Emulating Emergency Vehicle Navigation in Mixed Traffic Environment Using Graph Prediction and Simulation

Shubham S[1], Amogh N Rao[1], Shuchith B U[1], Siddarth M P[1], and Bhaskarjyoti Das[2]

[1] Department of Computer Science and Engineering,
PES University, Bengaluru, Karnataka, India
[2] Department of Computer Science and Engineering in AI & ML,
PES University, Bengaluru, Karnataka, India
shubhamsbhatt@gmail.com , amogh090502@gmail.com ,
shuchithbu@gmail.com ,siddump2002@gmail.com ,
bhaskarjyoti01@gmail.com

Abstract. Emergency vehicles (EVs) must move through areas with mixed traffic as fast as they can in order to deliver aid in a timely manner. The dynamic and unpredictable nature of traffic makes this a difficult task. In some bottleneck situations, such as highway ramping and intersections, reinforcement learning (RL) has been used to simulate traffic and improve EV navigation. However, the complexity of real-world traffic environments is not fully captured by RL-based simulations, which are frequently restricted to a small number of entities. In this paper, our paper proposes a novel method that utilizes GraphSAGE and a custom action policy function to simulate EV navigation in mixed-traffic environments using SUMO simulation. A graph neural network (GNN) algorithm called GraphSAGE can be utilized to learn node representations, or the representations of the relationships between nodes in a graph, through training. The custom action policy function uses the link states that we predict using GraphSAGE to determine the EVs' next course of action and makes decisions about lane changes, speed adjustments, and acceleration based on the predicted link states, vehicle types, and relative positions of vehicles. We demonstrate that, when compared to baseline approaches, our approach can significantly improve the navigation efficiency of EVs. Our findings demonstrate how well graphs model and simulate intricate traffic situations. Our simulation can be used to investigate how various traffic situations and vehicle kinds affect EV navigation.

Keywords: GNN, GraphSAGE, mixed traffic environment, custom action, SUMO simulation, Dynamic Graphs

1 Introduction

The integration of autonomous vehicles (AVs) into our traffic networks is happening quite quickly. Although completely autonomous driving is still quite a long way away,

in the near future, traffic will be composed of both human-driven and autonomous vehicles. The unpredictable behavior of other drivers, the restricted sensor range, and the intricate traffic networks present a variety of difficulties for emergency vehicles (EVs) operating in this mixed-traffic environment. In situations where there is mixed traffic, EV navigation must be both safe and effective in order to quickly assist individuals in need. However, because traffic is dynamic and complex, this task is difficult.

Learning representations of nodes in a graph is possible with the use of machine learning models called graph neural networks (GNNs). The efficacy of GNNs has been demonstrated in numerous traffic network-related tasks, including congestion prediction and traffic forecasting [1,6,12,14,16,17]. As well as learning representations of EVs that capture their objectives and current state, GNNs can be used to model the traffic network's overall state. Equipped with this data, an action policy function that is comprehensive in nature can be developed to assist EVs in safely and efficiently navigating mixed traffic environments.

An innovative method that uses GNNs for efficient EV navigation in mixed-traffic environments has been proposed in this paper. Our method tackles the problem of sensor failures in real-life situations by utilizing GraphSAGE for link prediction. Additionally, we create a unique action policy function that improves upon the computational limitations of Reinforcement learning-based action systems [2]. Our function considers the EV's current condition as well as the types of vehicles in the network and the traffic network's overall state. We assess our suggested method on the generated environment and demonstrate that, in terms of both navigation efficiency and safety, it performs noticeably better than baseline methods.

2 Related Work

2.1 Simulation Environment

A diverse range of simulation environments has been employed to assess and validate autonomous driving systems in mixed-traffic environments. The most prevalent choice is the SUMO platform [1,2,3,4,9], which provides a realistic traffic simulation framework for modeling various road networks and the interaction between autonomous and human-driven vehicles. Additionally, the CARLA simulation [15] has been used to train and evaluate the DiGNet system, offering high-fidelity scenarios across complex maps, including urban, rural, and highway environments.

