Hindawi
Journal of Advanced Transportation
Volume 2023, Article ID 1286977, 12 pages
https://doi.org/10.1155/2023/1286977

Research Article

TCN-SA: A Social Attention Network Based on Temporal Convolutional Network for Vehicle Trajectory Prediction

Qin Li,[1] Bingguang Ou,[1] Yifa Liang,[1] Yong Wang,[2] Xuan Yang,[1] and Linchao Li[3]

[1]*School of Mechanical Engineering, Guangxi University, Nanning 530004, China*
[2]*School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China*
[3]*Urban Smart Transportation Safety Maintenance, Shenzhen University, Shenzhen 518060, China*

Correspondence should be addressed to Yong Wang; 17862709675@163.com

Received 5 May 2023; Revised 22 September 2023; Accepted 26 October 2023; Published 9 December 2023

Academic Editor: Jingda Wu

Vehicle trajectory prediction can provide important support for intelligent transportation systems in areas such as autonomous driving, traffic control, and traffic flow optimization. Predicting vehicle trajectories is an extremely challenging task that not only depends on the vehicle's historical trajectory but also on the dynamic and complex social-temporal relationships of the surrounding traffic network. The trajectory of the target vehicle is influenced by surrounding vehicles. However, existing methods have shortcomings in considering both time dependency and interactive dependency between vehicles or insufficient consideration of the impact of surrounding vehicles. To address this issue, we propose a hybrid deep learning model based on a temporal convolutional network (TCN) that considers local and global interactions between vehicles. Specifically, we use a social convolutional pooling layer to capture local interaction features between vehicles and a multihead self-attention layer to capture global interaction features between vehicles. Finally, we combine these two features using an encoder-decoder structure to predict vehicle trajectories. Through experiments on the Next-Generation Simulation (NGSIM) public dataset and ablation experiments, we validate the effectiveness of our model.

1. Introduction

Trajectory prediction plays a crucial role in intelligent transportation systems (ITSs) as it helps autonomous vehicles perceive the current behavior of surrounding agents (SAs) and continuously predict their future actions to maintain efficient motion planning and navigation decisions, ensuring safety for all agents [1, 2]. Additionally, trajectory prediction is beneficial for vehicle communication, vehicle control, and traffic safety and management, reducing high latency and data transmission interruptions in the vehicle network through early registration and resource allocation [3–7]. In *L*2 and *L*3 intelligent driving with mixed traffic flow, trajectory prediction is essential for maneuvering and potential collision warnings, as human driver intentions are often unavailable [8, 9]. Therefore, trajectory prediction is one of the indispensable capabilities for autonomous driving.

Predicting the future trajectories of surrounding vehicles is an incredibly challenging task. It relies not only on the vehicle's historical trajectory but also on the dynamic and complex social-temporal relationships of the surrounding traffic network [10]. Early traditional methods used physics-based models such as constant velocity (CV) or Kalman filters (KFs) [11], which only considered the agent's dynamics to generate the agent's future path. However, these methods are only applicable to short-term trajectory prediction and relatively simple traffic scenarios. They mainly focus on individual historical information of each vehicle and ignore the complex social interaction between vehicles [12]. Early works typically relied on traditional machine learning methods such as Bayesian learning, hidden Markov model (HMM), support vector machine (SVM), and Gaussian process (GP) [13] for trajectory prediction. However, these methods require manually crafting features from raw data, which cannot serve as a universal

representation of complex traffic environments. Therefore, traditional methods are difficult to obtain satisfactory accuracy, especially for long-term prediction (3–5 seconds), which has been proven to be more challenging than short-term prediction (1–3 seconds) [14].

