

PAPER • OPEN ACCESS

GCN-SATO: A Graph Convolutional Network with Self-Attention based Car-Following Model

To cite this article: Qiran Li *et al* 2024 *J. Phys.: Conf. Ser.* **2832** 012017

View the [article online](#) for updates and enhancements.

You may also like

- [Modeling and analysis of car-following behavior considering backward-looking effect](#)
Dongfang Ma, , Yueyi Han et al.
- [CO₂ emission control in new CM car-following model with feedback control of the optimal estimation of velocity difference under V2X environment](#)
Guang-Han Peng, , Rui Tang et al.
- [Modeling the heterogeneous traffic flow considering the effect of self-stabilizing and autonomous vehicles](#)
Yuan Gong, , Wen-Xing Zhu et al.

GCN-SATO: A Graph Convolutional Network with Self-Attention based Car-Following Model

Qiran Li¹, Qian Chen², Chengli Jian¹, Qingnan Wang³, Jihui Tu^{1,*}

¹School of Electronic Information and Electrical Engineering, Yangtze University, Jingzhou, 434023, PR China

²Department of Traffic Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, PR China

³School of Physics, Electronics and Intelligent Manufacturing, Huaihua University, Huaihua, 418000, PR China

*Corresponding author's e-mail: tujh@yangtzeu.edu.cn

Abstract. Car following is critical for the overall safety, efficiency, and smooth operation of autonomous vehicles in traffic. However, the existing car-following model primarily focuses on local feature extraction, overlooking the spatial-temporal relationships between vehicles and key information within the time series. This limitation negatively impacts prediction accuracy and generalization capabilities. To tackle these problems, this paper proposes a novel car-following method based on a Graph Convolutional Network (GCN) and a self-attention mechanism. Firstly, the GCN network is utilized to construct the spatial-temporal structure of the car-following model, which can effectively obtain topological structure relationships among vehicles. Secondly, the self-attention module is designed to assign different attention weights to different neighbors of a node in the car-following model, which can facilitate the capture of various aspects of node relationships simultaneously and enhance its expressive power. Finally, the Multi-Layer Perceptron (MLP) layer is employed to predict the future behaviors of following vehicles. The proposed car-following model (CF) was trained and evaluated using five real-world datasets: HighD, SPMD, Waymo, NGSIM, and Lyft. The results indicate that our method outperforms other existing methods in Mean Squared Error (MSE) of spacing and achieves a zero collision rate on all datasets. These findings indicate that the GCN-SATO proposed in this paper enhances safety and smooth operation compared to other existing models.

Keywords: Car-following, Traffic Simulation, Traffic Flow, Data-driven approach.

1. Introduction

Autonomous driving describes a technology that permits a vehicle to run without requiring direct input from a human driver. This technology holds promising potential, including reducing traffic accidents caused by human error[1], improving traffic flow and efficiency, and providing mobility to individuals unable to drive. The car following model significantly contributes to creating autonomous driving systems that are more efficient, safer, and more comfortable to use. Therefore, the car following model plays a critical role in the development of autonomous driving systems. With the rapid development of artificial intelligence, the automatic vehicle following model, which simulates human driving behavior with the help of intelligent technology, has become a research hotspot.

Over the past seven decades, the car-following model has been studied and discussed by researchers, which was first proposed by Pips in 1953[2]. From then on, A significant



