Digital-Twin-Based Deep Reinforcement Learning Approach for Adaptive Traffic Signal Control

Hani Kamal*, Wendy Yánez†, Sara Hassan‡ and Dalia Sobhy §

***Abstract*—Urban vehicle emissions are one of the main contributors to air pollution since most vehicles still rely on fossil fuels, despite the growing popularity of alternative options such as hybrids and electric cars. Recently, Artificial Intelligence (AI) and automation-based controllers have gained attention for their potential use in adaptive traffic signal control. Studies have been conducted on applying Deep Reinforcement Learning (DRL) models to reduce travel time in adaptive traffic signal control. However, little research has been done on adapting traffic signal control to reduce CO2 emissions and fuel consumption of urban vehicles. As such, this paper proposes a *digital-twin-based adaptive traffic signal control approach*. This approach comprises five phases, from traffic data collection to control signal actuation. It uses the DRL Multi-Agent Deep Deterministic Policy Gradient (MADDPG) to optimise for reduced fuel consumption and CO2 emission. To assess the effectiveness and applicability of the proposed approach, a quantitative simulation is performed using synthetic and real-world traffic datasets from a multi-intersection network in a neighbourhood in Amman, Jordan, during peak hours. The findings suggest that the DRL approach based on digital twins on synthetic networks can reduce CO2 emissions and fuel consumption even when using a basic reward function based on stopped vehicles.**

***Index Terms*—Digital twin, deep reinforcement learning, agent-based simulation, traffic management.**

I. Introduction

URBAN air pollution has become a significant health problem in many large cities around the world as the number of motor vehicles increases [1]. Nearly 56 billion pounds of hazardous CO2 emissions in 2011 were due to traffic congestion [2]. Although traffic junctions significantly lower mobile air pollution, repeated vehicle speed changes and stop-and-go congestion increase fuel consumption and CO2 emissions [3]. Traffic lights are also used to manage traffic in conjunctions at peak hours, but their operation relies on human expertise, which could be prone to errors, delays, and inefficiencies.

Adaptive traffic signal control approaches based on AI and automation have been developed to adapt traffic light schedules to reduce travel time [4], [5]. However, most of these approaches cannot optimise for reduced CO2 emissions and fuel consumption in urban vehicles. . Moreover, with the development of technologies such as the Internet of Things (IoT), Machine Learning (ML), and Deep Reinforcement Learning (DRL), DT contributes to the advancement of urban traffic management, highways, intelligent vehicle infrastructure, and autonomous driving [6].

This paper therefore addresses the following research questions:

- **RQ1:** How can integrating DT enhance traffic signal control's efficiency and decision-making process?
- **RQ2:** How can DRL algorithms enrich the DT infrastructure and optimise for reduced urban vehicle CO2 emissions and fuel consumption?

To address the above RQs, this paper proposes *a digital-twin-based approach for adaptive and efficient traffic signal control*. The approach comprises five phases, starting from traffic data collection to traffic control signal actuation to improve decision-making in traffic flow and reduce congestion (addressing RQ1). The approach uses MADDPG to optimise for reduced CO2 emissions and fuel consumption in urban vehicles (addressing RQ2).

The remainder of the paper is structured as follows. Section II briefly summarises the relevant literature, and then motivates this paper's contribution. Section III describes the proposed approach, including the different phases and how they are used. Section IV presents the experimental evaluation of our approach using a real case study. The paper concludes and provides future work in Section V.

II. Related Work

This section compares and contrasts the relevant existing literature with the target problem and the contribution of this article. The literature is divided into: *deep reinforcement learning* (Section II-A) and *traffic management* (Section II-B).

A. Deep Reinforcement Learning

DRL has also been successfully applied in [5] to adopt traffic light schedules to reduce the total system travel time . However, the associated impact of DRL in adaptive traffic signal control for urban vehicles on air quality remains unexplored. The work in [4] proposes a novel multi-agent recurrent deep-deterministic policy gradient algorithm (MARDDPG) based on the deep-deterministic policy gradient algorithm (DDPG) for traffic light control (TLC) in vehicular networks.

*Birmingham City University, Birmingham, UK, Email: h.kamal95@hotmail.com
†School of Computer Science, University of Birmingham, Birmingham, UK, Email: w.yanez@bham.ac.uk
‡Birmingham City University, Birmingham, UK, Email: Sara.Hassan@bcu.ac.uk
§Arab Academy of Science and Technology and Maritime Transport, Alexandria, Egypt, Email: dalia.sobhi@aast.edu

The results indicate that these algorithms can be combined in multiple scenarios and coordinate multiple intersections, significantly reducing vehicle and pedestrian congestion. Similarly to the work above , this paper can be complemented by the dimensions targeted in our work, especially since it is based on the same deterministic approach.

