Modern OpenAI Gym Simulation Platforms for Vehicular Ad-hoc Network Systems

Anjie Qiu, Donglin Wang, Sanket Partani and Hans D. Schotten
Chair for Wireless Communication & Navigation
University of Kaiserslautern
{qiu, dwang, partani, schotten}@eit.uni-kl.de

***Abstract*—The great demands of Machine Learning (ML) are required in many application domains. In the vehicular communications, new technologies and applications based on ML appear more frequently. Many studies already show significant benefits of deploying ML to the Intelligent Transport Systems (ITS), among which Reinforcement Learning (RL) delivers the best compatibility. Many vehicular networks have chosen Simulation of Urban MObility (SUMO) as the mobility simulator, which provides realistic traffic traces and real-world road maps. In recent years, many simulation platforms based on the SUMO as traffic/mobility part and the OpenAI Gym platform are put into usage, hence, this work serves as an overview of these modern and realistic simulation platforms, and a comparison is made from different aspects. Also, the advantages and disadvantages of each simulator are discussed, and recommendations on different simulators for researchers are introduced based on their research topics.**

***Keywords*—Reinforcement Learning, Vehicular Ad-Hoc Network, traffic simulation, OpenAI Gym**

I. Introduction

The simulation of the traffic is also vital for deploying new technologies from its theorem to the practice at a minimum or even without cost. Vehicular simulations draw more and more attention to researchers since recent communication technology has been proposed and developed, such as Sidelink communication (proposed in 3GPP release 14 [1]). Modern vehicular communication systems are based on wireless communications among moving or parking vehicles, which helps to improve traffic throughput, road safety, driving ability, etc. Thus, Vehicular Ad-Hoc Networks (VANETs) are proposed for the realistic exchange of messages on road, which includes the MANET as a subset [2]. Intelligent Transportation Systems (ITS) is proposed by European Telecommunications Standards Institute (ETSI) to support wireless communication between vehicles and infrastructures to deliver diverse traffic applications [3].

As for the state-of-the-art simulations for the vehicular network, Machine Learning techniques are considered a new era. Since the networks nowadays are highly designed and maintained by computational devices, a large amount of traffic data are collected and calculated both offline and online. However, these data are merely just buried underground, and their potential usage is wasted due to the high computational complexity. Thus, the idea of deploying Machine Learning algorithms on the network is to dig deeper into traffic data, like vehicle speed, position, accident rate, and congestion rate, and make use of them. Also, various algorithms and learning techniques, like Deep Learning, Reinforcement Learning can be deployed to achieve optimized results for vehicular communication systems.

With the fast development and evolution of the ML technology, there is an option to adjust the network configuration through the Deep Neural Networks (DNNs). Also, a better deployment is to use the RL, which is no longer necessary to produce potentially large training data sets with known desired outcomes before the training phase. Instead, a *reward* function is to be created, that determines and evaluates, whether a particular behavior is good or not, according to a given state of the environment. Then an *agent* will explore the environment to collect observations, and then make a new behavior based on the prior training (or guessing), and adapt to itself by the above-mentioned reward function.

OpenAI Gym, proposed by Brockman et al. [4], is well known among the RL community. A standardized interface to an environment is provided by Gym for an RL agent, which reduces entry barriers and necessary boilerplate, and offers easy extensions to other projects. The Gym is therefore widely used in many domains, such as road traffic control [5], and wireless communication networks [6]. To our knowledge, there are already well-developed benchmark tools based on Gym for the Vehicular Ad-Hoc Network (VANET) research, such as the Flow-Project [7], Veins-Gym [8], SUMO-RL [9].

This study aims to summarize the recent advanced VANET simulation platform based on the ML/RL, and the work can be summarized as follows:

- Capability of different VANET simulators with ML/RL are studied
- A comparison among different simulators is made
- Suggestions for different simulators for different research areas are proposed

The remainder of this paper is organized as follows. In Section II, the major traffic simulator is introduced. In Section III, different VANET simulators based on ML/RL are discussed. In Section IV, a comparison between introduced simulators is made, and suggestions for different simulators are given. In Section V, the conclusion is drawn.