As a branch of machine learning, deep learning can automatically extract features from rich data to overcome the limitations of handcrafted features. Trajectory data can be viewed as interactive multivariate time series, so capturing time and interaction dependencies is one of the key steps for achieving accurate predictions. In order to capture the temporal correlations between trajectories at different timestamps, recurrent neural networks (RNNs), especially a range of improved variants such as long short-term memory (LSTM) [15], gated recurrent unit (GRU) [16], and bidirectional RNN (BiRNN) [17], have been widely used for trajectory prediction due to their ability to model sequential data. However, due to structural limitations, RNN and its variant networks suffer from low computational efficiency and are unable to capture long-term dependencies on excessively long sequences, making them unsuitable for long-term prediction. Faced with these shortcomings of RNN, another new sequence processing network has been proposed. In a study by Bai et al. [18], a specialized convolutional neural network called temporal convolutional network was designed for processing sequential data, such as time series and natural language [19–21]. Employing dilated causal convolution layers, TCN effectively captures long-term dependencies across various time scales within input sequences. Previous research has documented significant performance improvements achieved by the TCN model in both regression and classification tasks [21–24].

However, modeling historical time series alone is not sufficient for vehicle trajectory prediction, as it also needs to consider the complex interaction between vehicles. For instance, in dense highway environments, if a driver attempts to change lanes, drivers in adjacent lanes may slow down to make way. Therefore, in order to accurately predict future trajectories, besides the raw historical trajectories, the interactions among participants need to be considered as one of the parameters for the model's prediction.

In this article, we introduce a network that utilizes the TCN and attention mechanisms to model the historical temporal features and interaction features of vehicles. This approach is aimed at enhancing prediction performance. The main contributions of the article are as follows:

(1) We use an encoder-decoder structure based on TCN to capture temporal dependencies and improve computational efficiency.

(2) We use social pooling layers to capture local interaction features between vehicles and attention layers to capture global interaction features. Then, we combine the local and global interaction features to assist in prediction.

(3) We conducted experiments on public datasets, and the experimental results of trajectory prediction show that the proposed model is superior to classical models.

2. Related Work

By predicting the trajectories of surrounding vehicles, intelligent cars can react to changes in the motion state of surrounding vehicles in advance, make accurate decisions on future traffic situations, and plan safe, easily controllable, comfortable, and not overly conservative driving trajectories. After summarizing, there are mainly three types of vehicle trajectory prediction methods: physics-based methods [25, 26], behavior-based methods [27, 28], and deep learning-based methods [29, 30].

The vehicle trajectory prediction method based on physical models simplifies the target vehicle into a relatively simple vehicle dynamic or kinematic model. It iteratively calculates the future state of the vehicle based on inputs to the model, such as acceleration and steering angle, as well as external conditions like road surface friction coefficient [31].

The dynamic model is based on different forces acting on the vehicle during motion, such as longitudinal and lateral tire forces, to model the vehicle's motion [32]. The kinematic model is based on the mathematical relationship between vehicle motion parameters, such as position, velocity, and acceleration, without considering the forces that affect motion [33]. In trajectory prediction research, because the internal parameters required by the dynamic model are difficult to observe with the vehicle's sensors, the use of the kinematic model is more common. Study [34] proposed a trajectory prediction model based on constant velocity and acceleration by assuming that the predicted temporal vehicle motion state remains unchanged, but this method does not consider the varying characteristics of vehicle lateral dynamics in the prediction time domain and is not applicable to conditions such as vehicle turning and lane changing. Study [35] studied the variation characteristics of vehicle yaw rate within the prediction time domain based on the above model, which effectively characterizes the vehicle's lateral position changes and has high online calculation efficiency. However, the above method assumes that most vehicle states remain unchanged within the prediction time domain, ignoring their uncertainty, and thus only applies to short-term prediction. To solve the above problems, study [36] used a mixture of Gaussian matrix to model the uncertainty of vehicle states and used a switching Kalman filter to predict future trajectories, while another study [37] established a model for characterizing the uncertainty of model input variables based on Monte Carlo methods to improve the prediction accuracy of motion trajectories. However, the above methods did not fully consider the impact of vehicle interaction behavior on the uncertainty of predicted trajectories.