B. Traffic management

The work in [7] highlights the need to use predicted traffic conditions to generate appropriate intersection traffic control plans. The proposed work addresses this gap, focusing on traffic planning that targets reduced CO2 emissions and fuel consumption. Another research gap in [7] corresponds to the need to validate a particular traffic control plan in a realistic setting. In the proposed work, we overcome this challenge by using a DT to enhance traffic signal control based on real simulation.

Moreover, in [8], deep learning in traffic signal control has been presented without considering the reduction in CO2 emissions and fuel consumption. In another work [9], a systematic review on smart city traffic control was published.

More generically, the work in [9] reviews control systems for smart cities. Smart traffic control is one trending dimension of smart cities presented in this work. Optimising for environmental sustainability in traffic signal control is a critical aspect of smart cities. Therefore, our work can contribute significantly to smart cities.

In [10], two approaches for environmentally friendly traffic signal control, "eco-routing, corresponding to planning the routes for each specific vehicle, given its origin and destination, and eco-driving, consisting of calculating vehicle trajectories along a given route, considering technical limitations of vehicle and environmental constraints such as traffic lights". The "routing" approach was also presented in [11] using the cluster-based hybrid routing system to avoid traffic congestion. The use of digital twins proposed in this paper can complement these approaches, making them more proactive in covering a variety of traffic scenarios.

Regarding AI-based contributions to traffic signal control, [12] proposed a traffic intersection management algorithm considering the "nonlinear vehicle dynamic model and weather conditions". However, the proposed algorithm does not benefit from using digital twins, leaving space for incorporating what-if analysis to stress test the intersection system under multiple scenarios. In [13] [14], a cooperative centralised intersection control approach was developed that reduces travel time, CO2 emissions, and fuel consumption. In [15], smart traffic signal control was used as a case study to present Graphical Neural Networks as a solution to avoid smart traffic congestion. These approaches can benefit from the what-if analysis provided by digital twins.

On the digital twin front, the work in [16] presents a digital twin that analyses high streams of data points from a racing car vehicle to optimise its racing strategy. "Before a car even reaches a track, AI and simulated environments are combined through digital twins with real-world testing to enable data-driven engineering changes every 20 minutes on average". In essence, our work takes the same approach to achieve a different goal: optimising for reduced CO2 emissions and fuel consumption for urban vehicles.

Table I summarises the related work (presented in Section II). It shows that the proposed approach is the *first* to target CO2 emissions and fuel consumption using a digital twin-based solution.

III. Proposed Digital Twin Traffic Control Framework

This section describes the phases and components of the proposed approach, together with the inputs and outputs of each phase, as shown in Figure 1.

A. Data Collection

The components of the physical twin are used to collect data describing the monitored physical system (Phase 1: data collection) or to actuate in the physical system (Phase 5: actuation phase) as illustrated in Figure 1. In the data collection phase, IoT sensors collect the required metrics periodically or continuously from traffic junctions and feed them into the middleware layer of the digital twin. It is worth noting that the exact location and the number of sensors used depend on the particular traffic junction being monitored and the frequency with which data needs to be collected. For example, more frequent data collection might be necessary for a particularly congested traffic junction or peak traffic hours. In those cases, more sensors and/or more frequent data collection exercises might be required to determine an accurate representation of traffic junctions. Deciding on the optimal data collection rate and the exact number and location of the sensors are research questions outside the scope of this paper. However, the proposed approach is flexible enough to be applied regardless of the underlying number of sensors, the sensors' location, and the data collection frequency. The only "restriction" of the proposed approach is to ensure that the sensors count the number of traffic light cycles (i.e., how many times the traffic light switches from green to red and vice versa), the CO2 emissions at the junction, and the average fuel consumption at the junction. Traffic light cycle counters are necessary to determine the optimal traffic signal control strategy in the data collection phase. CO2 emissions and fuel consumption values must have a baseline that can be tracked to check whether improvements occur after a control strategy is actuated.