II. Traffic Simulators

Traffic simulator for VANET communications shall provide realistic traffic patterns and variable on-road components, including personal cars, freight vehicles, public transport, and even pedestrians. Therefore the SUMO [10], developed by the German Air and Space Centre (DLR) [11] in 2000, can be recommended as the traffic simulator. SUMO is an open-source, microscopic and multi-modal traffic simulation, which offers user-defined traffic demand modeling throughout a given road network, that can be model as needed or use the real-world maps from OSM. Since SUMO is microscopic, each vehicle is explicitly modeled and has its own driving route, own vehicle type, and this also fits the modeling of pedestrians, and the basic components of SUMO are shown in the Figure 1.

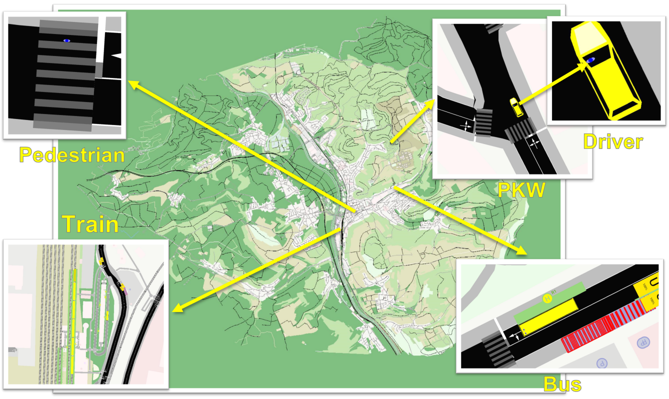


Fig. 1: Basic on-road moving components of SUMO, demonstrated in the city map of Merzig in Germany

Variety of helpful tools are also provided by SUMO and its community, such as:

- **osmWebWizard** - extract and create simple real-world scenario from OSM, using web browser
- **Interfacing TraCI from Python** - offer simple interface between SUMO and Python
- **Xml Tools** - convert SUMO output into CSV/Spreadsheet and vice versa
- **Visualization Tools** - offer simple and various visualization of simulation outputs
- **randomTrips** - generates simple random routes for vehicles on demands
- **activitygen** - generates realistic multi-modal routes for vehicles, pedestrians, buses, etc.

Among them the Traffic Interface (TraCI) is most important tool, since it bridges the gap between the traffic simulator and the network simulator via TraCI interface in Python. The VANET simulator involved in this work all use the SUMO as the traffic simulator and also use the TraCI to deliver traffic information to the network simulator, such as geographic locations, vehicle speed, vehicle acceleration, heading angle, and etc.

III. Reinforcement Learning Platform

ML/RL frameworks are made accessible to the vehicular networks in the following open-source simulation platforms, i.e., the *ns3-Gym* [12], *flow project* [7], *veins-Gym* [8], and *sumo-rl* [9]. All the above-mentioned projects deploy the OpenAI gym as the interface to the environment since the interface is standardized and well acknowledged to the community. In addition, the environment class offered by gym can be redesigned and extended as the need of the users, as well as the observation and action spaces. After the simulation run, the environment can be reset with the simple calling function *env.reset* from gym. The reward function can be selected and defined by users and the *action* function can be called by the RL-agent to perform actions to the environment. After the whole simulation, a reproducible result can be obtained by simply applying the pseudo-random number generators, if needed.

As the agent needs real-time traffic information, the introduced ML/RL projects offer a simple interface from the network layers to the traffic layers of SUMO, via the TraCI. The RL-agents collect all the necessary traffic data, such as average vehicle speed, lane occupancy, average waiting time, etc., and also the geographic position, including de-/acceleration, heading angle and CO_2. In addition, since the SUMO supports modeling the pedestrians and the traffic lights, the RL-agent can be also deployed on the walking persons and the Road-Side Units (RSUs). In addition, the RL-agent can also be found in the network layer, such as during the (de)modulation and for Handover predictions, according to the user-specific problems.

A. *FLOW project*

Flow [7] is a deep reinforcement learning framework for mixed autonomy traffic. It offers the user justifiable traffic scenarios, and integrated Deep Reinforcement Learning libraries. The system-level architecture of Flow is shown in Figure 2. The environment module delivers the states to the RL-agents simultaneously during the simulation and the training phase. Then, the RL-agents response to the environment by their actions via the installed controller on the vehicle, which adapts the vehicle speed, de-/acceleration to control the flow of traffic.