Existing behavior identification algorithms can be divided into methods based on driving behavior classifiers (such as support vector machines and multilayer perceptron) [38] and methods based on probabilistic graph models (such as Markov random fields and Monte Carlo sampling) [39–41]. In terms of trajectory prediction based on behavior identification, study [42] generated predicted trajectories of different lengths using a kinematic model by combining the

identified driving behavior with the current vehicle state, but this method ignored the uncertainty of the vehicle's current state and driving behavior. To solve the problem of modeling vehicle state and behavior uncertainty, Gaussian process methods and random search tree methods are widely used. For example, another study [43] fitted a Gaussian process based on vehicle historical trajectory training data that can satisfy the probability distribution characteristics of driving behavior and used this to create sample trajectories for each behavior. Study [36] researched the sampling method of vehicle model input parameters and obtained the probability distribution characteristics of predicted trajectories based on Gaussian process and fast search random tree algorithm combined with the results of vehicle behavior recognition.

Many deep learning-based methods use end-to-end learning models for trajectory prediction, which take the historical observation information of the target vehicle as input and directly output various types of predicted trajectories. Such methods can effectively combine prior and posterior knowledge in traffic scenarios to achieve long-term trajectory prediction while maintaining good computational efficiency. Study [44] proposed an LSTM network based on an encoder-decoder structure, which uses a convolutional network to extract vehicle spatial grid features and ultimately outputs a multimodal distribution of predicted trajectories. Study [33] introduced an encoder-decoder structure LSTM network based on spatiotemporal occupancy grid maps. The maximum prediction duration can reach 2 seconds, but the training data for the above models need to be manually annotated, increasing the difficulty of model training. A different study [45] used graph convolutional networks to extract vehicle interaction features and utilized an encoder-decoder structure LSTM network to simultaneously output different vehicle predicted trajectories. The aforementioned LSTM-based trajectory prediction methods have solved problems such as gradient vanishing and explosion during long-term sequence training and have higher prediction accuracy in long-cycle prediction. However, LSTM networks have disadvantages such as complex structure, large computation, inability to perform parallel computing, and long training time.

There have been numerous studies incorporating TCN and social modules into trajectory prediction tasks. For instance, paper [46] addressed the issue of shared LSTM models neglecting the uniqueness of the ego-perspective, which hinders the extraction of interaction features from the perspective of ego pedestrian. They employed two separate LSTM models as social modules to extract features for ego pedestrian and their neighbors. Subsequently, they used egocentric features to guide an attention mechanism for aggregating the features of interacting neighbors, thereby generating effective interaction features. An innovative dual-attention architecture for encoding observations in RNNs was introduced in [47]. This architecture effectively handles both environmental (map) and social (neighbors) features in a unified manner, providing a comprehensive approach to information integration. In [48], a specially designed deep neural network with a switch-like structure, incorporating TCN layers, BiLSTM, and attention mechanisms, was employed. This approach led to improved predictive performance. In [49], a time convolutional network with attention mechanism (TCN-ATM) model was developed for lane-changing intention recognition. When considering an input sequence of 150 frames, the proposed TCN-ATM model achieved an impressive overall classification performance of 98.20%.

In summary, physics-based trajectory prediction methods only consider the constraints of vehicle motion characteristics on the trajectory and do not take into account the impact of factors such as road structure, traffic rules, and vehicle historical trajectory on the future motion state of the vehicle. This results in problems such as low prediction accuracy and poor environmental adaptability, limiting its use to low-speed short-term prediction. Compared to physics-based prediction methods, behavior-based prediction methods can achieve better prediction accuracy and longer prediction time. However, in complex multivehicle interaction traffic scenarios, there are issues with low scene adaptability and poor robustness. Deep learning-based trajectory prediction methods incorporate intervehicle interaction information into the training process through various data fusion techniques. After multiple rounds of training to converge the loss value, they can output different types of long-term prediction results. However, existing methods do not fully consider the modeling of intervehicle interaction behavior or do not consider vehicle-to-vehicle interactions, making it difficult to apply them to complex traffic scenarios. Therefore, during the model training process, it is necessary to fully model the interaction behavior to further improve prediction accuracy.

3. Problem Formulation

As shown in Figure 1, we assume that the autonomous vehicle can observe the motion of vehicles within ± 90 feet longitudinally and in adjacent lanes laterally and can collect their past trajectories at a certain frequency.