B. Modelling

This phase takes place in the DT and is intended to model, analyse, simulate, and perform decision-making exercises in a virtual representation of traffic junctions using data from the data collection phase. Thus, the modelling phase acts as a middleware layer between the physical and DT. In particular, *Traffic Control Interface (TraCI)* module and the *Simulation of Urban Mobility (SUMO)*[1] are used to build a model given

[1]For more information about the Simulation of Urban Mobility (SUMO), visit: https://sumo.dlr.de/docs/index.html. For details about the Traffic Control Interface (TraCI), refer to: https://pypi.org/project/traci/.

TABLE I: Summary of existing literature.

Paper	Uses Digital Twins	Optimises for less CO2 emissions	Optimises for less fuel consumption
[5]	No	No	No
[4] ,	No	No	No
[10]	No	Yes	Yes
[11]	No	Yes	No
[12]	No	Yes	Yes
[13], [14]	No	Yes	Yes
[15]	No	Yes	Yes
[16]	Yes	No	No

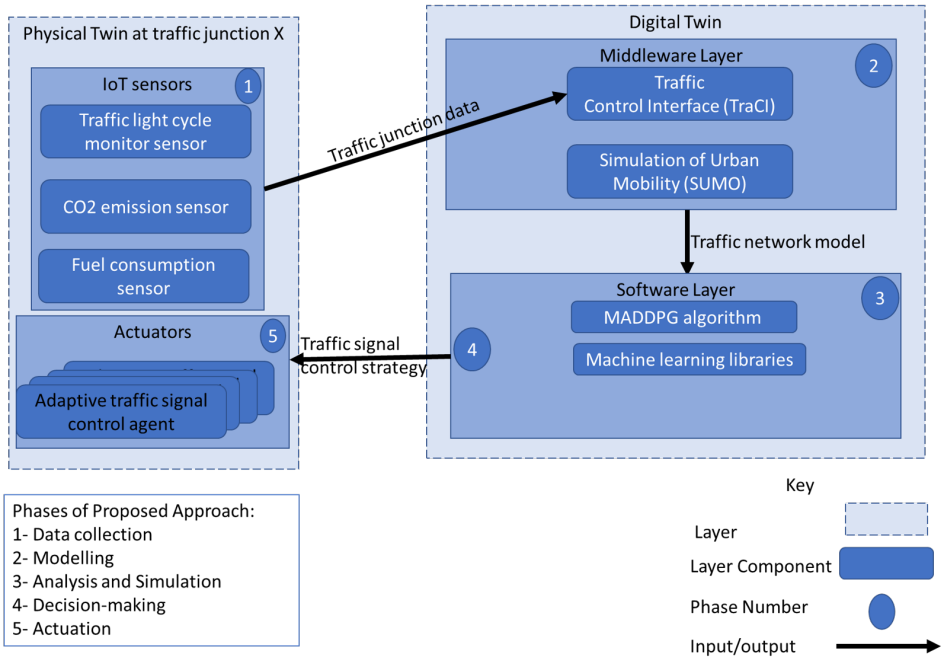

Fig. 1: Proposed Digital Twin-Based Approach.

the traffic junction data. In SUMO, the traffic network model consists of a set of roads, intersections, lanes, vehicles, and pedestrians moving through the network. The model can be configured to reflect real-world traffic conditions or can be used to explore hypothetical scenarios to test and evaluate different transportation strategies. The TraCI module is a Python module provided as part of the SUMO package. It allows users to interface with SUMO simulations using the TraCI protocol, a communication protocol for transportation simulation programmes. The TraCI module provides a set of functions that can be called from a Python script to control and retrieve information from a SUMO simulation—for example, the traci.vehicle.setSpeed() function can be used to set the speed of a vehicle in the simulation and the traci.vehicle.getPosition() function can be used to retrieve the position of a vehicle.

The output of the middleware layer (and hence the modelling phase) is a traffic network model representing the physical traffic junction being monitored by the physical layer.

C. Analysis and Simulation

In this phase, DRL is applied to stress-test multiple what-if scenarios given the traffic network model. The rationale is to eventually decide which strategy is the "best" given this model. The analysis and simulation exercise is located in the software layer of the proposed approach because it uses software to perform its role. In particular, it uses the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm and some machine learning libraries. The MADDPG algorithm is used to learn the optimal traffic control strategies based on the traffic network model. In contrast, machine learning libraries, such as PyTorch, are used to implement the MADDPG algorithm.

1) Multi-Agent Deep Deterministic Policy Gradient (MADDPG) Description: In a single-agent system like the (DDPG), the agent takes actions within an environment to optimise its behaviour through rewards. This process is called a Markov Decision Process. It can be represented as a quintuple (S, A, P, r, γ), where S is the state space, A is the action space, P is the probability of state transition, r is the immediate reward and γ is the discount factor. The agent uses rewards to guide decision-making and navigate the environment [17].