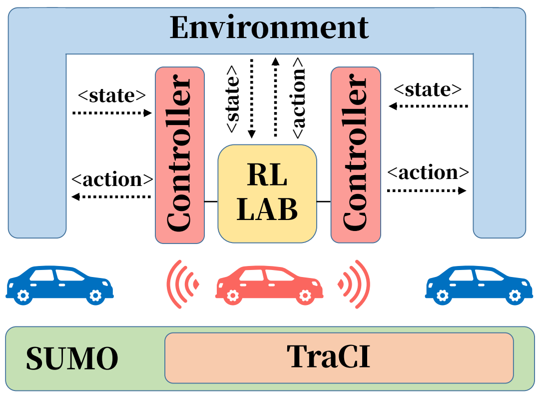


Fig. 2: Architecture of Flow, red car represents autonomous car with RL-agent; blue car represents the human-driven car

The major idea of the Flow is to deploy autonomous vehicles in the traffic flow, installed with RL-agents. And then, the RL-agents control the vehicle's behavior according to the environment. A simple single-lane ring road from Flow's example is represented, which is inspired by the 230m track studied by Sugiyama et al. [13]. Large shock-waves have been spotted in the ring road, when only human drivers and no autonomous vehicles are deployed, which causes large traffic congestion and very low traffic throughput. However, in the example experiment of flow, several autonomous vehicles are inserted into the human-driven traffic flow. Each autonomous vehicle has the abiliy to sense and collect the traffic data from the environment (vehicles from the behind and in the front). Thanks to the installed RL-agent, the data is fed into the trainer in RLlib [14], a Reinforcement Learning library, so that the autonomous vehicle can de-/increase their velocity to either slow down the vehicles behind them or to speed up the traffic flow by catching up the vehicle in the front. After a large number of trails and training, the autonomous vehicle will reach an optimized solution and produce the maximum traffic throughput. Thus, for this ring road example, the theoretical optimum is, that the vehicles are driving in the road with equivalent distance.

One of the examples provided by Flow is shown in Figure 3, where a eight-form intersection is presented, and 14 vehicles are driving to cross an intersection. The space-time diagrams are also shown in the Figure 3, and the improvement of applying RL-based autonomous vehicles can be achieved and the traffic can be further smoothed with increasing number of RL-based vehicles in the VANET.

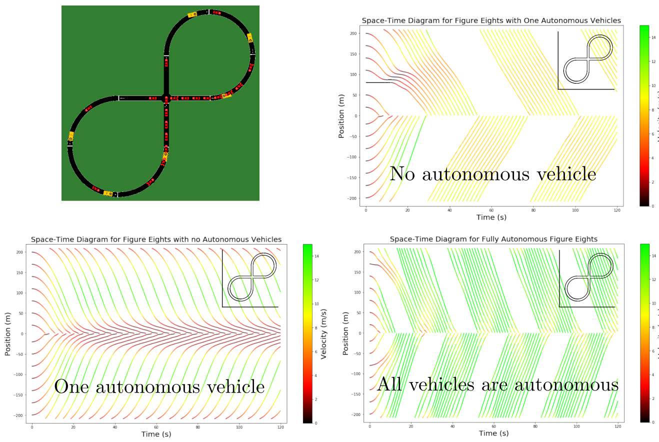


Fig. 3: Example eight-form intersection scenario from Flow, and space-time diagrams of deploying different number of RL-based autonomous vehicles [7]

B. NS3-Gym

ns3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. ns3 is free, open-source software, licensed under the GNU GPLv2 license, and maintained by a worldwide community.

ns3 aims to simulate a realistic networking environment and can be used as a real-time network emulator. ns3 also allows the user to interconnect with the real world through many existing real-world protocols, that are implemented in the community. Also, the ns3 has simple and extendable containers for users, such as the *Applications*, *Protocols* and *Network Devices*, where the user can define its own applications on demand, and existing real-world protocols like Transmission Control Protocol (TCP). Then, the network device can be installed on the simulating entities, connecting to the channel, as well as in real-world, which allocates the proper Medium Access Control (MAC) address, configuring the protocol stacks and parameters.