The robustness and generalizability of the suggested autonomous driving systems are greatly enhanced by the versatility and depth of these simulation platforms. However, rather than concentrating on particular use cases, the majority of research studies currently in existence assess the overall performance of autonomous driving systems. The lack of research in this area restricts the development and assessment of customized solutions for particular traffic situations, like emergency vehicle navigation.

2.2 Graph Representation and GNNs

Graph representation and the application of Graph Neural Networks (GNN) emerge as pivotal elements in enhancing decision-making for autonomous vehicles. Several papers leverage graph-based models to represent vehicle interactions and properties. In [1], every vehicle is depicted as a node, and interactions as edges in an undirected network, forming the basis for the proposed modular framework's state space. In [3], the Graphic Convolution Q network (GCQ) integrates Deep Q Network (DQN) and Graph Convolutional Neural Network (GCN) to facilitate cooperative lane change decisions, using graph-based techniques to aggregate collaborative sensing data. The adoption of GNN is further exemplified in [5,6,11,13,15], where graph structures capture spatial and temporal relationships among vehicles, enabling efficient decision-making in various cooperative and mixed-traffic scenarios. These approaches underscore the efficacy of graph-based representations and GNN architectures in modeling complex interactions and improving the decision-making capabilities of autonomous vehicles.

2.3 Action Policy and outcome

The proposed methodologies in the surveyed papers exhibit a variety of action policies designed to optimize autonomous vehicle decision-making. Reinforcement Learning (RL) algorithms [10], such as Deep Q-Learning [3,7,8], Curriculum through Self-Play [2], and Policy-based training methods [9,13], are recurrent themes. The utilization of advanced RL techniques, like Duelling Double DQN [1], Curriculum through Self-Play [2], and LSTM-Q [3], highlights the importance of effective learning strategies in achieving superior outcomes in interactive traffic scenarios. The incorporation of novel approaches, such as the Multi-Agent Reinforcement Learning (MARL) framework with Connected Automated Vehicle Graph [4] and the DiGNet system [15], showcases the diversity of methodologies aimed at improving efficiency, safety, and cooperation in autonomous driving.

However, lane changes are rarely taken into consideration, and the majority of current action spaces are restricted to simple movements like acceleration and deceleration. In addition, the number of vehicles that could be simulated was constrained by the complexity of RL algorithms. This limitation is addressed by our proposed custom action policy function, which allows complex maneuvers such as lane changes based on the current graph state. This enables us to model more realistic traffic scenarios and simulate a larger network.

3 Simulation Environment

Three simulation scenarios—highway merge shown in Fig.1, intersection shown in Fig. 2, and citywide scenarios which can be viewed in Fig. 3 are used to estimate the efficacy of our suggested method for emergency vehicle (EV) navigation in mixed traffic environments. All the scenarios are based on real-world traffic situations with human vehicles, autonomous vehicles, and emergency vehicles. For all scenarios, the complex traffic network is modeled using SUMO, a well-known and reliable simulation platform.