In a multi-agent setting, the agents interact with their environment and one another. The reward for an individual agent is affected not only by its actions but also by the actions of other agents. To model multi-agent systems, Markov games are often used, which can be represented by (n, S, A_1, A_2, ..., A_n, r_1, r_2, ..., r_n, γ) where S denotes the joint state of all agents (i.e., the combined state of multiple agents), n is the number of agents, A_i is the action space of agent i, and r_i is the immediate reward for agent i.

A multi-agent actor-critic method called the MADDPG algorithm first appeared in [18], [19]. This algorithm's structure, essentially an extension of the DDPG algorithm, is shown in Figure 2. It uses a critical network to train agents to anticipate other agents' actions using deep neural networks. It does this by using the ongoing information the environment has provided over time. In the MADDPG algorithm, each agent has its actor-network that produces actions based on its observation of the state. Each agent also has a corresponding critic network trained using data from all actors simultaneously. This allows the algorithm to evaluate all agents' joint strategies (π_1,π_2,...,π_n). The policy gradient of the joint strategy is obtained using the DDPG algorithm.

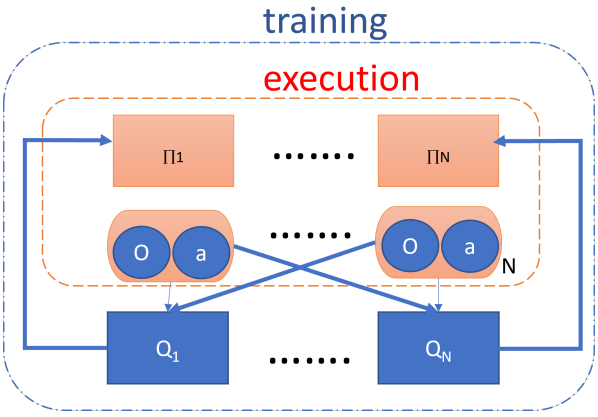

Fig. 2: MADDPG Framework.

2) State space: The real-time traffic situation at an intersection can be represented using all vehicles' positions., which can be achieved using a matrix representation method, as illustrated in Figure 3. In this approach, the road that extends from the parking line is divided into several fixed-length cells, and an element in the matrix represents each vehicle's location. Each element corresponds to the number of vehicles in a specific cell, and the safe distance between the vehicles establishes the cell size. This matrix representation captures the current traffic flow and state in the next moment. This method is useful because (i) it can reduce the dimensionality of the data, (ii) it can eliminate unnecessary information, and (iii) it can identify critical features of the traffic network. It enables the agent to make effective decisions and accelerates the training process.

3) Action Space: In DRL for adaptive traffic control, the action space is the set of actions the traffic control system can take to influence traffic flow. These actions include adjusting

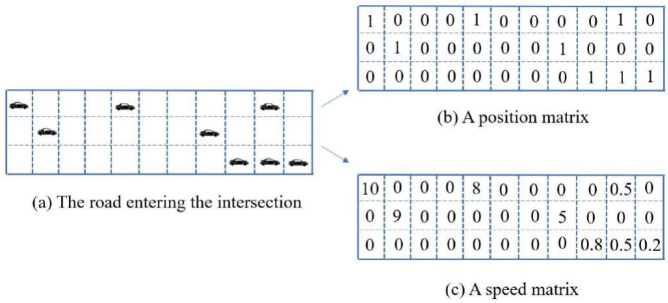

Fig. 3: State space for the MADDPG algorithm

the timing of traffic signals, changing the speed limits on certain roads, or rerouting traffic to alternative routes.

The action space is an essential concept in DRL because it defines the range of options the traffic control system has to respond to changing traffic conditions. The goal of the traffic control system is to select actions that will optimise traffic flow and minimise congestion. The action space defines the set of actions that the system can consider when making this decision.

The size and complexity of the action space can vary depending on the specific needs of the traffic control system. In some cases, the action space may be relatively simple, with a limited number of discrete actions available to the system. In other cases, the action space may be more complex, with many possible actions available and the need to consider continuous variables such as traffic signal timing or speed limits.

The proposed approach implements multi-agent traffic using 1*2 intersections with two discrete action spaces.

Figure 4 shows the four action spaces at each intersection. Each state shows the three possible actions for an incoming vehicle from one lane. There are two agents for the ten-intersection state dimension with two action spaces and two phases per intersection.