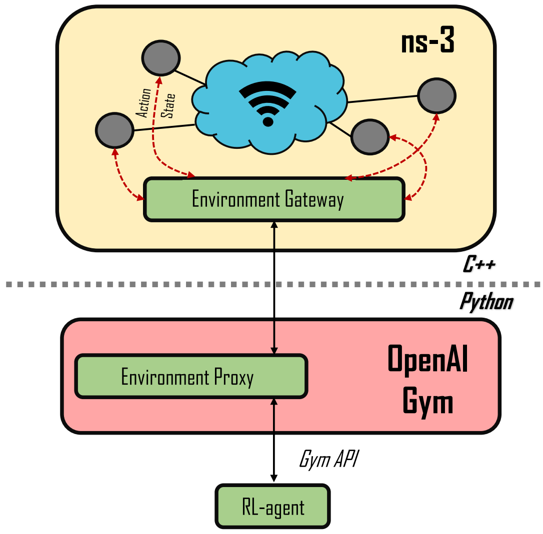


Fig. 4: Architecture of ns3-Gym

The system architecture of ns3-Gym is shown in Figure 4. In order to enable the ML/RL feature on the network device, ns3-Gym has developed an interface called *ns3-Gym Middleware*, which interconnects the ns3 network simulator and the OpenAI Gym platform. The major tasks of the middleware is to transfer the current state and control the RL-agent, i.e. the observation and actions, to the simulating environment. Thus, the middleware has two components, the *Environment Gateway* and *Environment Proxy*. The gateway is installed inside the network device and serves as the information collector from the environment state into the numerical data, and also the corresponding actions are translated. The proxy passes the environment state to the RL-agent via Gym API based on the Python.

An example scenario is provied by the ns3-Gym, where a problem of radio channel selection in a wireless multi-channel environment is considered. The example scenario is shown in Figure 5, and the goal is to select a free channel without interference in the coming time window.

The example scenario can be modeled easily by an RL-agent based on the proper observation for each channel and given time slot, and the model will be like:

- **Observation** - occupation on channel 1-4 and in every time slots
- **Actions** - choose the channel being occupied for the next coming time slot

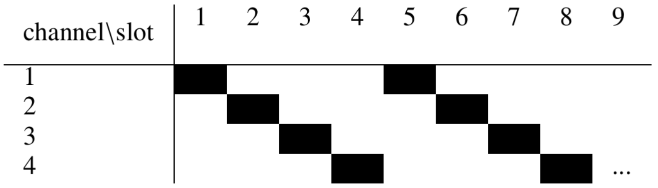


Fig. 5: Example scenario from ns3-Gym: Radio channel selection in a wireless multi-channel environment [12]

- **Reward** - gain +1, if no collision is detected due to interference, and otherwise gain -1
- **End** - More than three collisions are detected during the last 10 time slots

The results of the learning progress is shown in Figure 6, and after around 80 episodes, the next channel state can be well predicted by the RL-agent based on the current observation and hence avoid further collisions caused by the interference.

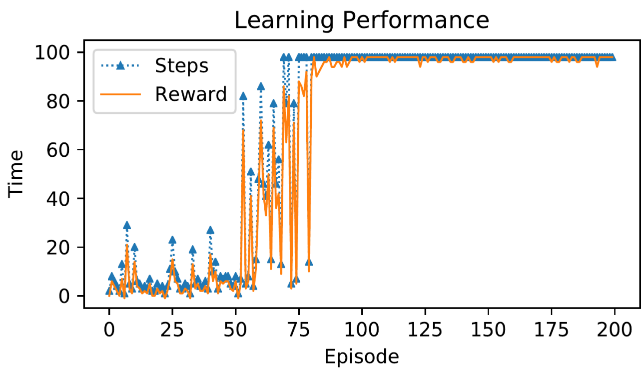
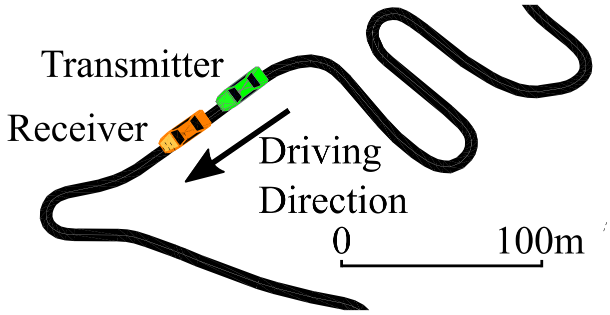


Fig. 6: Learning performance of the example scenario of radio channel selection [12]