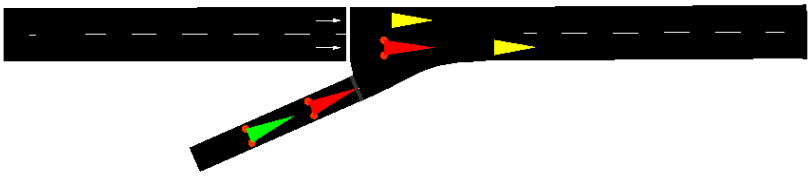

Fig. 1. Merge Network

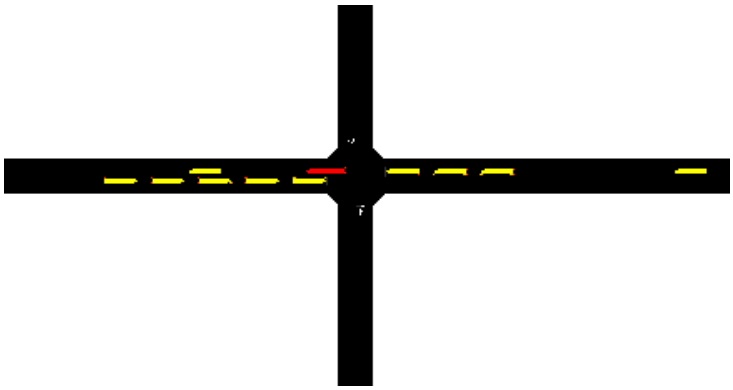

Fig. 2. Intersection Network

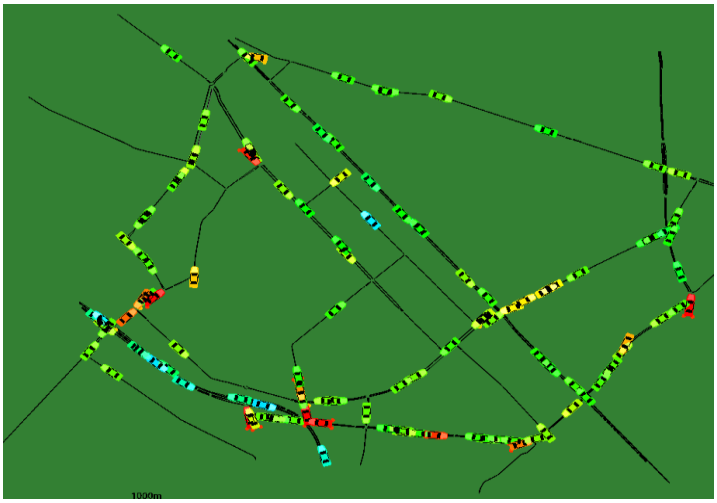

Fig. 3. City Network

The IDM3 algorithm, which accurately replicates human travel behaviour, is used to model human-driven vehicles. Our proposed action policy function is used by autonomous vehicles to receive actions and also applies the IDM3 algorithm. This combination ensures safe and effective navigation by enabling autonomous vehicles to make decisions according to the action policy function. Emergency vehicles are capable of moving through the traffic network faster because they are given priority over other vehicles. By setting priorities, emergency vehicles can get to their destinations faster and possibly save more lives.

Three metrics are considered in order to fully assess the effectiveness of our suggested approach: overall traffic waiting time, average speed, and navigation waiting time. The effectiveness of the navigation strategy is evaluated by measuring the amount of time an emergency vehicle waits while attempting to reach its destination, a measure known as navigation waiting time. Average speed determines the congestion level, which assesses how well the action policy function performs in averting collisions and guaranteeing smooth travel of traffic. The effect of the navigation strategy on the larger traffic system is measured by overall traffic waiting time, which is determined by the overall flow and congestion of the traffic network.

We compare two baseline approaches: Baseline 1 uses the IDM3 algorithm only and does not use any custom action policy, while Baseline 2 uses a custom action policy function that uses simulation obtained by the traCI library instead of GraphSAGE to predict links.

4 Methodology

The overall architecture of the simulation's interaction with the GraphSAGE algorithm is presented in Fig. 4. It can be essentially divided into three primary components: Graph representation, GraphSAGE and link prediction, and custom action policy function.