The trajectory prediction can be formulated as a problem that estimates the future positions of target vehicle based on all previously observed trajectories. Specifically, let X denote the past trajectories of all observed vehicles in a traffic scene:

$$X = [p_1, p_2, \ldots p_t, \ldots, p_T], \tag{1}$$

where $t = 1, 2, \ldots, T$ is the timestamp and $p_t = [(x_t^1, y_t^1), \ldots, (x_t^n, y_t^n), \ldots, (x_t^N, y_t^N)]$ represents the observation states of all vehicles at time t, including the target vehicle and surrounding vehicles. x_t and y_t are the coordinates of vehicles at time t. N is the number of vehicles.

The predicted trajectory in the future time range L is represented as

$$Y = \left[\widehat{P}_{T+1}^{\text{tar}}, \widehat{P}_{T+2}^{\text{tar}}, \ldots, \widehat{P}_{T+t}^{\text{tar}}, \ldots, \widehat{P}_{T+L}^{\text{tar}}\right], \tag{2}$$

where $\widehat{P}_{T+L} = (x_{T+L}^{\text{tar}}, y_{T+L}^{\text{tar}})$ represents the future position of the target vehicle at time $T + L$.

In summary, the problem of using historical observation data X to predict the future trajectory Y of the target vehicle can be formulated as finding a mapping relationship F between Y and X:

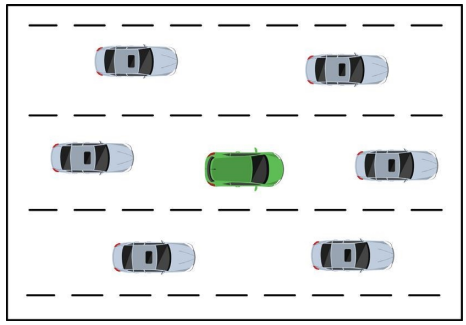

Figure 1: The traffic scene in this paper, where the green one is the target vehicle (TV) and the grey ones are the surrounding vehicles (SVs).

$$Y = F(X). \tag{3}$$

4. Model Overview

In this section, we introduce our deep learning-based trajectory prediction method. Figure 2 illustrates the proposed model, named TCN-SA, which consists of a TCN-based encoder-decoder structure. To extract both local and global interaction features among vehicles, we embed convolution pooling modules and multihead attention modules. Detailed information for each part is as follows.

4.1. Encoder-Decoder Structure Based on TCN. To capture the time dependence of historical vehicle data X, we employ a TCN encoder. Specifically, we feed the feature sequence X into the TCN encoder as input and obtain its output, denoted as $X_{\text{enc}} \in R^{N\times T\times H}$, where H is the dimensionality of the TCN layer's hidden state.

Both the TCN encoder and decoder have similar structures, as shown in Figure 3. Figure 4 illustrates the encoder-decoder structure based on TCN. TCN utilizes causal convolution and dilated convolution. Causal convolution is a time-constrained model that only allows accessing information up to the current time step t and its past values in the upper layer, while future information cannot be accessed. On the other hand, dilated convolution introduces a hyperparameter called dilation, which specifies the spacing between the values in the kernel. This enables the convolutional layer to process temporal data with larger receptive fields while keeping the same number of parameters. The formula for dilation convolution can be expressed as follows:

$$Y_{t,i} = \sum_{d=1}^{D}\sum_{k=1}^{K} W_{k,d}\cdot X_{(t\cdot d)+k-1,d}, \tag{4}$$

where Y represents the output feature map, t represents the time step, i represents the output channel, W is the convolution kernel, K is the kernel size, and D represents the input feature dimension. We use a convolution kernel of length K and dilation rate d in this formula, where each element of the convolution kernel $W_{k,d}$ is weighted and summed with d adjacent elements from the input X, and the results are accumulated to obtain the output $Y_{t,i}$.

4.2. Social Pooling Module. The TCN encoder can only capture temporal dependencies between time steps and cannot capture motion correlations between vehicles. To address this issue, we adopt a similar approach as in article [44] to construct a social tensor among vehicles and then use a social pooling layer to capture the motion correlations between vehicles. To reduce computational complexity, we use only a total of 7 cars from the scenario in Figure 1 to construct a 9×9 social tensor. We then use two layers of 2×2 convolutional layers with padding = 1 and a pooling layer to build the social pooling module. The constructed social tensor is shown in Figure 2.