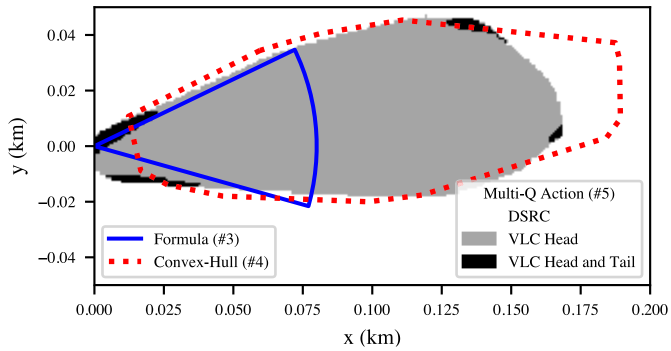
C. Veins-Gym

Veins-Gym is a simulation platform for breaking the gap between the vehicular network simulator *Veins* and the OpenAI Gym, which is similar to the ns3-Gym. The major difference to the ns3-Gym, is that the Veins-Gym includes the OMNet++-based vehicular network simulator Veins, which includes the modal mobility simulator SUMO, i.e. realistic traffic data can be collected. Thus, the applicability of the RL-agent is extended from the network layer to the vehicular traffic layer. In the Veins-Gym, an interface between Veins and OpenAI Gym is implemented, which manages the embedded Veins sequences and communicates with them. As the same, the Veins-Gym offers users a simple and extensible environment definition to the desired observation, action and reward spaces.

The Veins-Gym also provides an example scenario, where RL-agent is applied on the driving vehicles for selection of suitable communication link technology, and the scenario is shown in Figure 7



(a) Example of two vehicles driving along a serpentine road



(b) Selection of communication link technology

Fig. 7: Example scenario of Veins-Gym [8]

In the example scenario, two connected vehicles are driving on a serpentine road, modeled by the Lysevegen pass in southern Norway. The transmitter periodically sends a message to the receiver through the Visible Light Communication (VLC) channel, and the RL-agent installed on the transmitter selects one of the eight communication link technology, made from the combination of the Dedicated Short-Range Communication (DSRC) radio, the VLC headlight, and VLC taillight. The reward is gained, whenever a successful transmission is achieved, and reduced by the exceeding cost based on the chosen communication link channel.

During the learning progress, different polices have been implemented by Veins-Gym as follows:

- **DSRC only** (#1) - serve as the baseline policy, where only DSRC is applied
- **VLC Head only** (#2) - serve as the baseline policy, where only VLC Head is applied
- **Formula** (#3) - select between DSRC and VLC Head, based on the angle and the distance between the transmitter and the receiver
- **Convex-Hull** (#4) - 1000 pre-sampled observations of successful transmission with VLC Head is made before the training progress, and VLC Head is chosen if a transmission fails into the convex hull denoted by the

red dots in 7, and DSRC otherwise.

- **Multi-Q-Regressor** (#5) - an RL-based policy with 8 DNNRegressor estimators from Tensorflow framework. Selection is made among the above-mentioned communication link technologies.

The learning performance of different policies is shown in Figure 8.

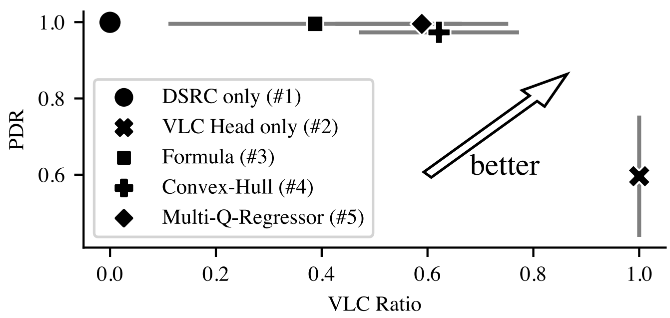


Fig. 8: The performance of the different policies of Veins-Gym example scenario. The Packet Deliver Ratio (PDR) is plotted against the ratio of packets sent via VLC, which indicates the resource usage. The grey bars are error bars, that indicate the standard deviation of the two metrics. [8]

D. SUMO-RL

SUMO-RL provides a simple interface to instantiate Reinforcement Learning environments with SUMO for Traffic Signal Control. The main class SumoEnvironment behaves like a MultiAgentEnv from RLlib [15]. If instantiated with parameter *'single-agent=True'*, it behaves like a regular Gym Env from OpenAI. Call *env* or *parallel_env* to instantiate a PettingZoo environment. Traffic Signal is responsible for retrieving information and actuating on traffic lights using TraCI API. The system-level architecture is shown in the Figure 9

