Cyber-Physical System and Deep Reinforcement Learning-Enabled Driving Behaviour Study in Connected and Autonomous Driving System

Xiangyu Li, Ding Cao, Ivan Wang-Hei Ho

***Abstract*—This project aims to investigate and demonstrate a cyber-physical system (CPS) and deep reinforcement learning (DRL)-enabled driving behaviour system for the design and optimization of vehicular networks and road safety assessment of cooperative autonomous driving in urban areas. The virtual environment can be interpreted as a digital twin or metaverse, a 3D representation of a real-world system, which has been widely adopted in Industry 4.0. From the perspective of research, the cyber-physical system can help us realize conditions that cannot be realized in reality, so that we can use the virtual world to conduct multiple simulations and obtain simulation data for analysis. Since urban areas with dense traffic and high-rise buildings pose critical concerns on vehicle-to-everything (V2X) communications, and hence vehicular traffic safety and efficiency, a safe and well-controlled virtual environment for evaluating and assessing various connected driving scenarios is of paramount important. To fully exploit the potential of V2X message in busy cities, the virtual environment also needs to have a realistic characterization of the street layout, buildings, wireless communications, and the mobility of human-controlled and autonomous driving vehicles. V2X communication technology is the enabler for advanced driver assistance systems (ADAS) for cooperative autonomous driving in smart cities. As assisted driving technologies and connected vehicles are becoming more popular, the market will have a significant need for integrated products. Together with initiatives that require vehicles to be equipped with onboard units and perform V2X communications, it is foreseeable that our cities will be filled with connected self-driving vehicles that drive cooperatively in the future. Therefore, we propose to apply V2X communication technology to assisted driving and test the performance of various products in a safe, reliable, and virtual environment. I have participated in the initial development of a digital twin simulation platform for emergency message warning applications based on dedicated short-range communications (DSRC), but further development is required, and the virtual environment must be verified through real-world experiments to ensure an effective foundation. In this proposal, CPS will be built and tested in different environments with different participant to explore the potential of digital twin technology in connected and autonomous driving system (CADS). Moreover, the application of deep reinforcement learning (DRL) will be carried on to evaluate the possibility of training autonomous vehicles in such environments. Findings from the future project can provide references for CPS and autonomous driving. The methodologies used in this project can contribute to further V2X systems development.**

***Index Terms*—Traffic Signal Control, Reinforcement Learning, Urban Traffic.**

Xiangyu Li is the Department of Civil and Environmental Engineering, University of California, Berkeley, United States

Ding Cao and Ivan Wang-Hei Ho are the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong

I. INTRODUCTION

A Cyber-physical system (CPS) is the virtual (often 3D) version of the real-world system that uses the real-world system's data regarding the characteristics and attributes of physical objects as input and outputs simulations or predictions that change depending on those inputs. Particularly, there is frequently real-time data stream communication between the matched real-world system twin and the virtual twin. The virtual twin depiction will be updated via the real-world system's sensors, allowing for real-time monitoring of the real-world system's condition. In this method, the virtual system twin is programmed to execute the fictitious scene in order to find the best parameters, in addition to updating the state of the real-world system. In 2003, Professor Grieves of the University of Michigan originally proposed this idea for the upkeep and defense of aerospace vehicles' health [1].

Advanced driver-assistance systems (ADAS) are one of the fastest-growing intelligent transportation system (ITS) safety applications for reducing traffic accidents and enhancing traffic efficiency. It utilizes electronic devices to assist drivers in driving and braking functions through a human-machine user interface (UI). ADAS normally use Internet-of-things (IoT) technology, such as sensors and cameras, to detect nearby vehicles or driver errors, and respond accordingly to reduce risk and improve road safety. Some examples of ADAS that have progressed are listed below.

1) Adaptive cruise control (ACC) that can automatically accelerate or brake considering the distance between the ego vehicle and the vehicle ahead [2].
2) Collision avoidance systems (CAS) that utilizes radar sensors installed in the front of the vehicle to detect nearby obstacles and inform the driver of potential crash situations in advance [3].
3) Lane change assistance that helps drivers make lane changes safely by using sensors to scan nearby vehicles and monitor the driver's blind spots [4].

In addition to the above examples, there are many other similar microscopic ADAS applications, and some of them are based on IoT sensors that perform calculations to assist the driver. The vehicle-to-everything (V2X) communication system [5] is different, it can realize the interactions among vehicles, the infrastructure, and pedestrians, to achieve more extensive and complicated information dissemination. Compared with other applications, the advantages of ADAS assisted by V2X communications are as follows:

1) V2X-aided ADAS are not limited by the sensors on the vehicle, it can also receive information from other vehicles and roadside units to make driving decisions.
2) With the onboard units (OBUs) and roadside units (RSUs) installed on vehicles and on roadside resepectively, the V2X-aided ADAS can have an extended range of sight to provide wiser driving instructions and warnings to drivers.
3) In general, V2X-aided ADAS can achieve better performance in terms of information range and efficiency.

The accident rate is always rising due to the growing number of vehicles on the world's highways. Formulating reasonable and appropriate coping mechanisms to avoid collisions and traffic accidents by researching driving behaviour is an efficient technique to lower the accident rate. Therefore, research on the relationship between driving behaviour, risk perception, and traffic accidents is urgently needed. In the previous literatures on driving behavior, such as [6], the most commonly used method is to collect participants' opinions through driving behavior questionnaires (DBQ) and evaluate the quality of driving behavior through driving behavior data such as the standard deviation of speed (SDspeed). In this proposal, we accept this questionnaire method of inviting participants to judge the impact of V2X messaging applications in our CPS system on driving behavior.