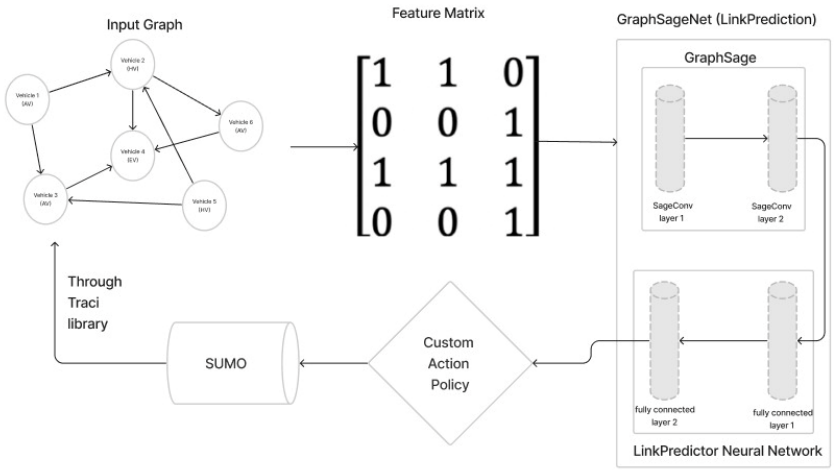

Fig. 4. Architecture

4.1 Graph Representation

At each timestep of the simulation, the vehicles present are identified and categorized as either HV i.e. Human Vehicle, AV i.e. Autonomous Vehicle, or EV i.e. Emergency Vehicles. The TraCI library is used to extract their attributes, such as position, speed, and lane information. This library makes communication with the SUMO traffic simulation software easier. Next, using the extracted vehicle data, a graph is built, with nodes representing the vehicles. On the basis of the position data of the two vehicles, an edge is established by calculating the distance between them using the formula mentioned in Eq. 1. An edge is added to the graph if this distance drops below a set threshold.

$$d = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2} \tag{1}$$

Edge formation is limited to interactions between AVs and HVs, EVs and HVs, or AVs and EVs in order to better represent real-world scenarios. Connections between HVs themselves are not included in this. This distinction arises from the fact that HVs are treated as passive actors in the simulation, with decision-making powers exclusively assigned to AVs and EVs. The graph better captures the dynamics of traffic scenarios involving AVs, EVs, and HVs by restricting edge formation to these interactions.

The graph representation consists of two components. One is the Node or Vehicle features matrix which is essentially made up of features list of individual vehicles in the environment.

$$V_T = [v_t^1, v_t^2, v_t^3, \ldots . v_t^i, \ldots \ldots v_t^n]^T \tag{2}$$

Here in Eq. 2. $v^i{}_T$ represents the feature matrix of vehicle v^i. The second component of the graph representation is edge list E. It is a two-dimensional array consisting of vehicle IDs as values.

$$E = \begin{bmatrix} v^i & \ldots\ldots & v^k \\ v^j & \ldots\ldots & v^l \end{bmatrix} \tag{3}$$

In Eq. 3. v^i and v^j are connected by an edge and similarly vehicles v^k and v^l also have an edge connecting them.

4.2 GraphSAGE and Link Prediction

When an emergency vehicle needs to get to its destination fast and effectively, its route requires to be optimized. This is known as emergency vehicle routing. The act of optimizing emergency response techniques can be further enhanced by link prediction, which forecasts the possibility of links or interactions between vehicles. Real-world applications of sensors include gathering vehicle location data, which is subsequently utilized to connect vehicles based on their proximity to one another. But malfunctioning sensors can cause gaps in data, which means there won't be any connections between

the cars. The decision-making process for emergency vehicles can be improved by using link prediction techniques to fill up these gaps by forecasting the missing edges.

To predict the missing links between vehicles, we employ a specialized model called GraphSageNet. This model is made up of two principal components:

GraphSAGE. This component creates the node embeddings for cars. In this instance, the traffic network's vehicles' attributes and interactions with one another are captured by the node embeddings produced by the GraphSAGE model.

LinkPredictorNN. This component forecasts whether or not there are links connecting the cars. For each pair of nodes, GraphSAGE produces node embeddings, which are then used to calculate a link prediction score. The higher the link prediction score, the more probable it is that there is a connection between the two nodes.

The GraphSAGE model comprises of two layers of SAGEConv, which is a strain of GraphSage graph neural network (GNN), that aggregates information from neighboring nodes to generate node representations.

$$G_T = \emptyset_{GraphSage}(V_t, E) \tag{4}$$

Here in Eq. 4. G_T refers to the node embeddings of vehicle nodes given by GraphSage module. The LinkPredictionNN(LNN) consists of two fully connected layers. The first layer takes the output-dimensional node embeddings from the GraphSAGE model as input and produces hidden-dimensional representations. The second layer takes the hidden-dimensional representations as input and produces a single output value, which represents the probability of a link existing between the two corresponding nodes.

$$P_T = \sigma_{LNN}(G_T) \tag{5}$$

And in Eq. 5. P_T refers to prediction scores given by the LinkPredictionNN module. In order to identify possible links, the LinkPredictionNN efficiently learns patterns in the informative node embeddings created by the GraphSAGE model, which captures the interactions between vehicles. Making use of the advantages of both the LinkPredictionNN and the GraphSAGE model, the combined GraphSageNet model predicts the missing links between vehicles with high accuracy.