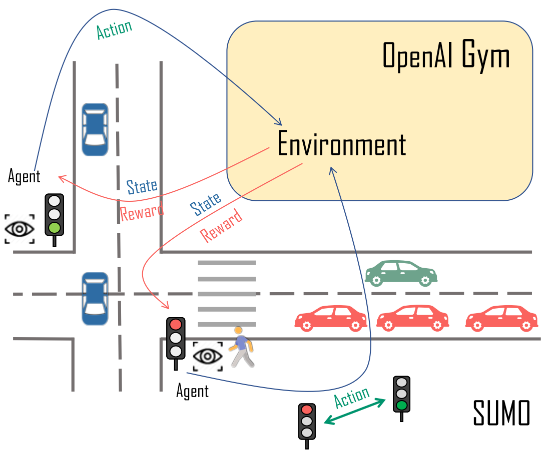


Fig. 9: Architecture of the SUMO-RL

In the related Project, namely the Reinforced Signal Control (RESCO), SUMO scenarios are extracted from the well-know SUMO traffic network project, "TAPAS Cologne" [16] and "InTAS" [17], based on the German cities of Cologne and Ingolstadt. And three benchmark control tasks are assigned to the two different scenarios:

- Single Signalized intersection
- Coordinated control over multiple intersections along an arterial corridor, around 500 m long
- Coordinated control over multiple intersections within a congested area, around 50 km^2

Also, comparisons are made among different conventional traffic signal control and the RL control algorithms:

Baseline controllers:

- **Fixed-time**: Common Traffic Lights (TLS), changing Red-Yellow-Green (RYG) phases with fixed time interval
- **Max-pressure**: Control the TLS phases according to the traffic pressure, designed in [18]
- **Greedy**: Control the TLS phases according to the queue length and approaching vehicle count, designed in [19]

RL controllers:

- **IDQN**: Assign each intersection with one independent Deep Q-Network (DQN)
- **IPPO**: Assign the same deep neural network as IDQN, except from the output layer from [20]
- **MPLight**: Implementation with FRAP [21], the ChainerRL DQN and the pressure sensing.
- **Extended MPLight**: Similar implementation to the MPLight, in addition to sensing information matching IDQN to the pressure state

The major implementation of the SUMO-RL focuses on the RL-agent on the traffic lights at the signalized intersections. The environment is specifically defined for the signalized intersections, including driving vehicles and phase-changing traffic lights. The observation space is made to each road lane controlled by the related traffic light, and the action is the choice of the discrete changing state of the signalized intersection. As for the reward, it is defined as the change in cumulative vehicle delays, and other different choices for the reward function can be defined by the user, specific to the problems.

IV. Comparison and Recommendation

In Table I, a comparison among different VANET simulators with OpenAI Gym is made with diverse factors. Based on the comparison, recommendations can be made to different research areas and topics of VANET simulations.

For those who would like to simulate large-scale traffic scenarios, especially simulating the whole city, *Flow* or *SUMO-RL* without any network communications is recommended since the VANET communication would results in severely computational complexity during the simulation, which is often not the main research subject of the macroscopic simulation. Related researches are the TAPASCologne project [22], where a large-scale traffic simulation of the city of Cologne, Germany is conducted. The simulation traces all vehicles for

TABLE I: Comparison among VANET simulators Recommendations

Simulator	Flow	ns3-Gym	Veins-Gym	SUMO-RL
Portability	✓	✓	✓	✓
Scalability	small	small	Large	small
GUI	×	✓	✓	×
Language	Python	C#/Python	C#/Python	Python
Documentation	✓	×	×	×
Win/Linux	× /✓	✓/✓	✓/✓	✓✓
Tutorials/Examples	✓	✓	✓	✓
Module design	✓	✓	✓	✓
VANET features				
IEEE 802.11p	×	✓	✓	×
LTE	×	✓	×	×
5G	×	×	×	×
OSI layers	×	✓	✓	×
Shadowing effect	×	✓	✓	×
Channel models	×	✓	✓	×
Obstacle models	×	✓	✓	×
Mobility				
Traffic management	SUMO	×	SUMO	SUMO
Machine Learning				
ML type	RL	RL	RL	RL
AI platform	Gym/rllib	Gym	Gym	Gym/rllib
Single agent	✓	✓	✓	✓
Multi agent	✓	✓	✓	✓

24 hours in the city of Cologne covering a 400 km^2 region, and more than 700.000 individual car trips are simulated.