As an example of practical application, illegal pedestrian crossing (IPC) is a case with a risk factor as high as 93.5% of road dangerous behaviors. By setting the occurrence probability of IPC in CPS, participants will experience one or more IPC events on the way. At the same time, the On board unit (OBU) installed on each vehicle will send a V2X message to the rear vehicles, informing them that they need to slow down urgently to avoid a collision. The invitation of participants can be divided according to age and driving proficiency. For example, the drivers' age is divided into [18, 28], [29, 39], [39, 49], [49, 59], [59, 69], [69, 79] groups etc. Proficiency can be divided into, the driving age is [0,10], [10,20], [20,30], [30,40] and so on. The DQB questionnaire will set the level of acceptance of the V2X news, and the driving data will be collected. After collecting a huge amount of driving data, we were able to use this data for reinforcement learning. Deep reinforcement learning is very popular in the simulation of traffic and the modeling of driving behavior [7-8]. By setting appropriate states, actions and rewards, the vehicle can learn the optimal driving route, speed, acceleration, car following and lane changing behavior.

This project is based on the cyber-physical system for the vehicular networks and intelligent transportation systems. After implementing the algorithms in the CPS, the driver (player) can react according to different V2X messages, and we can collect a huge amount of driving data through the physical driving simulator. These driving data are invaluable for evaluating the efficiency of the proposed algorithms and models, studying the impact on drivers, as well as training deep learning models for autonomous vehicles that are sensitive to V2X messages. In this project, we set up an emergency vehicle broadcast message scenario based on the digital twin platform as shown in the figure below.

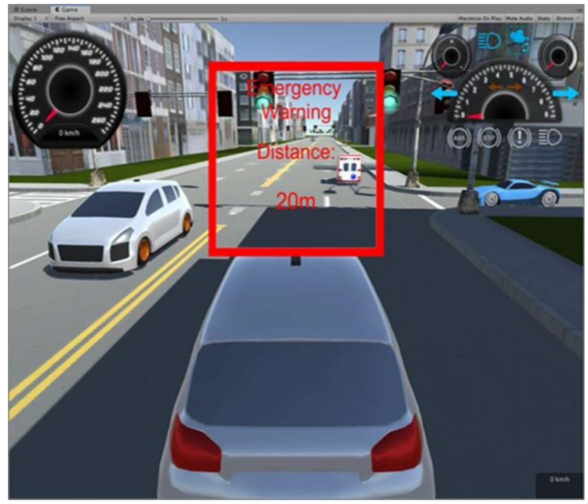

Fig. 1. Historical Trend for Time Spent in Highway Congestion - Bay Area .

We want to train an automatic driving system that can perfectly adapt to the emergency vehicle situation, and improve the efficiency while ensuring the safety of the overall system, that is, the average waiting time. We collected the standard deviation of speed (SDspeed), standard deviation of heading error (SDHE), mean heading error (meanHE), standard deviation of lateral position (SDLP) data for the simulations on the Logitech G29 driving simulator. We use these data to train the autonomous driving system, optimize these four parameters to ensure the safety of each autonomous vehicle, and ultimately reduce the average waiting time of the entire system.

II. Methodology

A. The development of the cyber-physical system

1) Driving behavior capturing with the physical driving simulator: The physical driving simulator plays an important role in this research because we use it to acquire human driver inputs to the virtual environment and collect real-time driving data from it. Some racing game controllers available in the market such as the Logitech G29 steering suite would be suitable for constructing the driving simulator, because it provides real-time data collection and near-real driving experience.

2) Driving behavior capturing with the cyber system: To the best of our knowledge, there are few existing joint traffic and communication simulation platforms for ITS, and the commercial platforms are mainly autonomous driving platform, such as Baidu Apollo [9]. It is also hard to find an existing digital twin platform for vehicular networks, the only references are IBM Digital Twin Exchange, Gazebo, and Microsoft Azure for Industry 4.0.

In the prototype, the proposed physical and cyber systems will be connected bidirectionally in real time via a communication protocol (e.g., TCP socket). The physical system captures human driving data in real-time, such as speed, acceleration, direction. The cyber system consists of the traffic and V2X simulators. For the traffic simulation, the simulation of urban mobility (SUMO) and Aimsun are common vehicular traffic simulators. Unity 3D and Unreal engine 4 are also good choices for 3D traffic simulation, as they can provide the integration of 3D models, scripts, and viewpoints. For the V2X simulation, OMNeT++, NS3, and Matlab wireless communication toolbox will be exploited in this project.

Eclipse simulation of urban mobility (SUMO) [10] is a microscopic and continuous multi-modal traffic simulation package designed to handle large traffic. The driving parameter can be exported to other software in real-time via traffic control interface (TraCI) [11] via TCP socket. SUMO can serve as the traffic simulation server and provide data exporting and importing to and from other simulators. Open Street Map (OSM) is a database of street layout, building information, and road information in the world, which can be integrated with SUMO for realistic traffic simulation.

Objective Modular Network Testbed in C++ (OMNeT++) Discrete-Event Simulator is a modular, component-based C++ simulation library and framework, primarily for building network simulators [12]. VEINS is an open-source event-based framework [13] for running vehicular networks simulations between OMNeT++ and SUMO via TraCI. They will be used as the network simulator to calculate and transmit V2X messages to the game engine.