variety of car-following models were introduced and categorized into two primary groups: mathematical driven models and data-driven models. Firstly, mathematical car-following models mathematically describe driving behavior in different traffic conditions. The Optimal velocity model (OVM)[3] was introduced based on the intuitive changes in car speed, the Gipps model[4] applied the notion of a safe distance, the intelligent driver model (IDM)[5] represents a driver-based desired goal method, and the Gazis-Herman-Rothery model (GHR)[6] is recognized as the initial stimulus-response model. Some researchers stated that the mathematical model-driven approach has a relatively safe performance, while it has a lower accuracy rate. Secondly, data-driven car-following models accurately simulate driving behavior based on data without manual intervention. Toledo[7] stated that driver's behaviors always depend on their historical driving behaviors and the departed traffic flow state, and other researchers[8, 9] also pointed out that CF models depending on the historical driving behaviors can describe the traffic flow characteristics better. The LSTM-NN model[10] can effectively catch the asymmetric driving behavior due to its advantage on taking historical information into account. The RNN model[11], and the Gate Recurrent Unit (GRU) NN model[12] including 10 second historical driving behavior, outperform IDM. In addition, reinforcement learning (RL) was also intensively applied in car-following behavior modeling[13, 14]. Utilizing deep deterministic policy gradient (DDPG) model[14] results in less collision and lower MSE of spacing. Data-driven car-following models could extract more important details from realistic traffic flow and accurately predict the future following vehicles' behaviors, compared to mathematical driven models.

The above methods have achieved satisfactory performance, but the following challenges still need to be addressed: (1) The spatial-temporal structure of the car-following model. This structure describes the behavior of vehicles over time and space as they follow one another. The model primarily considers three dimensions: time, the spatial position of the vehicle, and the speed of the vehicle. These dimensions together form the decision-making mechanism for the following vehicles, which can ensure optimal conditions for driver behavior in terms of safety and efficiency. However, current methods pay little attention to the spatio-temporal characteristics of vehicle following, which can impact the accuracy of predictions. (2) Dynamic relationship between vehicles. The dynamic relationship between vehicles in a car-following model involves a complex interaction of multiple factors, which is essential for autonomous vehicles to make secure and effective decisions depending on their surroundings. The topology relationship of a car-following model can be represented as a dynamic graph structure, with its size and structure constantly changing over time. The importance of nodes and their neighbors in the graph structure also varies. If this problem is ignored, it will make the model difficult to adapt to different topologies, which will affect the robustness and generalization ability of the model. To tackle this issue, we introduced a novel car-following model based GCN and self-attention mechanism, which is named GCN-SATO. The proposed method was trained and tested on five real-world datasets(HighD, SPMD, Waymo, NGSIM, and Lyft.), and it has achieved state-of-the-art performance compared to current existing models. The experimental results demonstrate that our method exhibits strong robustness and generalization capabilities, which provides a more comfortable and safe driving behavior for autonomous driving.

In short, the novelty and significant contributions of the study are as follows:

- A novel method that utilizes Graph Convolutional Networks (GCN) and self-attention is proposed for the car-following model. The GCN network is utilized to build the spatial-temporal structure within this model, which is aim to improve the prediction accuracy. This structure is specifically designed to create a realistic flow of traffic, which can enhance the efficiency, safety, and comfort of driver behavior.
- The self-attention mechanism is introduced into our model to handle the dynamic graph of the car-following model with varying sizes and structures. This mechanism assigns different attention weights to different neighbors of a node, which makes it adaptable to