For evaluating and simulating VANET communication networks, *ns3-gym* is to be recommended, since *ns3-gym* offers both WiFi-based and cellular-based V2X communication. However, the *ns3-gym* lacks the traffic mobility models, and thus, for investigating and deploying any Machine Learning and autonomous technologies on the VANET, *Veins-gym* is to be recommended. Although *Veins-gym* only offers the IEEE 802.11p wireless communication protocol, it is still a recommended choice to evaluate the joint performance of RL between the network level and the traffic level.

V. Conclusion

Since convincing results are needed for research VANET systems based on the ML/RL, a simulation for such vehicular communication should be more realistic to the real world, and also applied with the easy-to-use AI platform. In this work, various modern and advanced network simulators with the OpenAI Gym are introduced for simulating different research areas of VANET and ML/RL topics. The advantages and disadvantages of each simulator and their specialty for different topics are recommended.

It turns out that, in order to represent a realistic mobility simulation, most of the VANET simulators choose SUMO as their traffic simulator, which has been proved to be the best mobility simulation for VANET due to its simplicity, high efficiency, and flexibility.

NS3-Gym and Veins-Gym are recommended for simulating any VANET topics, which are related to WiFi-based or cellular networks since the completed OSI-layers are implemented for both simulators, which can easily be modified into new network protocols as wished. SUMO-RL and Flow are recommended for simulating any traffic management problem, specifically without any network implementation.

Yet, there are still a large number of VANET simulators available, but the simulators in the future should include more realistic models, more integration of Machine Learning and autonomous cars, easier installation and modification, more environment models to simulate under various weather effects, and smoother visualization for demonstration.

References

[1] "Technical specification group services and system aspects; release 14 description; summary of rel-14 work items," 3rd Generation Partnership Project, Standard, 2017.

[2] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular mobility simulation for vanets," in *40th Annual Simulation Symposium (ANSS'07)*, 2007, pp. 301–309.

[3] "Intelligent transport systems (its); vehicular communications; basic set of applications; part 2: Specification of cooperative awareness basic service," European Telecommunications Standards Institute, Standard, 2012.

[4] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[5] K. Jang, L. Beaver, B. Chalaki, B. Remer, E. Vinitsky, A. Malikopoulos, and A. Bayen, "Simulation to scaled city: zero-shot policy transfer for traffic control via autonomous vehicles," 12 2018.

[6] P. Gawlowicz and A. Zubow, "ns-3 meets openai gym: The playground for machine learning in networking research," 11 2019, pp. 113–120.

[7] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," 10 2017.

[8] M. Schettler, D. S. Buse, A. Zubow, and F. Dressler, "How to Train your ITS? Integrating Machine Learning with Vehicular Network Simulation," in *12th IEEE Vehicular Networking Conference (VNC 2020)*. Virtual Conference: IEEE, 12 2020.

[9] L. N. Alegre, "SUMO-RL," 2019.

[10] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[11] (2020). [Online]. Available: https://www.dlr.de/

[12] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research," in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, November 2019.

[13] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, "Traffic jams without bottlenecks - experimental evidence for the physical mechanism of the formation of a jam," *New Journal of Physics*, vol. 10, p. 33001, 03 2008.

[14] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "Rllib: Abstractions for distributed reinforcement learning," 2018.

[15] ——, "Rllib: Abstractions for distributed reinforcement learning," 2017.

[16] C. Varschen and P. Wagner, "Mikroskopische modellierung der personenverkehrsnachfrage auf basis von zeitverwendungstagebüchern," vol. 81, 01 2006, pp. 63–69.

[17] S. C. Lobo, S. Neumeier, E. M. G. Fernandez, and C. Facchi, "Intas – the ingolstadt traffic scenario for sumo," 2020.

[18] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Z. Li, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3414–3421, 04 2020.

[19] J. Ma and F. Wu, "Feudal multi-agent deep reinforcement learning for traffic signal control," in *AAMAS*, 2020.

[20] J. Ault, J. P. Hanna, and G. Sharon, "Learning an interpretable traffic signal control policy," in *AAMAS*, 2020.

[21] G. Zheng, Y. Xiong, X. Zang, J. Feng, H. Wei, H. Zhang, Y. Li, K. Xu, and Z. Li, "Learning phase competition for traffic signal control," 2019.

[22] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, 2014.