Unity is a cross-platform 3D game engine [14] for creating a 3D virtual world to compile the joint traffic and network simulation. Unity also serves as the gateway between the physical and cyber systems, where human drivers can control virtual vehicles in the large-scale 3D virtual simulation environment. The integration of these systems is mainly based on TCP connections, the traffic simulator is the server while other simualtors are clients that establish real-time connections with it. Traffic and network data are transmitted bidirectionally among these simulators for the virtual V2X-aided ADAS platform to operate.

Features:

1) Design V2X message in network simulator
2) Connect to the physical part environment in real time
3) Multi-user access possible
4) High data rate transmission
5) Obtain driving data from the software

B. Cross-layer protocol design for safety message dissemination

Dedicated short-range communications (DSRC) [15] and Cellular-V2X [16] are existing networking technologies for connected vehicles. For the transmission of the safety and warning messages, there is the wireless access in vehicular environments (WAVE) [17] protocol in DSRC for real-time message transmission, including the basic safety message (BSM), WAVE short message (WSM). Self-driving vehicles or drivers can make driving decisions based on these messages. We will propose more efficient and secure message dissemination protocols based on BSM and WSM, which can have faster urgent message transmission and better channel utilization. In WAVE (IEEE 1609.4 standard), multiple channel operation may result in high packet contention in the vehicular network, especially when the vehicular density is high. A solution is to design a suitable channel switching mechanism to balance the contention. In addition, the DSRC protocol stack lacks the mechanism of message verification and retransmission, and the receiver (driver) will not know about the lost messages. This may lead to critical hidden danger in V2X-aided ADAS, and we will address this flaw in the protocol design from a cross-layer perspective based on the analyses conducted with the developed virtual platform.

Traffic modeling is a popular area in transportation and ITS studies, it usually contains microscopic traffic mobility, driver behaviors, traffic control algorithms, etc. For research in vehicular networks, macroscopic stochastic traffic mobility is usually considered for evaluating the performance of different V2X communication protocols. However, it might not be realistic enough as there are different types of traffic scenarios in real-world vehicular networks, such as highway, ramp, intersection, urban street, etc. Under different traffic scenarios and driving strategies, it will lead to different communication impact and hence the performance in ITS applications. With the virtual V2X-aided ADAS platform, the existing V2X protocols can be rigorourly and conveniently evaluated and optimized with respect to different traffic situations.

Features:

1) Design V2X message in network simulator
2) Use of existing V2X protocol stacks (DSRC, CV2X)
3) No interference with existing wireless networks
4) Multi-user access possible
5) Real-time high-speed data exchange of traffic-relevant content with the cloud

C. Autonomous driving deep reinforcement learning models

After implementing the algorithms in the CPS, the driver (player) can react according to different V2X messages, and we can collect a huge amount of driving data through the physical driving simulator. These driving data are invaluable for evaluating the efficiency of the proposed algorithms and models, studying the impact on drivers, as well as training deep learning models for autonomous vehicles that are sensitive to V2X messages. Such deep learning model plays an important role in the research of self-driving vehicles to improve existing models such as the cooperative adaptive cruise control (CACC) model commonly used in many traffic simulators.

The experiment is divided into multiple tasks in different scenarios. We will first invite drivers of different backgrounds to conduct the experiment. According to their driving proficiency, we will test the efficiency of our V2X-aided CPS for experienced, general, and novice drivers. They will use different levels of V2X-aided CPS to drive freely in multiple driving scenarios in the virtual world, requiring safety and compliance with respective to different traffic rules. There will be stochastic traffic conditions in each trial, such as traffic accidents, running traffic lights, speeding, pedestrian's rushing out, etc. After each experiment, drivers will be asked to rate the user experience of the assisted driving system, and we will adapt the amount and frequency of V2X messages accordingly to optimize the effectiveness and driving comfort in specific road conditions.

With the collected dataset from the driving simulator, we will investigate deep learning models for V2X communication-sensitive autonomous driving and compare the performance with generic sensor-based models. Such model can be further imported into the traffic simulator for optimized cooperative autonomous driving. Regarding the training of the deep neural network (DNN) model, we can employ the most commonly used machine learning frameworks such as TensorFlow [18] and Keras [19]. In addition, the Unity game engine also listed the latest machine learning agent [20], which can be exploited to train autonomous driving models.

Features:

- Use of machine learning techniques
- Data collection from both the software and hardware

D. Driving behavior modelling and analysis

In the driving behavior questionnaire (DBQ), we will add the degree of adaptation to the V2X message warning, for example, the level of feeling very useful is 10 to 1, and 10 is the highest. Thus we can draw whether the effect of such notification messages is useful for those with higher driving proficiency. Similarly, this setting is also effective in studying age groups, and we can know whether V2X message promotion is more helpful to the elderly. Several questions in this section gauge participants' unsafe and aggressive driving behavior, which may include speeding, tailgating, distracted driving, and failing to buckle up. Other inquiries looked into the participant's response to encountering rude or hostile behavior from other drivers or users of the road. To mathematically analyse the driving behavior, we will propose several metrics to evaluate it, such as SDspeed. Others can be standard deviation of the acceleration, heading erorr, and so on.