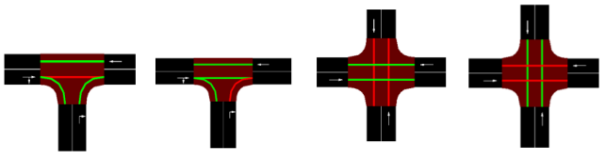

Fig. 4: Action space for the MADDPG algorithm

4) Reward Function: In adaptive traffic control field, reinforced machine learning algorithms have been used to develop systems that can learn and adapt to changing traffic conditions to optimise traffic flow and minimise congestion. One key aspect of these algorithms is the reward function, which is used to assess the effectiveness of the actions taken by the traffic control system and guide the system towards actions that are more likely to produce positive results [20].

The design of the reward function is crucial to the success of the DRL algorithm, as it determines how the system will evaluate and compare different actions. In general, the reward function should be carefully tailored to reflect the goals and priorities of the traffic control system, considering metrics such as congestion, safety, and environmental impact.

The proposed approach calculates the reward based on the change in the total number of halting vehicles for the last time step after the action is executed, as shown in Equation 1.

$$R_t = k(W_t – W_{t+1}) \quad (1)$$

The mean reward per step (R_t) is equal to the difference between the vehicles that halt at the red signal in the current step (W_t) and the one that halts vehicles in the next state (W_{t+1}).

D. Decision-making

The output of applying the above algorithm is an action space that maximises the reward function. In other words, the output of this phase is the result of stress testing the model in different scenarios to eventually decide the best traffic control strategy from the perspective of the reward function. Therefore, the decision-making phase is whether the optimal action space is communicated to the actuators (i.e., the physical traffic control signals).

E. Actuation

In the actuation phase, the traffic signal controls switch the lights at the junction, depending on the control strategy proposed in the decision-making phase. In particular, the specific proposed action from the action space takes effect in the actuation phase. In Figure 1, the term "agent" is used to describe the traffic control signals that activate the proposed control strategy. It is worth noting that multiple agents are usually required to act on a strategy. In simple terms, multiple traffic lights must switch simultaneously from green to red or vice versa to fulfil the strategy proposed in the decision-making phase. Once the strategy is activated, another cycle of all phases described previously will recheck whether the actuated strategy improved CO2 emissions and fuel consumption compared to previous cycles.

The concrete definition of the state representation, the action set, the reward function, and the agent learning techniques involved for each specific junction are different and depend on the users of our approach.

IV. Evaluation

This section describes the application of our approach in a quantitative simulation setup and then uses that setup to evaluate the proposed approach. Quantitative evaluation metrics used to assess MADDPG results are travel time, fuel consumption, and CO2 emissions. These metrics highlight the effectiveness of our approach's phases in deciding an optimal traffic control signal strategy (i.e., addressing RQ1) and the suitability of MADDPG to the adaptive traffic signal control problem (i.e., addressing RQ2).

A. Quantitative Simulation Setup

Simulation of Urban Mobility (SUMO) software generates realistic traffic scenarios for testing. SUMO creates synthetic traffic situations and mirrors real-world conditions, including traffic volume, vehicle types, and driver behaviour. These synthetic scenarios are the foundation for evaluating the proposed traffic control strategies.

The next step is establishing a well-structured experimental environment to assess the traffic control system rigorously. This environment encompasses the physical infrastructure of the traffic junction, the deployment of sensors for real-time data collection, the seamless integration of SUMO-generated data with the physical setup, and a control interface for the reinforcement learning model to communicate optimal control strategies to traffic signals. Within the above setup, the evaluation is conducted as described in the following subsections.

B. Approach Application

1) Data Collection: Data Collection: Gathering essential traffic data is a critical requirement for both training the machine learning model and evaluating the effectiveness of adaptive traffic control system. The dataset used in this project originates from real-world sources, acquired through thorough manual counting methodologies.

2) Modelling: As previously illustrated, the simulation was carried out within a traffic network structured as a 1x2 grid. The placement of the intersection adhered to a separation of 200 metres. A consistent traffic volume was introduced at the start of the simulation. Noteworthy aspects of the simulation include the yellow-light phase exclusion and the absence of buffer intervals during traffic light phase transitions. This simulation represented two intersections, each embodied by its respective agent. Agents underwent iterative training known as "episodes," each comprising 3600 time steps, equivalent to a 60-minute simulation period.