4.3. Interaction Module Based on Multihead Attention Mechanism. As the social pooling layer is built upon convolutional neural networks which are restricted by convolutional kernels, they can only capture local features. Therefore, we believe that the social pooling layer can only capture local vehicle interaction features rather than global interaction features. To address this issue, we have designed a global interaction feature capturing module based on multihead self-attention mechanism.

As shown in Figure 5, firstly, we use the encoding vector output by the TCN encoder to generate the query vector, key vector, and value vector. The formula for generating Q, K, and V is as follows:

$$\begin{aligned} Q_j &= W_{\text{qj}} X^t_{\text{enct}}, \\ K_j &= W^j_k X^t_{\text{enc}}, \\ V_j &= W_{\text{vj}} X^t_{\text{enct}}, \end{aligned} \tag{5}$$

where j represents the j-th attention head and W_q, W_k, W_v are learnable weight matrices.

Then, we can calculate the attention coefficients based on Q, K, and V vectors:

$$\alpha_j = \text{softmax}\left(\frac{Q_j K_j^{\top}}{\sqrt{d_k}}\right), \tag{6}$$

where d_k is the dimensionality of the K_j vector.

The output vector of the multihead self-attention mechanism is as follows:

$$\text{I} = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_j\right) W^O, \tag{7}$$

where $\text{head}_j = \alpha_j \text{V}_j$ and W^O is a learnable weight matrix used to map the concatenated vector back to the original dimension.

After the above operations, the vector I we obtain has aggregated the features of all other vehicles.

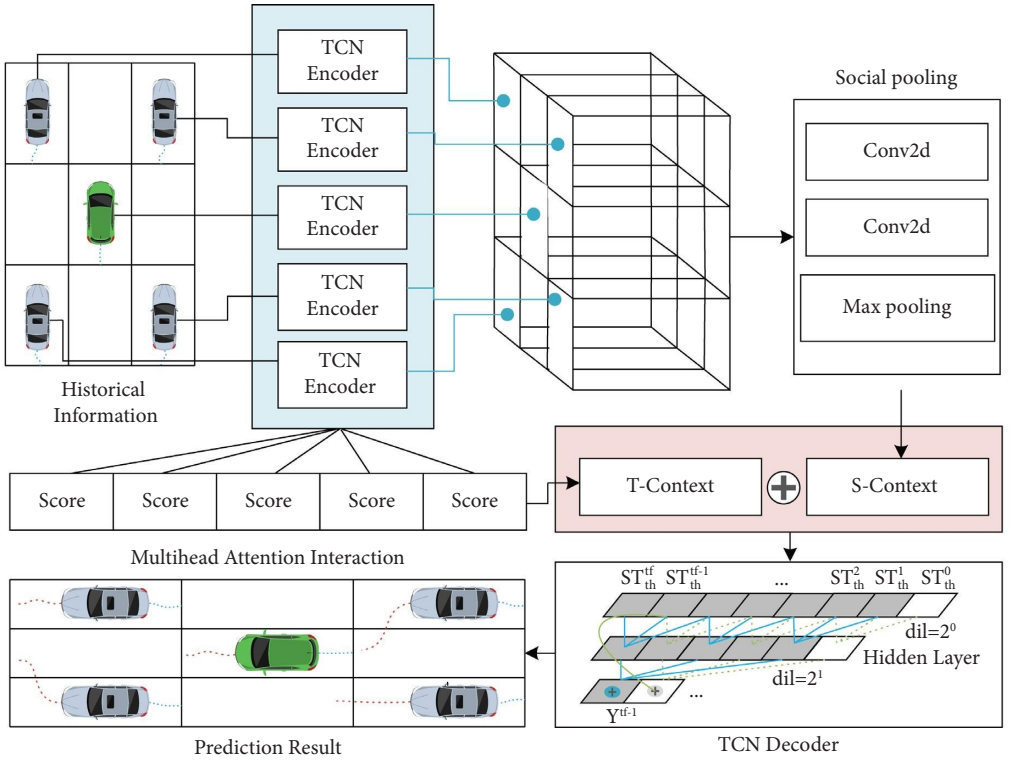

Figure 2: The proposed prediction model: TCN-SA.