Features:

- Mathematic analysis of driving behavior
- Data collection from different settings of experiment group

III. System Model

A. Traffic Mobility Model Generation

In the research of the Internet of vehicles, if we use real vehicles and real roads for experiments, it will be a great expense, and the controllability is not high, such as adjusting the average speed of vehicles. For this reason, we need to build a traffic mobility model based on real vehicles and roads, in this way, we can directly analyze the performance of different traffic scenarios by modifying various parameters and locations in the mobility model. The existing traffic mobility models (e.g., the Random Waypoint Model, Manhattan Model), cannot fully reflect the real traffic conditions, nor can they capture the real vehicle action. In other words, the real modeling of vehicle trajectory, speed, and other mobility parameters have a very important impact on the performance of the vehicular networks. There are many methods to generate a real traffic mobility model. In the design of firmware distribution, we use open street map (OSM), SUMO, ns3 software to generate traffic mobility models and simulate them. The specific steps to generate a mobility model in Tsim Sha Tsui, Hong Kong is given below.

B. Realistic Tsim Sha Tsui Traffic Mobility Model

Nowadays, the urban traffic model is the communication carrier in the research of the Internet of Vehicles. In urban road topology, there is a strong relationship between vehicles, and their movements are also related. When selecting the location of the urban traffic model, we need to consider the following points:

1. Vehicle types and their trajectory Different vehicle types determine the vehicle's trajectory, speed limit, and other parameters. For example, the trajectory of private cars is mostly different, and the randomness is very strong. But the bus, the school bus's traveling track has a fixed route. Emergency vehicles, such as police cars, ambulances, and fire engines. Their mobility is protected by law. They can violate traffic rules when performing emergency tasks, such as running red lights and exceeding the speed limit on the road.

2. Traffic density The traffic density of different places in the city is different, and the traffic density of the same place at different times is also different. Some urban roads are crowded for a long time every day, some urban roads are idle, and some roads are only crowded during rush hours. In some specific areas like bus stops and parking lots, the traffic density is always high.

3. Characteristics of high-speed motion vehicles: The biggest difficulty in vehicular networks is the high mobility of vehicles, which will cause the vehicle nodes to disconnect from communication from time to time. Even in the high-density traffic flow model, the communication link will be easily damaged. Besides, high-speed vehicles will introduce inter-carrier interference (ICI), which will weaken the communication signal.

The above points are taken into account when building a real traffic mobility model. To meet the above requirements and for the sake of local traffic conditions in Hong Kong, we selected Tsim Sha Tsui, a famous landmark in Kowloon, Hong Kong, as the realistic area for simulation analysis. The software we used is SUMO and Open Street Map (i.e., osmWebwizard in the SUMO library).

The above points are taken into account when building a real traffic mobility model. To meet the above requirements and for the sake of local traffic conditions in Hong Kong, we selected Tsim Sha Tsui, a famous landmark in Kowloon, Hong Kong, as the realistic area for simulation analysis. The software we used is SUMO and Open street map (i.e., osm Webwizard in SUMO library). First of all, the traffic scenes we designed are summarized in Table 2.1 below. We consider three different types of traffic conditions in Tsim Sha Tsui (i.e., Sparse, Moderate, Congestion). The number of buses and vehicles is 123, 408, and 513, respectively.

TABLE I
MULTI-INTERSECTION DQN'S ACTION SPACE LOGIC TABLE

Location	Traffic conditions	Vehicle number in this area
Tsim Sha Tsui (474.86 × 418.99 m2)	Sparse	123 (bus+vehicle)
Tsim Sha Tsui (474.86 × 418.99 m2)	Moderate	408 (bus+vehicle)
Tsim Sha Tsui (474.86 × 418.99 m2)	Congestion	513 (bus+vehicle)

C. Project setting

The data is collected from different types of participants on the driving simulator, and preprocessing was made on the data. After that we used the Simulation of Urban MObility (SUMO) simulator platform, on a windows-based operating system, where I made single agent simulations, and multi agent (2 agents) simulations, which we can see further in this document. The data of cars flow was generated in the SUMO simulator, and was transferred to python with the Traci library.

D. Single agent case

In this setting we first create a traffic simulation which only contains a single intersection. The traffic is coming from all sides of the road, and we named each side of the road as North, East, West and South. Each road has three options to cross the intersection – Going straight, turning left or right. For traffic generation we use Simulation of Urban Mobility (SUMO), as shown in the picture below.

Fig. 2. Historical Trend for Time Spent in Highway Congestion - Bay Area.

E. WHAT IS Reinforcement Learning?

Reinforcement Learning is a branch of Machine Learning with a specific structure and flow, as described in the chart. An RL agent performs an action in an environment, which causes some kind of reaction/result in the system. The result that appeared yields a reward – whether the consequences were good or bad. These results are being transferred back to the agent, which adapts itself corresponding to the reward it got from his last action. RL algorithms can be divided into two types: model-free RL algorithms and model-based RL algorithms. In model-free RL algorithms, we do not have an exact model of the environment, which means that we do not know what will happen in the next time step after we performs an action. On the other hand, in model-based RL algorithms, agent must have to learn the model of the environment which may not be available in most of the real-world problems.

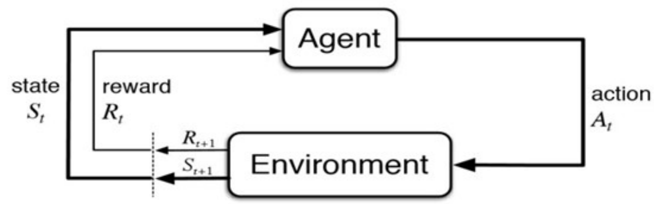

Fig. 3. DQN process.

In this work I used model-free RL algorithms to control the overall automatic driving system. Model-free algorithms further can be classified as value-based (Q-learning and QL with Neural Networks = DQN) and policy gradient (A2C, PPO) algorithms. In this work we will work with both value based and policy gradient algorithms, and examine their success on our problem.