The validation of the proposed traffic control algorithm transpired through simulation tests on the SUMO platform. SUMO, a microscopic traffic microsimulator recognised for its ability to model and analyse urban traffic dynamics, was instrumental in evaluating the algorithm's efficacy. Table II enumerates the relevant parameters of the traffic environment. In particular, the fuel type is denoted as PC — representing average passenger cars across fuel variants. Table III details hyperparameters pertinent to the reinforcement learning network.

TABLE II: Environment Parameters.

Parameter	Values
Distance between intersections	200 m
Vehicle size	4 m
Speed limit	11 m/s
Maximum distance between vehicles	2.5 m
Total vehicles through the network	2201
Vehicle Fuel Type	HBEFA3/PC

As mentioned in Table II, a fixed 2.5-meter inter-vehicle distance was maintained to optimise outcomes. Furthermore, the impact of the speed limit on the final results is discernible; excessively high-speed limits might compromise the results.

Table III enumerates the parameters employed in implementing the MADDPG algorithm. To effectively manage the considerable influx of data from the environment and the

TABLE III: Hyperparameters for MADDPG.

Parameter	Values
Replay memory size	5000
Batch size	64
Discount factor (Gamma)	0.99
Initial Epsilon Value	0.9
Learning rate	1e-3
Reward rate	0.1

concurrent operation of multiple neural networks, a batch size of 64 was maintained, and an expansive buffer memory was used.

3) Analysis and Simulation: After the agent's training phase, a series of assessments compared actual traffic lights with those under MADDPG control. Before delving into the analysis, it is imperative to acknowledge that real-world traffic signals within the simulation emulate fixed traffic lights, implying that their transitions adhere to predetermined time intervals based on historical real-world switch timings. Unlike their RL-controlled counterparts, these simulated signals lack the adaptability to respond dynamically to the simulated traffic flow. While efforts were made to align the simulated traffic conditions with actual scenarios, the inherent non-reactive behaviour of the real-world signals placed them disadvantaged when pitted against the trained traffic signals.

4) Decision-Making: Effective decision-making is paramount in devising optimal traffic signal strategies in the proposed adaptive traffic control system. This process is heavily based on the MADDPG algorithm. Post-training, our RL-controlled traffic signals quickly render real-time decisions guided by observed traffic conditions.

The decision-making process encompasses:

- **Observation**: DRL agents, representing traffic signals, continuously monitor the traffic network, including factors such as traffic flow, vehicle positions, and time.
- **Action Selection**: DRL agents choose actions based on observed conditions, employing deep neural networks to approximate the optimal action, factoring in local and neighbouring interactions.
- **Policy Learning**: DRL agents continually improve their policies through reinforcement learning, striving to maximise rewards by reducing travel time, fuel consumption, and CO2 emissions.
- **Communication**: In multi-agent settings, communication among DRL agents is pivotal for optimising traffic flow and mitigating congestion.

5) Actuation: refers to the execution of selected phases of traffic signals based on RL-controlled decisions. This section shows practical implementation.

- **Signal Control**: RL decisions are relayed to physical traffic signal controllers, which execute signal changes, responding to prevailing traffic conditions.
- **Real-Time Adaptation**: RL-controlled signals adapt in real-time to changing conditions, reacting to sudden traffic surges or congestion.
- **Monitoring and Feedback**: Continuous monitoring ensures RL decisions produce the desired results, prompting adjustments if suboptimal performance is detected.

C. Results and Discussion

This section presents the evaluation results of the above setup. The results reveal the effectiveness of our approach's phases in deciding an optimal traffic control signal strategy (i.e., addressing RQ1) and the suitability of MADDPG to the adaptive traffic signal control problem (i.e., addressing RQ2).

1) Average Travelling Time: Through 100 training episodes, the MADDPG algorithm successfully reduced the average travelling time by approximately 42%. In particular, this reduction stabilised after around 70 episodes, indicating the convergence of the learning process. However, some episodes showed deviations in travel time, which can be attributed to increased road congestion, making it challenging for the reinforcement learning algorithm to maintain stability.

2) Metrics Comparison: Following training, the model demonstrated an impressive accuracy rate of 99% in decision-making based on the information provided. This translated into substantial real-world improvement, where all vehicles completed their trips, unlike 43 vehicles left stranded without adaptive control.

3) CO2 Emissions and Fuel Consumption: Figure 5 and 6 compare CO2 emissions and fuel consumption during a one-hour rush hour period for the real-world traffic system and the MADDPG model. The prototype managed to reduce CO2 emissions by 11.78% and fuel consumption by 4.57% throughout its duration, indicating its positive impact on environmental sustainability.

