, 2001, doi: 10.1016/S1389-0417(01)00015-8.
[21] M. Lauer and M. Riedmiller, "An algorithm for distributed reinforcement learning in cooperative multi-agent systems (2000)," in *Seventeenth International Conference on Machine Learning*, 2000, pp. 535–542.
[22] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proceedings of the 11th International Conference on Machine Learning, ICML 1994*, 1994, pp. 157–163, doi: 10.1016/B978-1-55860-335-6.50027-1.
[23] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv-Computer Science*, pp. 1-14, 2016.
[24] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: an overview," *Studies in Computational Intelligence*, vol. 310, pp. 183–221, 2010, doi: 10.1007/978-3-642-14435-6_7.
[25] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO--simulation of urban mobility: an overview," *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, pp. 1-6, 2011.
[26] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.
[27] J. Gao, Y. Shen, J. Liu, M. Ito, and N. Shiratori, "Adaptive traffic signal control: deep reinforcement learning algorithm with experience replay and target network," *arXiv-Computer Science*, pp. 1-10, 2017.
[28] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: an overview," *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, vol. 16, pp. 426–440, 2018, doi: 10.1007/978-3-319-56991-8_32.
[29] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

BIOGRAPHIES OF AUTHORS

Nada Faqir is a Professor in the BTS Specialized Technician Certificate program at Ismail High School. She specializes in the field of information systems development. She holds a Ph.D. in Urban Mobility, conferred in 2023, and a Master's degree in Intelligent Systems and Decisional, obtained in 2019, from the Faculty of Sciences Dhar el Mehraz in Morocco. She can be contacted at email: nada.faqir@usmba.ac.ma.

Jaouad Boumhidi is a Professor of Computer Science at the Faculty of Sciences, Fez Morocco. Board of Governors member of the International Neural Network Society. He received the Ph.D. in Computer Science from The University of Sidi Mohamed ben Abdellah in 2005. His research interests are in computational intelligence and intelligent transportation systems. He can be contacted at email: jaouad.boumhidi@usmba.ac.ma.

Chakir Loqman is a professor of computer science at Sidi Mohamed Ben Abdellah University in Morocco. With a focus on cutting-edge research, his expertise lies in the realms of text mining, artificial intelligence, and optimization. His dedication to advancing knowledge in these areas underscores his commitment to shaping the future of computer science education and research at the university. He can be contacted at email: chakir.loqman@usmba.ac.ma.

Youness Oubenaalla is co-founder of the International Neural Network Society Morocco, co-founder of the IEEE Computational Intelligence Morocco Chapter. His current research includes deep reinforcement learning in large distributed datasets, neural networks for medical image classification, and applications related to clustering in vehicular ad-hoc network using artificial neural network. He can be contacted at email: y.oubenaalla@umi.ac.ma.