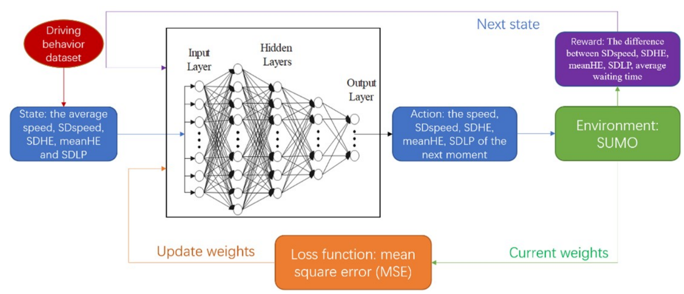

Fig. 4. DQN.

Since we didn't learn these in class, I'll elaborate: A2C – The Advantage Actor Critic algorithm is somewhat of a merge between the value-based and policy-gradient methods. Each iteration, it computes two values – the Critic function, which is the same function the DQN goes by, to minimize the error, and the Actor function, which takes the Critic's result in consideration and updates the policy the agent goes by.

PPO - The Proximal Policy Optimization algorithm does what on-policy algorithms do, which is to continuously update the policy, so that the agents using the policy will be guided to making the optimal decisions for the entire system, to maximize our reward OVERALL (opposed to maximizing it in the short term, like DQN)

F. State setting

We used the average speed, SDspeed, SDHE, meanHE and SDLP in the automatic driving system as the state input.

G. Action setting

We used the speed, SDspeed, SDHE, meanHE, SDLP of the next moment as the action space.

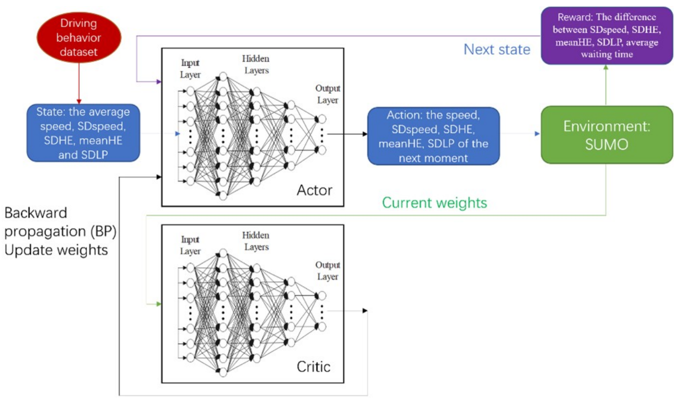

Fig. 5. A2C.

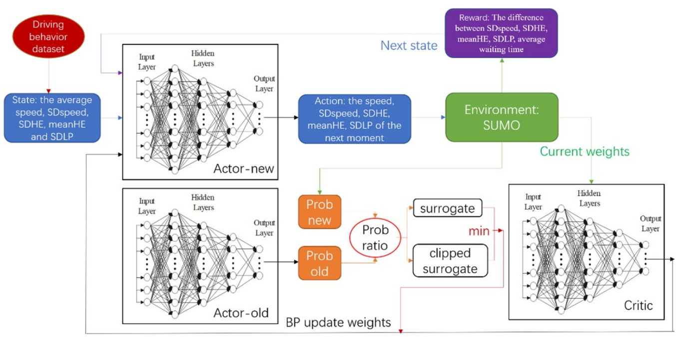

Fig. 6. PPO.

H. Reward setting

The difference between SDspeed, SDHE, meanHE, SDLP, average waiting time at the next moment and SDspeed, SDHE, meanHE, SDLP, average waiting time at the previous moment. The reward values are the differences.

IV. Expected Results and Implication of Results

A. Expected Results

The proposed Cyber-physical system and V2X message can meet the needs of smart city and can be commercialized. They can fully demonstrate the advantages of CPS technology in CATS systems and the advantages compared with traditional technique/measurement.

For simulation and data analysis, they can fully demonstrate the feasibility and high reliability of deep reinforcement learning and driving behavior improvement.

We expect that the CPS system can effectively help collect the data of the family members of the participants, which can be used to analyze the effectiveness of our proposed V2X message reminder and help improve road safety. Secondly, these data can also be used to train self-driving vehicles to optimize the metric of driving behavior.

B. Practical Implications

- Findings from this proposal can provide references for cyber-physical system design.
- The methodologies used in this project can contribute to further CPS and V2X communication development.
- Potential deep reinforcement learning algorithm (e.g., PPO, DDPG) for the CPS will be discovered.

C. Theoretical Implications

- Theoretical findings from this proposal can help strengthen the existing theory in CPS technology and create new idea for further research.
- Theoretical findings can help us understand the relationship between driving behavior and CPS.

D. Experimental results- Single Agent (Intersection) problem:

We first run Random agent, which makes random actions, which aren't a result of his previous actions rewards, and fixed agent, which gives each vehicle a fixed action such as "speed+=5", and compare their performance. Figure 7 clearly

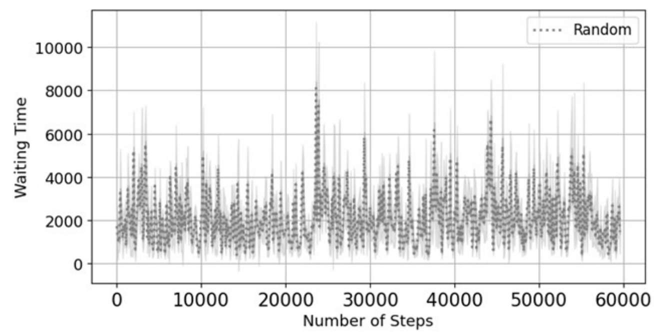

Fig. 7. Random action space.