4) Traffic Flow Comparison: When subjected to a realistic traffic flow scenario, MADDPG outperformed the real-world system by clearing the traffic slightly faster, taking approximately 4600s compared to 4800s.

TABLE IV: Traffic Flow Comparison.

Parameter	Real World	MADDPG
loaded	2201	2201
Time to clear the traffic	$\approx$ 4800	$\approx$ 4600

Our results underscore the potential of MADDPG in adaptive traffic signal control. The achieved reduction in the average travel time and the improvements in CO2 emissions and fuel consumption align with the address of RQ2. The model's to real-world scenarios aligns with addressing RQ1.

D. Threats to Validity

1) Internal threats: In considering validity threats to this paper's approach, three key internal concerns arise. Firstly, unaccounted confounding variables could influence observed results, challenging causal relationships. Secondly, accurate tuning of model parameters and hyperparameters is critical for effective and generalisable outcomes. Lastly, potential sampling bias in scenario selection may limit real-world applicability. Addressing these concerns rigorously will strengthen the validity and robustness of this paper's approach.

2) Training Variability: The MADDPG training process can exhibit variability due to hyperparameter settings, random initialisation, or environmental noise. Multiple training runs were conducted to mitigate this threat and analysed the consistency of results.

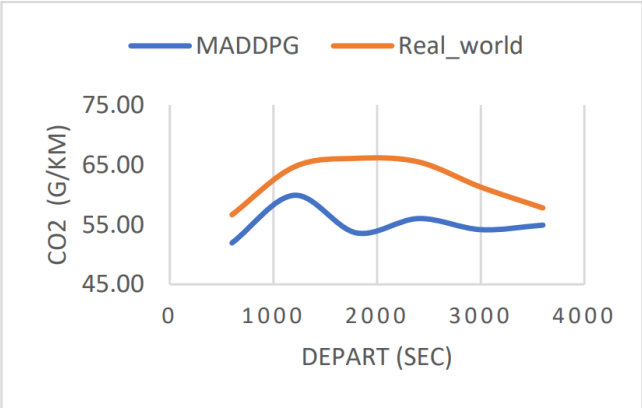

Fig. 5: Average CO2 emission rate for 1 hour traffic flow.

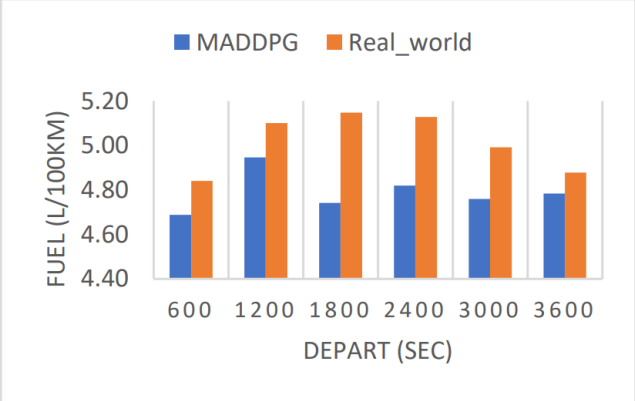

Fig. 6: Average Fuel Consumption for 1 hour traffic flow.

3) Data Collection Errors: Errors in data collection and pre-processing could introduce inaccuracies in the collected metrics. Best efforts were made to mitigate this threat by carefully validating and cleaning the data before analysis.

4) External threats: External threats to validity concern how our findings can be generalised to broader contexts or populations. The effectiveness of our approach is partly dependent on the choice of the reward function, the parameters, and the action spaces. However, our approach is flexible and systematic enough to be used regardless of the concrete choice of these dimensions.

5) Simulation Environment Realism: The realism of our simulation environment may differ from that of real-world traffic scenarios. The transferability of the results to real-world implementations may vary. Sensitivity analyses and real-world validation are necessary to assess this threat.

6) Model Generalisation: This paper focused on a specific configuration of the traffic network. The generalisability of MADDPG to other urban environments and traffic conditions remains a topic for further investigation. Future research should explore its applicability in various contexts.

In conclusion, while this paper demonstrates promising results in adaptive traffic signal control using digital-twin-based approaches and MADDPG, careful consideration of internal and external threats to validity is essential to ensure the reliability and applicability of our findings across varying transportation systems.