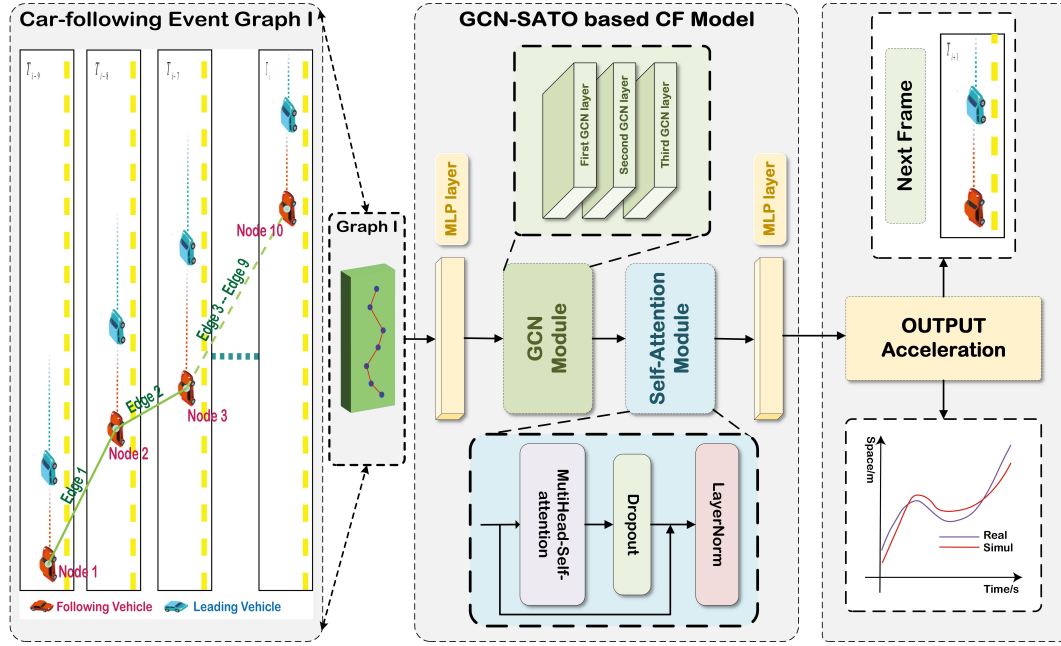


Figure 1: System Architecture

different graph topologies. The self-attention mechanism can improve the robustness and generalization ability of the proposed model.

- We conducted a comparison of our method with other state-of-the-art methods using five public datasets (HighD, SPMD, Waymo, NGSIM, and Lyft.). The experimental results revealed that our method outperforms other existing models in MSE of spacing, TTC, Jerk and achieves a zero collision rate in all datasets.

The rest of this paper is structured as follows: Section 2 outlines the proposed method; the details on datasets, evaluation criterion, and experimental results are provided in Section 3; the conclusion are presented in Section 4.

2. Methodology

In this section, the car-following based GCN-SATO model is proposed, which consists of four parts: the input module, the graph convolutional network module, the self-attention mechanism module, and the output module. For the purpose of obtaining better spatial-temporal feature from data, as shown in Fig.1, the historical n time series data is initially transformed into a graphical representation, serving as the input for the network. First, as the input module, the Multi-Layer Perceptron (MLP) serves to encode and transform the data, enhancing its comprehensibility and facilitating processing by subsequent layers within the network. Second, the processed data is fed into the graph convolutional network module, enabling model to catch effectively topological structure relationships among vehicles. Third, the obtained data with temporal-spatial features are input into the self-attention module to assign different attention weights to different neighbors of a node. Finally, we get the predicted vehicle acceleration through the Multi-Layer Perceptron (MLP).

2.1. Problem Definition

Definition1: car-following event network G_i . An unweighted graph $G_i = (V_i, E)$ is employed to represent the topological structure of the car-following event network, where each node corresponds to one frame of historical information, V_i is a set of frame historical information, $V_i = \{v_{i-9}, v_{i-8}, v_{i-7}, \dots, v_i\}$, with the total number of the nodes being 10. The set E represents

the edges between these nodes. The adjacency matrix A is employed to depict the connection among F frame historical information, $A \in R^{F \times F}$, where F is 10 in this paper.

Definition2: feature matrix $X^{F \times 3}$. In a car-following event network, the 10 frames of historical information are considered the attribute features of the nodes. This is expressed as $X \in R^{F \times 3}$, where F denotes the amount of node attribute features (the length of the historical time series). The matrix X_i represent the historical information at time i .

Thus, the car-following problem can be viewed as learning a map function f derived from the car-following event network G and feature matrix X , with the objective of forecasting vehicle acceleration for the subsequent T moments, as illustrated in eq.1:

$$Acc_i = f(G_i; X_i), i \in (F, Total_Event - F) \quad (1)$$

where $Total_Event$ is the number of historical time series in one car-following event.