shows that random action space does not learn anything (as expected), and the waiting time doesn't improve overtime. As

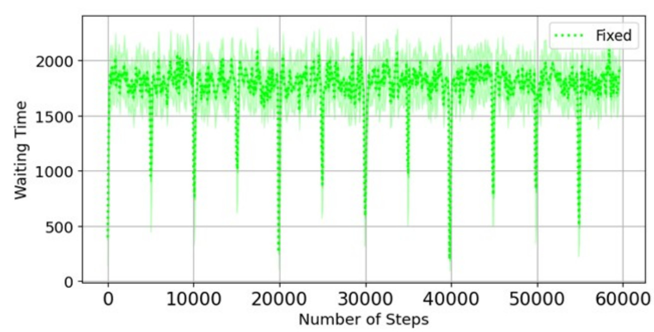

Fig. 8. Fixed action space.

shown in figure 8 and 9, fixed action space has much better performance than random action space in terms of speed (e.g., avg. waiting time), but still – as it's fixed, it doesn't improve overtime, and doesn't learn about the real flow of the traffic in the intersection.

Now, we move on to trying to manage the driver behaviors (SDspeed, SDHE, meanHE, SDLP) to be responsive of the actual traffic in the intersection, using DQN: DQN seems to converge and improve its results overtime, but the final results aren't outstanding. The DQN algorithm is an off-policy algorithm, which tries to maximize the reward it gets

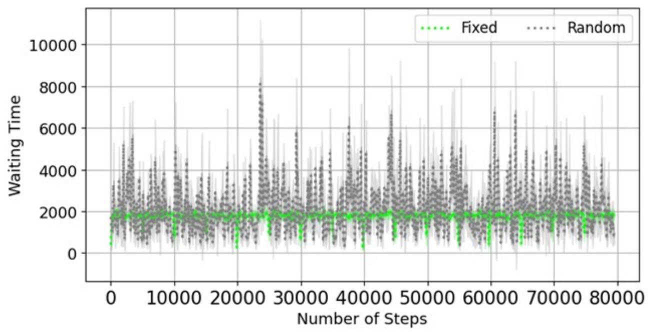

Fig. 9. DQN result of single agent traffic.

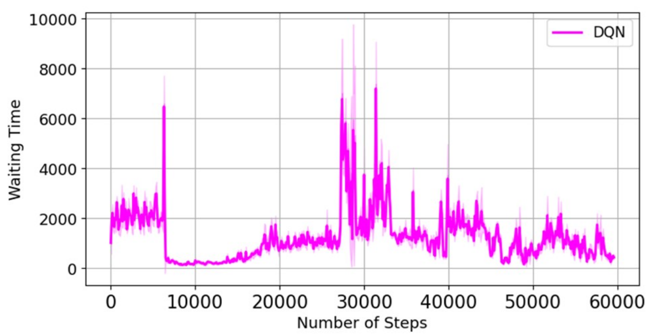

Fig. 10. DQN result of single agent traffic.

in each iteration separately. This approach gets us pretty good results, but not optimal. Optimal results can be achieved with algorithms that learn a POLICY to go by (like a manual for the agents), that way we achieve a MACRO view instead of a MICRO view in the DQN algorithm.

After realizing on-policy algorithms might perform better in this problem, let's take a look on their result, in the next page. Let's check the A2C performance: As we can see in the chart,

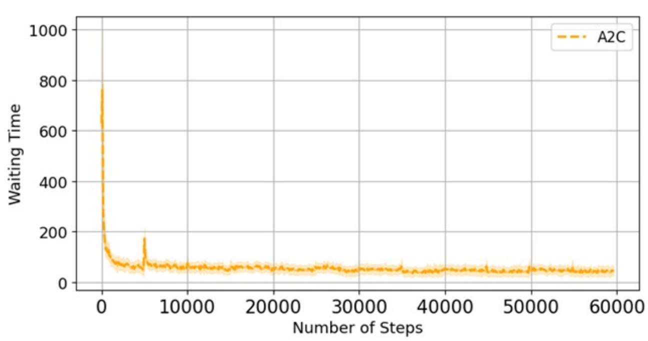

Fig. 11. A2C result of single agent traffic.

A2C algorithm manages to successfully deal with the single intersection driving behavior modelling problem and controls each vehicle in a way that minimizes almost completely the cars' waiting time.

Let's also check PPO performance: As they are relatively similar, PPO algorithm also performs well in the single agent problem.

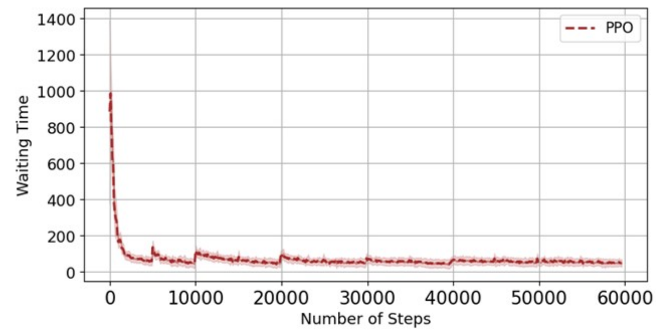

Fig. 12. PPO result of single agent traffic.

Single agent conclusion:

As shown in Figure 7:

Random Action – cannot learn any driving behavior.

Fixed Action – cannot learn any driving behavior, the result don't improve overtime.

DQN – Converges successfully overtime and is better than fixed, but still performs worse than on-policy algorithms.