V. Conclusion

This paper proposes a digital-twin-based adaptive traffic signal control approach that uses MADDPG to optimise for reduced fuel consumption and CO2 emissions. The proposed approach is evaluated using quantitative simulation, which uses synthetic and real-world traffic datasets from a multi-intersection network in a neighbourhood in Amman, Jordan, during peak hours. The findings suggest that using the digital twin-based DRL approach in synthetic networks can reduce CO2 emissions and fuel consumption even when using a basic reward function based on stopped vehicles.

References

[1] A. J. McMichael, "The urban environment and health in a world of increasing globalization: issues for developing countries," *Bulletin of the world Health Organization*, vol. 78, pp. 1117–1126, 2000.

[2] D. Schrank, B. Eisele, T. Lomax, J. Bak *et al.*, "2015 urban mobility scorecard," 2015.

[3] E. A. Riley, L. Schaal, M. Sasakura, R. Crampton, T. R. Gould, K. Hartin, L. Sheppard, T. Larson, C. D. Simpson, and M. G. Yost, "Correlations between short-term mobile monitoring and long-term passive sampler measurements of traffic-related air pollution," *Atmospheric Environment*, vol. 132, pp. 229–239, 2016.

[4] T. Wu, P. Zhou, K. Liu, Y. Yuan, X. Wang, H. Huang, and D. O. Wu, "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.

[5] A. Haydari, M. Zhang, C.-N. Chuah, and D. Ghosal, "Impact of deep rl-based traffic signal control on air quality," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–6.

[6] L. Bao, Q. Wang, and Y. Jiang, "Review of digital twin for intelligent transportation system," in *2021 International Conference on Information Control, Electrical Engineering and Rail Transit (ICEERT)*. IEEE, 2021, pp. 309–315.

[7] E. Namazi, J. Li, and C. Lu, "Intelligent intersection management systems considering autonomous vehicles: A systematic literature review," *IEEE Access*, vol. 7, pp. 91 946–91 965, 2019.

[8] H. Nguyen, "Deep learning methods in transportation domain: a review," *IET Intelligent Transport Systems*, vol. 12, pp. 998–1004(6), November 2018. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-its.2018.0064

[9] Q.-S. Jia, H. Panetto, M. Macchi, S. Siri, G. Weichhart, and Z. Xu, "Control for smart systems: Challenges and trends in smart cities," *Annual Reviews in Control*, vol. 53, pp. 358–369, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578822000256

[10] B. Othman, G. De Nunzio, D. Di Domenico, and C. C. de Wit, "Ecological traffic management: A review of the modeling and control strategies for improving environmental sustainability of road transportation," *Annual Reviews in Control*, vol. 48, pp. 292–311, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578819300446

[11] J. Huo, X. Wen, L. Liu, L. Wang, M. Li, and Z. Lu, "Chrt: Clustering-based hybrid re-routing system for traffic congestion avoidance," *China Communications*, vol. 18, no. 7, pp. 86–102, 2021.

[12] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1908–1916, 2019.

[13] J. Ding, H. Xu, J. Hu, and Y. Zhang, "Centralized cooperative intersection control under automated vehicle environment," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 972–977.

[14] Z. Cao, H. Guo, and J. Zhang, "A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, dec 2017. [Online]. Available: https://doi.org/10.1145/3078847

[15] Y. Wu, H.-N. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9214–9231, 2022.

[16] Dell Technologies, "Data-driven innovation starts at racing's edge to improve race car aerodynamics — and speed," Tech. Rep., 2021.

[17] S. Li, "Multi-agent deep deterministic policy gradient for traffic signal control on urban road network," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, 2020, pp. 896–900.

[18] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[19] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," 2020.

[20] X. Liang, X. Du, G. Wang, and Z. Han, "A deep reinforcement learning network for traffic light cycle control," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1243–1253, 2019.

Hani Kamal received his M.Sc. in Computer Science from Birmingham City University, UK, and a B.Sc. in Civil Engineering from the University of Jordan. Currently, He is a Software Implementation Engineer, in London, UK.

Wendy Yánez received her M.Sc. and PhD degrees from the University of Birmingham, UK. At present, she is an Assistant Professor at the University of Birmingham, UK.

Sara Hassan was born in Alexandria, Egypt, in 1994. She received her MEng on and PhD degrees from University of Birmingham, UK. At present, she is a senior lecturer in software engineering at Birmingham City University, UK.

Dalia Sobhy received her M.Sc. degree on 2014 from the Arab Academy of Science and Technology and Maritime Transport, Egypt, and PhD on 2019 from the University of Birmingham, UK. Currently, she is an Assistant Professor at the Arab Academy of Science and Technology and Maritime Transport, Egypt.