2.2. Graph Convolutional Network module

Currently, the Long Short-Term Memory[15] stands as the most prevalent neural network model for handling sequence data, renowned for its capability in long-term prediction. Additionally, in [16], both Long Short-Term Memory and Gated Recurrent Unit architectures were employed to model car-following behavior, contributing to enhanced performance of the car-following model. In [17], an unidirectional interconnected LSTM-based car-following model (Int-LSTM) addressed the issue of error propagation. However, these existing car-following model usually focus on temporal relationships between vehicles, overlooking spatial relationships.

In this research, we propose a GCN Module based the graph convolutional network[18] to acquire the spatial-temporal dependence. Given an adjacency matrix E and a feature matrix X , the GCN model creates a filter in the Fourier domain. This filter captures spatial-temporal characteristics between nodes through their immediate neighborhoods. The GCN module is then constructed through the stacking multiple convolutional layers, as shown in eq.2:

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \hat{E} \tilde{D}^{-\frac{1}{2}} H^{(l)} \theta^{(l)}) \quad (2)$$

where $\hat{E} = E + I_N$, I_N is the identity matrix, $\tilde{D} = \sum_j \tilde{A}_{ij}$, $\tilde{D}$ is the degree matrix, $H^{(l)}$ is the output features of l layer, $\theta^{(l)}$ embraces the weighting parameters of that layer, and $\sigma()$ is the sigmoid function.

The multiple GCN layers is utilized to capture temporal dependence, as illustrated in eq.3:

$$f(X, E) = \sigma(\hat{E} ReLU(\hat{E} ReLU(\hat{E} X W_0) W_1) W_2) \quad (3)$$

where $\hat{E} = \tilde{D}^{-\frac{1}{2}} \tilde{E} \tilde{D}^{-\frac{1}{2}}$ denotes pre-processing step. $W_0 \in R^{P \times M}$, $W_1 \in R^{M \times H}$ and $W_2 \in R^{H \times T}$ is the weight matrix between hidden layers relatively.

2.3. Self-Attention module

Acquiring the dynamic relationship between vehicles is a crucial problem in car-following model. The convolutional neural network (CNN) is capable of extracting local spatial features, but it is challenging to concentrate on crucial regions within a broader context or larger set of information. However, the Attention mechanism[19] addressed the problem mentioned above, which has been widely adopted in many fields.

Let $F \in R^{N \times T}$ be the output embedding features of GCN, which also the input feature for the self-attention module, and it outputs $G \in R^{N \times T}$ embedding features. Let $f_j \in R^T$,

$g_j \in R^{T_{out}}$ represent the corresponding tensor of node (j). Then, the output of the Muti-Head Self-attention layer is expressed as:

$$g_j^{(l)} = \parallel_{l=1}^N \left(\sum_{a \in N_k(j)} A(W_q^{(l)} f_j, W_k^{(l)} f_a) W_v^{(l)} f_a \right) \quad (4)$$

where N denotes N attention heads, and $W_q^{(l)}$, $W_k^{(l)}$, $W_v^{(l)}$ are the projection matrices for queries, keys and values in Muti-Head Self-attention layer l , respectively. $N_k(j)$ is a local region of nodes with spatial extent k centered around (j), and $A(W_q^{(l)} f_j, W_k^{(l)} f_a)$ is the attention weight related to the features within $N_k(j)$.

The attention weights are expressed as:

$$A(W_q^{(l)} f_j, W_k^{(l)} f_a) = \underset{N_k(j)}{softmax} \left(\frac{(W_q^{(l)} f_j)^T (W_k^{(l)} f_a)}{\sqrt{d}} \right) \quad (5)$$

where d represents the feature dimension of $W_q^{(l)} f_j$.

2.4. Loss Function

During the model training, the aim of proposed model is to narrow the gap between the actual values and the simulated values by iteratively updating the model parameters through loss calculation. The loss function of the GSTAM-CF based CF model is expressed as:

$$loss = (Acc_{n-1,n}(t) - Acc_{n-1,n}^{obs}(t))^2 \quad (6)$$

where $Acc_{n-1,n}(t)$ and $Acc_{n-1,n}^{obs}(t)$ are the predicted and objective acceleration at time step t , respectively.