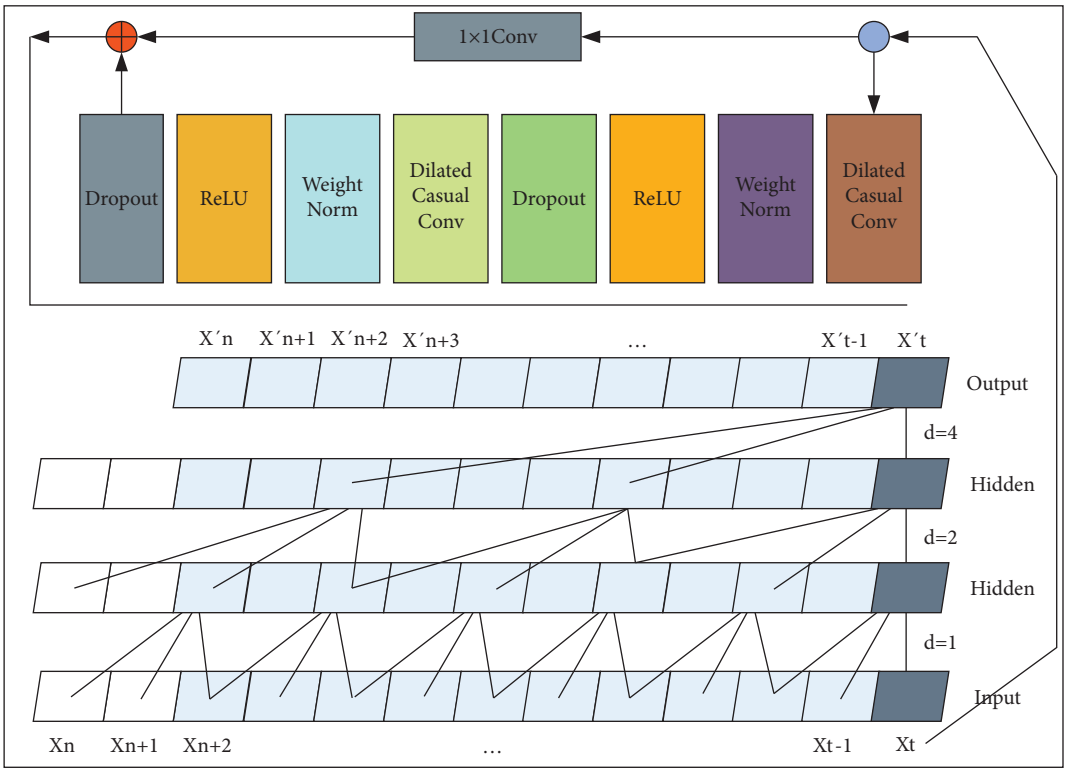

Figure 3: Internal structure of TCN.

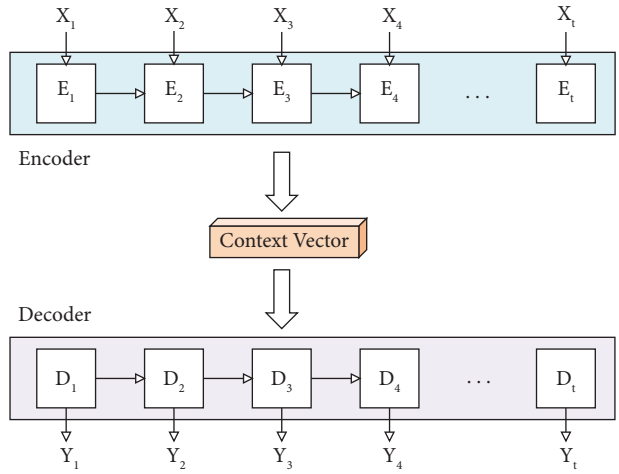

Figure 4: Encoder-decoder structure based on TCN.