The GraphSageNet model is trained using a dataset obtained from the SUMO (Simulation of Urban MObility) software, utilizing the TraCI (Traffic Control Interface) API. The dataset represents a traffic network extracted from OpenStreetMap (OSM) data. A snapshot of the traffic network taken at a particular moment in time is used to train the model. The training graph's edges are constructed using the same process as described in the section on graph representation. GraphSageNet is used in the actual simulation process to forecast the presence of any potential missing links after the traffic graph has been constructed. Then, in the simulated environment, the Custom Action policy is applied to make well-informed decisions for both autonomous and emergency vehicles.

4.3 Custom Action Policy

The main element in charge of choosing actions based on the traffic graph's current state is the custom action policy function. The presence of edges in the graph is necessary for it to function. The action policy function is triggered to determine a score for every vehicle when there are sufficient edges in the graph. Subject to the type of vehicle and the characteristics of its surrounding nodes and edges, this score is the basis for decision-making.

For emergency vehicles (EVs), a high score is assigned, prompting an adjustment of their speed towards their maximum allowable limit. This guarantees that electric vehicles (EVs) can quickly and effectively navigate the traffic network to get to their destinations.

The action policy function for autonomous vehicles (AVs) uses neighborhood data to decide what speed modifications are appropriate. The action policy function tells an AV to switch lanes to make room for an EV if they are in the same lane. When the autonomous vehicle (AV) is in a separate lane, it is advised to reduce speed to prevent possible collisions.

The custom action policy function uses the vast amount of information found in the traffic graph to plan an orderly and effective vehicle movement that guarantees emergency vehicles can navigate through mixed traffic situations in a timely and safe manner.

5 Results and Analysis

Unlike existing research which uses Reinforcement Learning to select actions, our method does not need the high processing power of GPU hardware to model the network. Extensive computations are not required as our custom action policy function generates concise actions considering the traffic graph's current state.

We examined three distinct cases for each of the three simulation environments in order to assess the efficacy of our methodology. Using no algorithmic intervention, the first case simulates a scenario and acts as a baseline. The second example builds the graph with our custom action policy function applied, utilizing simulation attributes which are taken from SUMO. In the final scenario, link states are inferred using GraphSAGE, and our custom action policy function is employed based on the resulting graph. The experimental parameters for the three scenarios merge, intersection, and city network scenarios are presented in Table. 1.

Table 1. Parameter Settings

Parameters	Scenarios		
	Merge	***Intersection***	***City***
Number of HVs	23	26	47
Number of AVs	14	16	26
Number of EVs	3	2	7
Speed limit of EVs and AVs	120 km/h	120 km/h	100 km/h
Speed limit of HVs	80 km/h	80 km/h	70 km/h
Number of Lanes	2	2	<3

5.1 Merge

The relatively sparse distribution of cars in the merge scenario suggests that individual vehicle driving behavior has little effect on the overall flow of traffic. This finding seemingly clarifies the performance patterns shown in Table. 2. Even though the improvement in performance over the baseline is not very large, it is still noteworthy. Furthermore, the GraphSAGE-based approach's performance is close to that of the condition where the graph was created with simulation attributes.

Table 2. Performance on Merge

Simulation Scenario	Parameters		
	Emergency Waiting Time	***Average Speed***	***Average Waiting Time***
Baseline	12.2s	67.2km/h	14.8s
Graph made by simulation attributes	8.5s	88.5km/h	9.2s
Graph made by GraphSAGE	6.1s	82.4km/h	5.1s

5.2 Intersection

The optimization effect attained in the intersection scenario was more visible than that seen in the merge scenario, as clearly seen in Table. 3. This discrepancy is probably due to the increased mutual influence between cars in the intersection situation, where their movements and interactions have a bigger impact on the direction of traffic flow as a whole. Interestingly, the simulation-based approach's performance and the GraphSAGE-based approach's closely match which highlights how well GraphSAGE captures the complex dynamics of the intersection scenario.