3. Experiments and Evaluations

3.1. Dataset

In this section, our model is trianed and inferenced using five real-world driving datasets: HighD, SPMD, Waymo, NGSIM and Lyft dataset. We extracted car-following events from these datasets as described in [14]. A total of 1930, 24247, 1440, 24093, and 12540 car-following events were extracted from these datasets, respectively. In particular, the time interval of data sampling for HighD dataset is 0.04 second, and for other datasets is 0.1 second.

3.2. Evaluation Criterion

In a real autonomous driving environment, comfort, accuracy and safety play a crucial role for car-following model, thus we utilized mean square error of spacing($mse_{spacing}$), collision rate, Jerk and TTC to evaluate the prediction per-formance of the car-following based GCN-SATO model.

For each car-following event, the MSE of spacing, collision rate, Jerk and TTC can be computed as:

$$MSE = \frac{1}{N} \sum_{i=1}^N (S_{n-1,n}(t) - S_{n-1,n}^{obs}(t))^2, \text{ Collision Rate} = \frac{N_{collision_event}}{N} \quad (7)$$

$$Jerk = \frac{a_n(t) - a_n(t-1)}{dt}, \text{ TTC}(t) = -\frac{S_{n-1,n}(t)}{\Delta V_{n-1,n}(t)} \quad (8)$$

where N represents the overall amount of objectives, i is an objective index, $N_{collision_event}$ describes the number of collisions that occurred in car following, and $S_{n-1,n}(t)$, $S_{n-1,n}^{obs}(t)$ represent the predicted and objective spacing value at time step t , respectively. $a_n(t)$ and $a_n(t-1)$ denote the acceleration of the following vehicle at time t and $t-1$ respectively, and $\Delta V_{n-1,n}(t)$ illustrates the relative velocity between the front and rear cars at time t .

Table 1: Model Performance: MSE of spacing and Collision rate

	MSE of spacing					Collision rate(%)				
	HighD	SPMD	Waymo	NGSIM	Lyft	HighD	SPMD	Waymo	NGSIM	Lyft
GHR[6]	23.76	119.92	45.52	62.18	178.01	0	0	0	0	0
IDM[5]	27.50	138.69	39.17	54.97	104.32	0	0	0	0	0
NN[14]	24.06	24.14	16.79	29.00	26.06	39.87	23.92	46.29	3.46	0
LSTM[14]	22.90	22.49	13.75	23.86	25.71	3.72	15.39	4.62	17.30	0
DDPG[14]	42.29	89.34	33.91	46.93	86.98	0	0	0	0	0
GCN-SATO	19.98	19.34	11.84	14.15	25.68	0	0	0	0	0

Table 2: Model Performance: Jerk and TTC

	Jerk					TTC				
	HighD	SPMD	Waymo	NGSIM	Lyft	HighD	SPMD	Waymo	NGSIM	Lyft
GHR[6]	0.34	0.13	0.42	0.53	0.24	101.38	85.53	61.10	14.03	78.70
IDM[5]	0.09	0.19	0.22	0.27	0.29	184.80	177.27	40.63	31.10	73.71
NN[14]	0.08	0.08	0.32	0.75	0.15	60.13	75.54	35.67	13.34	121.78
LSTM[14]	0.04	0.08	0.24	0.25	0.14	127.68	119.77	104.37	13.74	132.47
DDPG[14]	33.4	23.84	7.23	8.04	31.42	111.66	32.98	152.94	12.18	46.47
GCN-SATO	0.01	0.01	0.02	0.04	0.02	218.11	237.19	168.30	102.61	186.49

Table 3: Ablation experiment: MSE of spacing

	HighD	SPMD	Waymo	NGSIM	Lyft
GCN	22.7525	23.1186	11.8644	15.2457	39.6877
GCN-SATE	21.8089	20.9007	14.6378	14.6980	27.3005
GCN-SATO	19.9832	19.3352	11.8426	14.1470	25.6809