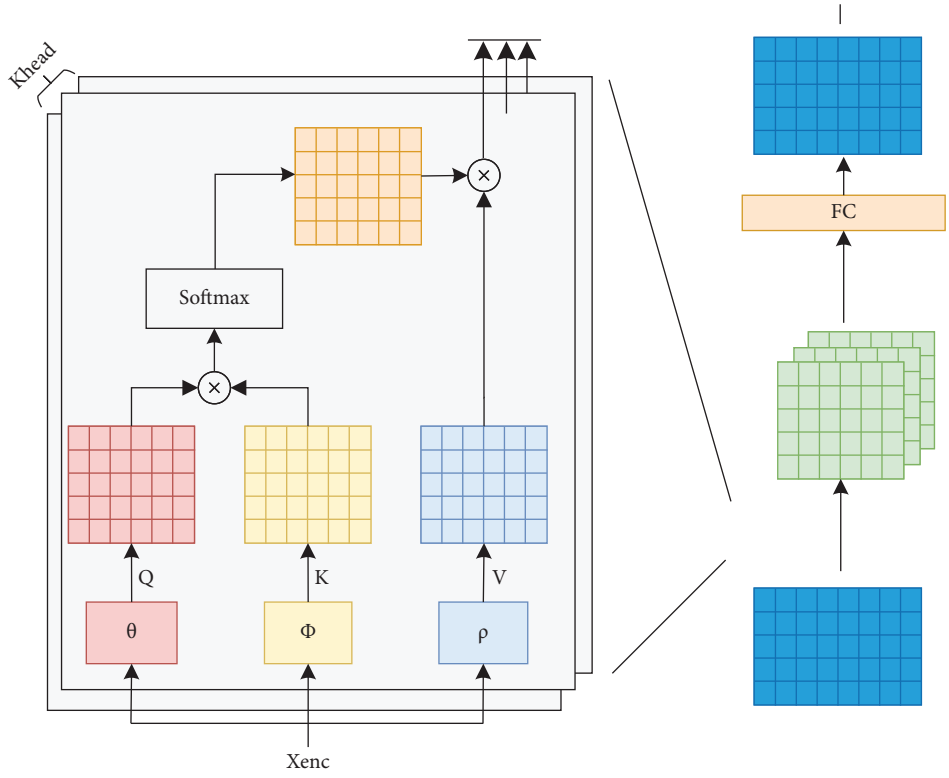

Figure 5: Interaction module based on multihead self-attention mechanism.

4.4. Distance-Based Spatial Information Embedding. To embed spatial information between vehicles in the process of capturing interaction features, we constructed a spatial embedding based on distance. As shown in Figure 6, we first calculate the relative distance d_{ij} between each pair of vehicles, then divide d_{ij} by a scaling factor a (where we take $\alpha = 90$) to obtain d_{ij}', and use d_{ij}' to construct a distance adjacency matrix D. Finally, we perform softmax normalization on the distance adjacency matrix D to obtain the spatial embedding representation matrix S:

$$D_{\mathrm{ij}} = \frac{d_{\mathrm{ij}}}{90},$$
$$S = \mathrm{softmax}(D), \tag{8}$$

we integrate the obtained spatial embedding information into 6.

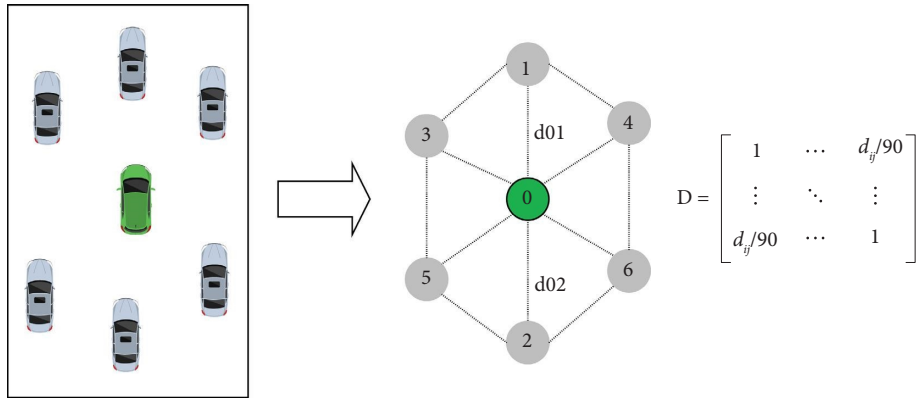

Figure 6: Example of using distance to construct spatial distance embedding.