Table 3. Performance on Intersection

Simulation Scenario	**Parameters**		
	Emergency Waiting Time	***Average Speed***	***Average Waiting Time***
Baseline	12.3s	55.8km/h	15.8s
Graph made by simulation attributes	3.6s	83.1km/h	3.2s
Graph made by GraphSAGE	3.8s	87.2km/h	3.1s

5.3 City Network

As can be seen from Table. 4, the custom action policy function significantly outperforms the baseline when combined with the graph for the city network. Furthermore, the simulation-based approach and the GraphSAGE-based approach for graph construction closely match, indicating GraphSAGE's strong prediction abilities. Due to computational constraints, large-scale traffic simulations are just not possible, so this is a significant improvement over RL-based action systems.

Table 4. Performance on City Network

Simulation Scenario	**Parameters**		
	Emergency Waiting Time	***Average Speed***	***Average Waiting Time***
Baseline	12.2s	54.7km/h	20.8s
Graph made by simulation attributes	4.8s	72.9km/h	8.1s
Graph made by GraphSAGE	4.9s	78.3km/h	9.6s

The outcomes show that the utilization of a graph in conjunction with a custom action policy function has greatly increased the simulation's efficiency. Moreover, GraphSAGE is an effective way for predicting links for graphs in such forms of traffic networks and can be used efficiently in these kinds of situations. The train-test curve for GraphSAGE, as depicted in Fig 5. illustrates the loss and highlights its low value.

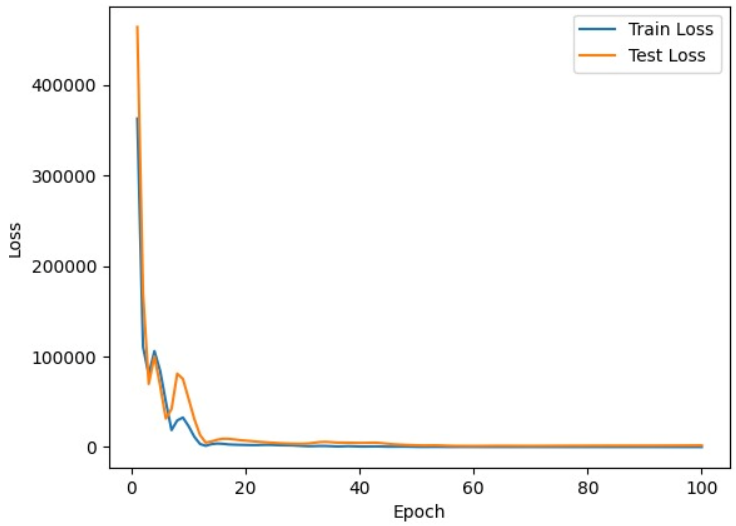

Fig. 5. Train Test curve for GraphSAGE

6 Conclusions and Future Scope

This paper utilizes GraphSAGE and a custom action policy function to present a novel system for EV navigation in mixed-traffic environments. By capturing the dynamic and unpredictable nature of traffic, our method performs better than baseline approaches in enabling EVs to navigate. EV navigation efficiency has been substantially improved by integrating GraphSAGE for graph representation and link prediction with a custom action policy function for EV navigation optimization. This reveals the potential of graph-based traffic modelling and prediction methods in practical traffic control situations, especially for vital applications such as emergency vehicle navigation.

The suggested method has a number of benefits over conventional RL-based techniques. GraphSAGE enables more precise link prediction and EV navigation by efficiently capturing the intricate spatial relationships present in the traffic network. Second, these relationships are accounted for by the custom action policy function, which optimises traffic flow for EVs and results in more effective navigation. Third, the SUMO simulation framework's integration of GraphSAGE and the custom action policy function enables large-scale traffic simulations, getting around the computational constraints of RL-based techniques.

On the whole, our paper shows the potential of graph-based traffic modeling and prediction methods in practical traffic management applications, especially for crucial situations such as electric vehicle (EV) navigation. Our suggested method provides a viable means of enhancing the effectiveness of EV navigation and guaranteeing prompt assistance delivery during emergencies. Future research will examine how various graph neural network architectures affect our method's effectiveness, create a more complex action policy function that considers other variables like pedestrians and road conditions, and test our method on a bigger and more varied dataset of traffic scenarios. These efforts will improve the efficacy of our suggested methodology and expand its suitability to an expanded array of actual traffic situations.