3.3. Performance Evaluation

3.3.1. MSE of spacing MSE of spacing refers to the variance between the predicted vehicle spacing and the target spacing, and reflects the system's accuracy in controlling vehicle movement. Table.1 shows a comparison of our methods with several existing algorithms (NN, LSTM and DDPG[14]) in terms of mean squared error (MSE) of spacing. LSTM[14] with forget gates is not only capable of extracting more important knowledge from the historical information than NN, but also forgets unimportant memory gradually when the historical information becomes outdated. Therefore, from Table.1, it is not difficult to find that LSTM has more excellent performance than NN in MSE of spacing in all five datasets. However, It's not difficult to find that our method (GCN-SATO) outperforms other existing methods in all five datasets. In addition, DDPG[14] based CF model has higher value in $mse_{spacing}$ than other all above algorithms in all datasets.

3.3.2. Collision rate Safety is an essential metric in autonomous driving, where a single collision can cause vehicle accidents or death, thus collision rate is an essential evaluation metric to test the performance of car-following model. From Table.1, compared with NN and LSTM, our proposed GCN-SATO model reaches zero collision rate in all datasets, which means that the network structure of our method can obtain more important knowledge from the historical information. In [14], Chen stated that the drivers in Lyft dataset have better driving behaviors than others, which can help these car-following models improve safety performance during the