A2C, PPO – Waiting time decreases massively and end up minimized – solves and deals well with one intersection. Multi-Agent:After finding a good driving behavior model and

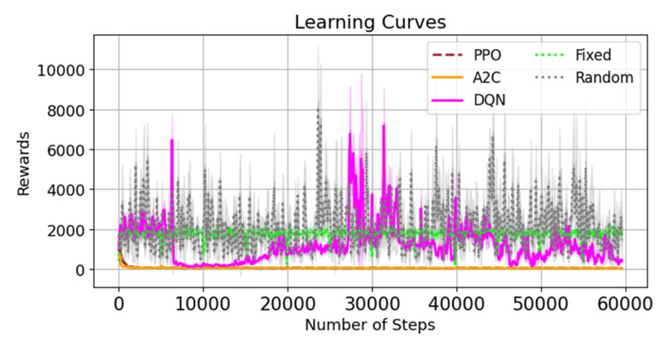

Fig. 13. Reward comparison of different RL of the single agent.

minimizing waiting time in the Single Agent problem (just one intersection), we move on to the multi-agent problem – The real world. In real life, there are a lot of intersections, where one road you take in an intersection leads you to another in the next one. We'll attempt to deal with this problem as well. In the picture above, we can see two intersections (therefore we have two agents), where each agent is controlling its own intersection. Here are the results:

E. What can we see?

DQN again successfully converges after a decent amount of learning, but is again worse than A2C and PPO (look at the y axis' scale).

As for A2C and PPO, not only we succeed at solving this problem, the waiting time for all cars in all intersections combined is the same as it was in the single agent problem.

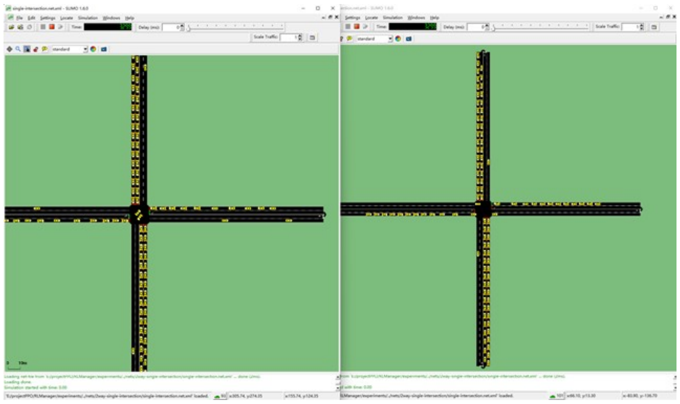

Fig. 14. Reward comparison of different RL of the single agent.

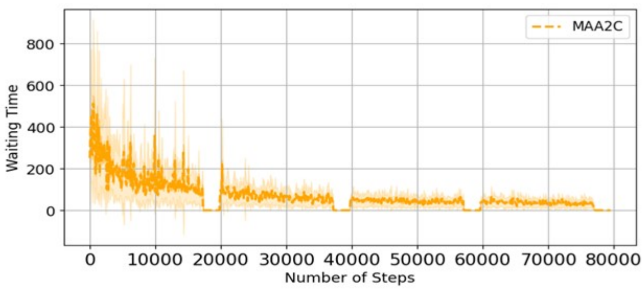

Fig. 15. A2C result of the multiple agent.

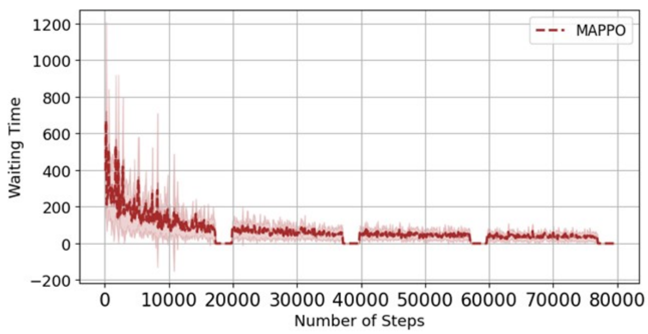

Fig. 16. PPO result of the multiple agent.

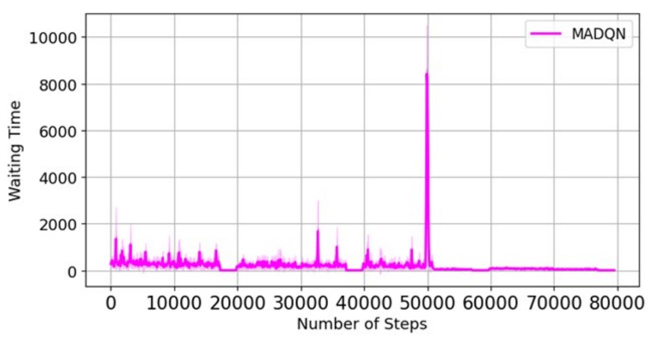

Fig. 17. DQN result of the multiple agent.

F. How is this possible?

The cause of this is the fact that in single agent, the agent has no idea where the traffic will come from. In Multi Agent we have a big advantage – the agents communicate, and tell each other – "Cars are coming from your side, be aware". This way the Multi Agent problem handles far more cars and intersections, but can maintain similarly low total waiting times. After seeing the great result with 1 2 intersections, we can look at the real world, where there's a net of intersections, all connected to each other, where cars flow in every direction.

G. Model Comparison Conclusion: We've seen that PPO and A2C work better than DQN on this case of problem. Why?

A – A2C tends to succeed in scenarios where information needs to be transferred between agents in an environment.