References

1. Liu, Qi, et al. "Graph Convolution-Based Deep Reinforcement Learning for Multi-Agent Decision-Making in Mixed Traffic Environments." arXiv preprint arXiv:2201.12776 (2022).
2. G. -P. Antonio and C. Maria-Dolores, "Multi-Agent Deep Reinforcement Learning to Manage Connected Autonomous Vehicles at Tomorrow's Intersections," in IEEE Transactions on Vehicular Technology, vol. 71, no. 7, pp. 7033-7043, July 2022, doi: 10.1109/TVT.2022.3169907.
3. Dong, Jiqian, Sikai Chen, Paul Young Joun Ha, Yujie Li, and Samuel Labi. "A drl-based multiagent cooperative control framework for cav networks: a graphic convolution q network." arXiv preprint arXiv:2010.05437 (2020).
4. T. Shi, J. Wang, Y. Wu, L. Miranda-Moreno, and L. Sun, "Efficient Connected and Automated Driving System with Multi-agent Graph Reinforcement Learning," IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2021.3067931, 2021.
5. Klimke, Marvin, Benjamin Völz, and Michael Buchholz. "Automatic Intersection Management in Mixed Traffic Using Reinforcement Learning and Graph Neural Networks." arXiv preprint arXiv:2301.12717 (2023).
6. Klimke, Marvin, Benjamin Völz, and Michael Buchholz. "Cooperative Behavior Planning for Automated Driving using Graph Neural Networks." In 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 167-174. IEEE, 2022.
7. Klimke, Marvin, et al. "An Enhanced Graph Representation for Machine Learning Based Automatic Intersection Management." 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2022.
8. J. Dong et al., "Spatio-weighted information fusion and DRL-based control for connected autonomous vehicles," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-6, doi: 10.1109/ITSC45102.2020.9294550.
9. Gao, Xin, et al. "Multi-agent decision-making modes in uncertain interactive traffic scenarios via graph convolution-based deep reinforcement learning." Sensors 22.12 (2022): 4586.
10. Liu, Qi, Xueyuan Li, Yujie Tang, Xin Gao, Fan Yang, and Zirui Li. "Graph Reinforcement Learning-Based Decision-Making Technology for Connected and Autonomous Vehicles: Framework, Review, and Future Trends." Sensors 23, no. 19 (2023): 8229.
11. Lu, Yuhuan, Wei Wang, Xiping Hu, Pengpeng Xu, Shengwei Zhou, and Ming Cai. "Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks." IEEE Transactions on Intelligent Transportation Systems (2022).

12. Liu, Tao, Aimin Jiang, Jia Zhou, Min Li, and Hon Keung Kwan. "GraphSAGE-Based Dynamic Spatial–Temporal Graph Convolutional Network for Traffic Prediction." IEEE Transactions on Intelligent Transportation Systems (2023).
13. Shang, Yigeng, Zhigang Hao, Chao Yao, and Guoliang Li. "Improving Graph Neural Network Models in Link Prediction Task via A Policy-Based Training Method." Applied Sciences 13, no. 1 (2022): 297.
14. Rathore, Heena, and Henry Griffith. "GNN-RL: Dynamic Reward Mechanism for Connected Vehicle Security using Graph Neural Networks and Reinforcement Learning." In 2023 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 201-203. IEEE, 2023.
15. Cai, Peide, Hengli Wang, Yuxiang Sun, and Ming Liu. "DiGNet: Learning scalable self-driving policies for generic traffic scenarios with graph neural networks." In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 8979-8984. IEEE, 2021.
16. Yu, Zishun, and Mengqi Hu. "Deep reinforcement learning with graph representation for vehicle repositioning." IEEE Transactions on Intelligent Transportation Systems 23, no. 8 (2021): 13094-13107.
17. Liu, Jielun, Ghim Ping Ong, and Xiqun Chen. "GraphSAGE-based traffic speed forecasting for segment network with sparse data." IEEE Transactions on Intelligent Transportation Systems 23, no. 3 (2020): 1755-1766.