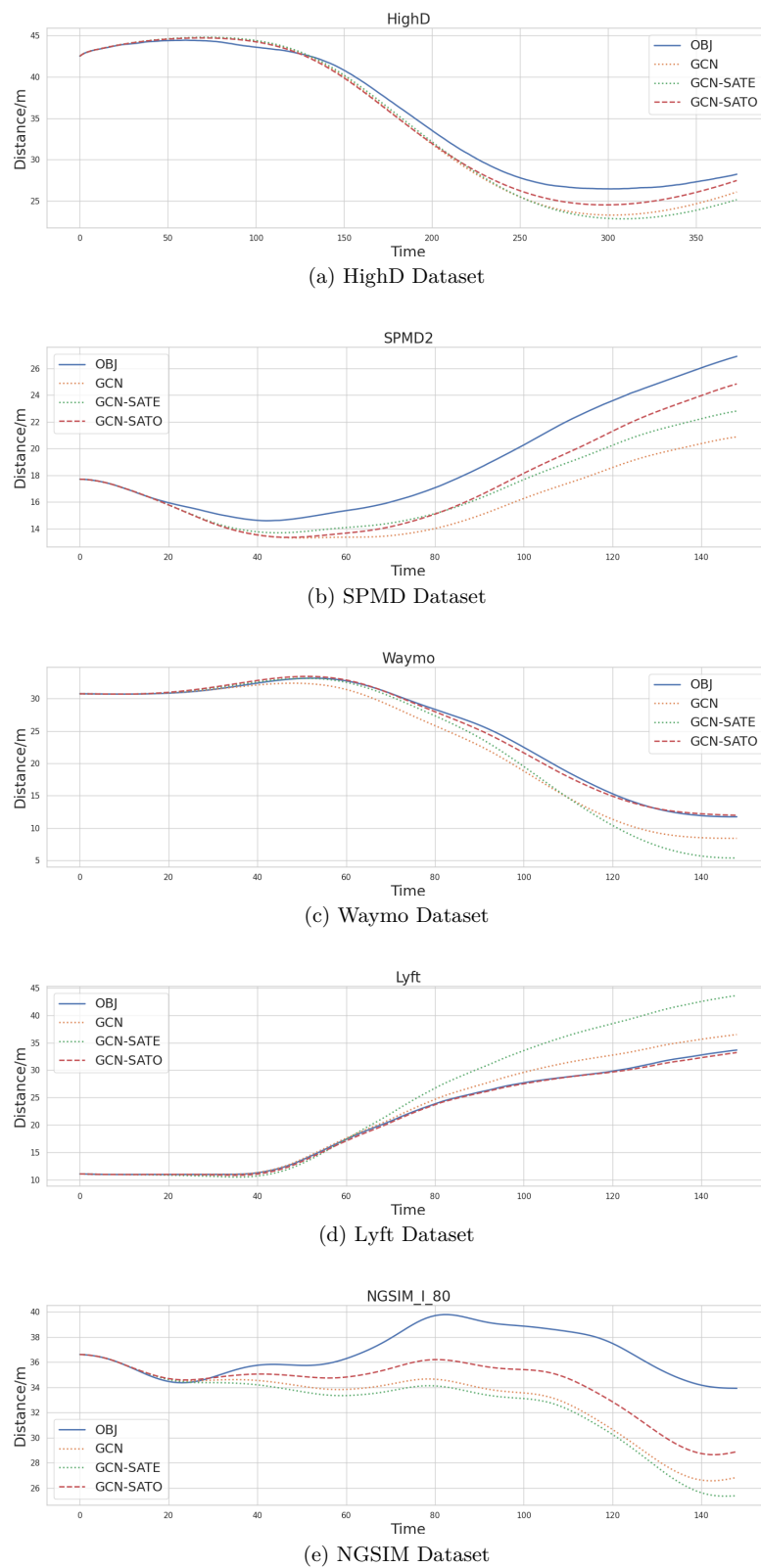


Figure 2: Ablation experiment: Performance of models on different Datasets

car-following process. Therefore, it is not difficult to find that zero collision rate appears in the Lyft dataset for NN, LSTM, DDPG and CF based GCN-SATO models. Specifically, both the DDPG model[14] and our GCN-SATO model have achieved a zero collision rate in all datasets.

3.3.3. Jerk Comfortable driving during car following is a crucial factor in autonomous driving. A lower Jerk value indicates that the acceleration of vehicle's motion changes more smoothly, resulting in a better driving experience for the vehicle's occupants. In regards to minimum Jerk in Table.2, LSTM outperforms with the lowest values in HighD and Lyft datasets. The lowest values illustrates that the network structure of LSTM can effectively offer a comfortable driving experience. However, it's not hard to find that our model (GCN-SATO) achieves the best results in all datasets, resulting in smoother car-following behaviors. Additionally, as shown in Table.2, DDPG has achieved a relatively higher Jerk across all datasets, leading to a relatively uncomfortable driving experience for the vehicle's drivers.

3.3.4. TTC The Time to Collision (TTC) serves as a critical measure in assessing the safety performance of car-following behaviors. From Table.2, it is noted that our GCN-SATO model achieves the highest minimum TTC values across all datasets, resulting in maintaining safer distance between vehicles. Except for our model, LSTM also demonstrates excellent performance with the higher values in HighD, SPMD, NGSIM and Lyft datasets.

3.4. Ablation Experiment

Table.3 shows the ablation experimental data of CF based GCN, GCN-SATE and GCN-SATO models. The CF based GCN model is a new algorithm obtained by removing self-attention module in output layer with the default CF based GCN-SATO model. The CF based GCN-SATE model is a new algorithm obtained by removing self-attention module in output layer and adding the Module in input layer with the default CF based GCN-SATO model.

As shown in Table.3 and Fig.2, compared with the CF based GCN model, GCN-SATE can effectively reduce $mse_{spacing}$ on HighD, SPMD, NGSIM and Lyft datasets, while actually increase it in Waymo dataset. It means that GCN-SATE does not necessarily capture more realistic traffic flow characteristics than GCN. It's not hard to find that GCN-SATO, compared with GCN and GCN-SATE, can effectively reduce $mse_{spacing}$ in all five datasets, which illustrates that putting self-attention module in output layer can effectively catch complex relationships within the data and additional characteristics from real-world driving datasets, improving the robustness and generalization ability of the car-following model.