B – This problem fits the on-policy model better, since it's very behavior-based, and doesn't converge well when actions are always taken in a greedy manner (like DQN).

C – Having a policy that is learned and can guide the agents what to do next helps us here, since it provides a lot of information to the agents, who can now make their decisions based on the entire environment instead of just making sure they do the best thing for their own "small world" in the specific moment in time. Also, a policy to guide the agents provide us with a macro view that is capable of capturing the whole scene, instead of each agent optimizing itself just by the next move.

V. Conclusion

Having a policy that learns from a macro view overtime works far better in this problem than having agents who each optimize their own benefit in every single independent action. This thesis includes three parts:Firmware distribution in vehicular networks;Digital twins for vehicular networks;Channel estimation for the vehicular environment.They are all hot topics on the Internet of vehicles and closely related. In firmware distribution,we mainly focus on the efficiency of data dissemination, proposed methodologies,and algorithm in many aspects.Ns-3 and SUMO are mainly used in this part,and their code is mainly composed of C++language.And then there is the Linux Ubuntu system for configuration. In digital twins,we mainly focus on its function in ITS application.For hardware twin, the configuration of Logitech G29 driving simulator is configured and used.For software twin, we consider OSM,SUMO,Unity 3D,OMNeT++.The main work focuses on Unity 3D and OMNeT ++The languages used in the code are C (for Unity 3D scripts),C++and NED (for OMNeT++).There is a Windows 10 system for implementation. In channel estimation,we mainly focus on the BER performance of the channel estimator in the vehicular network.This part mainly includes theoretical calculation,training set data collecting,and deep neural network(DNN)training.We use Matlab for the OFDM system and BEM simulation,TensorFlow,and Keras deep learning framework,the programming language is mainly python.

Acknowledgments

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

REFERENCES

[1] J. Wu, Y. Yang, X. Cheng, H. Zuo and Z. Cheng, "The Development of Digital Twin Technology Review," 2020 Chinese Automation Congress (CAC), Shanghai, China, 2020, pp. 4901-4906, doi: 10.1109/CAC51589.2020.9327756.

[2] MarkVollrath. Schleicher. Susanne. Gelau. Christhard. 19 December 2010. The influence of Cruise Control and Adaptive Cruise Control on driving behaviour -A driving simulator study. Accident Analysis Prevention. 43. 3. 1134–1139. 10.1016/j.aap.2010.12.023. 21376911. 0001-4575.

[3] How Pre-Collision Systems Work, 2022. [Online]. Available: https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/pre-collision-systems.htm

[4] Habenicht, Stefan; Winner, Hermann; Bone, Sven; Sasse, Fabian; Korzenietz, Peter (5 July 2011). "A maneuver-based lane change assistance system". 2011 IEEE Intelligent Vehicles Symposium (IV): 375–380. doi:10.1109/IVS.2011.5940417. ISBN 978-1-4577-0890-9. S2CID 9690965.

[5] Arena, Fabio; Pau, Giovanni (2019-01-24). "An Overview of Vehicular Communications". Future Internet. 11 (2): 27. doi:10.3390/fi11020027. ISSN 1999-5903.

[6] Sayed, I., Abdelgawad, H. Said, D. Studying driving behavior and risk perception: a road safety perspective in Egypt. J. Eng. Appl. Sci. 69, 22 (2022). https://doi.org/10.1186/s44147-021-00059-z

[7] C. Wu et al., "Framework for control and deep reinforcement learning in traffic," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 1-8, doi: 10.1109/ITSC.2017.8317694.

[8] Song, Yaofeng Zhao, Han Luo, Ruikang Huang, Liping Zhang, Yicheng Su, R.. (2022). A SUMO Framework for Deep Reinforcement Learning Experiments Solving Electric Vehicle Charging Dispatching Problem. 10.48550/arXiv.2209.02921.

[9] Baidu Apollo, 2022. [Online]. Available: https://github.com/ApolloAuto/apollo

[10] Tonguz, Ozan K. (25 Sep 2018). "How Vehicle-to-Vehicle Communication Could Replace Traffic Lights and Shorten Commutes". IEEE Spectrum.

[11] Chowdhury, D., Santen, L., Schadschneider, A. (2000). Statistical physics of vehicular traffic and some related systems. Physics Reports, 329(4-6), 199-329.

[12] Information Resources Management Association. Networking and Telecommunications: Concepts, Methodologies, Tools, and Applications. Idea Group Inc (IGI); ISBN 978-1-60566-987-8. p. 592.

[13] Christoph Sommer, Reinhard German and Falko Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," IEEE Transactions on Mobile Computing (TMC), vol. 10 (1), pp. 3-15, January 2011.

[14] "Unity 2022.1a". Unity. Retrieved November 5, 2021.

[15] Harvey J. Miller and Shih-Lung Shaw (2001). Geographic Information Systems for Transportation. Oxford University Press. ISBN 978-0-19-512394-4.

[16] "Cellular V2X as the Essential Enabler of Superior Global Connected Transportation Services". IEEE 5G Tech Focus. IEEE. 1 (2). June 2017.

[17] Morgan, Y. L. (2010). Notes on DSRC WAVE standards suite: Its architecture, design, and characteristics. IEEE Communications Surveys Tutorials, 12(4), 504-518.

[18] TensorFlow, 2022. [Online]. Available: https://www.tensorflow.org

[19] Keras, 2022. [Online]. Available: https://keras.io

[20] Unity ML-Agents, 2022. [Online]. Available: https://github.com/Unity-Technologies/ml-agents.