4. Conclusion

In this work, a novel car-following based GCN-SATO model is proposed to learn how to control vehicle movement during a car-following event. First, the GCN module is employed to extract spatial-temporal feature of car-following behaviors, enabling the model to effectively obtain topological structure relationships among vehicles. Then, the self-attention module is utilized to assign different attention weights to different neighbors of a node in the car-following model. Finally, the Multi-Layer Perceptron (MLP) layer is used to predict the future behaviors of following vehicles. Human driving data of the real world, including HighD, SPMD, Waymo, NGSIM and Lyft datasets, were used for model training and validation. The experimental results demonstrates satisfactory results and outperforms other existing methods in terms of MSE of spacing($mse_{spacing}$), collision rate, Jerk and TTC. In summary, our proposed GCN-SATO based CF model achieves competitive performance in understanding the human driving behaviors in mixed traffic flows, contributing to a effective, safe and comfortable driving experience for drivers.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62303325, and Shanghai Science and Technology Innovation Action Plan Morning Star Project (Sail Special) (no. 23YF1429700).

References

- [1] Fabian Hart, Ostap Okhrin, and Martin Treiber. Formulation and validation of a car-following model based on deep reinforcement learning. *arXiv preprint arXiv:2109.14268*, 2021.
- [2] Louis A Pipes. An operational analysis of traffic dynamics. *Journal of applied physics*, 24(3):274–281, 1953.
- [3] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, 51(2):1035–1042, 1995.
- [4] Peter G Gipps. A behavioural car-following model for computer simulation. *Transportation research part B: methodological*, 15(2):105–111, 1981.
- [5] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics*, 62(2 Pt A):1805–24, 2000.
- [6] Robert E Chandler, Robert Herman, and Elliott W Montroll. Traffic dynamics: studies in car following. *Operations research*, 6(2):165–184, 1958.
- [7] Tomer Toledo. Driving behaviour: models and challenges. *Transport Reviews*, 27(1):65–84, 2007.
- [8] Gentry Lee. A generalization of linear car-following theory. *Operations research*, 14(4):595–606, 1966.
- [9] Guanghan Peng, Weizhen Lu, Hongdi He, and Zhenghua Gu. Nonlinear analysis of a new car-following model accounting for the optimal velocity changes with memory. *Communications in Nonlinear Science and Numerical Simulation*, 40:197–205, 2016.
- [10] Xiuling Huang, Jie Sun, and Jian Sun. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transportation research part C: emerging technologies*, 95:346–362, 2018.
- [11] Mofan Zhou, Xiaobo Qu, and Xiaopeng Li. A recurrent neural network based microscopic car following model to predict traffic oscillation. *Transportation research part C: emerging technologies*, 84:245–264, 2017.
- [12] Xiao Wang, Rui Jiang, Li Li, Yilun Lin, Xinhua Zheng, and Fei-Yue Wang. Capturing car-following behaviors by deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):910–920, 2017.
- [13] Dianzhao Li and Ostap Okhrin. Modified ddpg car-following model with a real-world human driving experience with carla simulator. *Transportation research part C: emerging technologies*, 147:103987, 2023.
- [14] Xianda Chen, Meixin Zhu, Kehua Chen, Pengqin Wang, Hongliang Lu, Hui Zhong, Xu Han, Xuesong Wang, and Yinhai Wang. Follownet: a comprehensive benchmark for car-following behavior modeling. *Scientific data*, 10(1):828, 2023.
- [15] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.
- [16] Lijing Ma and Shiru Qu. A sequence to sequence learning based car-following model for multi-step predictions considering reaction delay. *Transportation research part C: emerging technologies*, 120:102785, 2020.
- [17] Yangxin Lin, Ping Wang, Yang Zhou, Fan Ding, Chen Wang, and Huachun Tan. Platoon trajectories generation: A unidirectional interconnected lstm-based car-following model. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2071–2081, 2020.
- [18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.