$$\alpha_j' = \text{softmax}\left(\frac{Q_j\left(K_j^{\top}\right)}{\sqrt{d_k}}\right) + S. \tag{9}$$

We can thus embed the relative spatial relationships between vehicles in the process of capturing interaction features.

5. Experiments

5.1. Dataset. Our model was trained on two publicly available vehicle trajectory datasets: I-80 and US-101 from NGSIM [50], which record trajectories at a frequency of 10 Hz in real highway scenarios. Both datasets contain vehicle trajectories for 45 minutes under light, moderate, and heavy traffic conditions, providing rich scenes for evaluating the robustness and effectiveness of the proposed network.

To ensure a fair comparison, we followed the training strategy outlined in reference [44]: we downsampled the raw data to 5 Hz and divided the trajectories into multiple segments every 8 seconds, with the first 3 seconds of each segment used as the past time range and the remaining 5 seconds as the prediction time range. In total, the sample data are split with the ratio of 7 : 1 : 2 into training sets, validation sets, and test sets.

5.2. Metric. To assess performance in a quantitative manner, the root mean square error (RMSE) is utilized to measure the disparity between predicted and ground truth trajectories. The RMSE is computed by taking the square root of the average of the squared differences between corresponding elements in the predicted and ground truth trajectories:

$$\text{RMSE}_t = \sqrt{\left(\hat{x}_t^{\text{tar}} - x_t^{\text{tar}}\right)^2 + \left(\hat{y}_t^{\text{tar}} - y_t^{\text{tar}}\right)^2}, \tag{10}$$

where $\hat{x}_t^{\text{tar}}$ and $\hat{y}_t^{\text{tar}}$ are the predicted coordinate values of the target vehicle at time step t.

5.3. Compared Models. The baseline models we use for comparison are as follows:

Constant velocity (CV): we use a constant velocity Kalman filter as our simplest baseline.

C-VGMM + VIM: as our second baseline, we have utilized maneuver-based variational Gaussian mixture models along with a Markov random field-based module for vehicle interaction, which is described in [51].

GAIL-GRU: This is a Generative Adversarial Imitation Learning model [52]. As both studies used the same dataset, we will directly cite the results from the original paper.

Convolutional Social Pooling LSTM (CS-LSTM): After establishing a grid of size 13 × 3, the target vehicle is placed at the center of the grid, and then surrounding vehicles are put into the grid to build a social tensor, and then the interactive information is extracted for prediction [44].

Nonlocal Social Pooling (NLS-LSTM): LSTM is used in an encoder-decoder structure that captures social interaction by combining operations that focus on both nearby and more distant interactions [53].

Social GAN (S-GAN): the model utilizes a combination of a recurrent sequence-to-sequence model and a generative adversarial network in order to gather information from different agents, which enables the generation of multiple possible outcomes that are socially realistic [54].

5.4. Ablation Experimental Models. The ablation experimental models we use for comparison are as follows:

Encoder-decoder (TCN-based): In order to conduct basic ablation experiments, we have chosen a sequence to sequence structure based on the TCN model for our model. This will enable us to compare different versions of our model and identify the impact of removing specific components.

Ours (removing convolutional social pooling module): to validate the effectiveness of our embedded convolutional social pooling module in extracting local interaction features, we removed this module from our model for ablation experiments.

Ours (removing multihead attention interaction module): to validate the effectiveness of our embedded

Table 1: Experimental results of different models.

Metric	Prediction horizon (s)	CV	C-VGMM + VIM	GAIL-GRU	CS-LSTM	NLS-LSTM	S-GAN	TCN-SA
RMSE (m)	1	0.73	0.66	0.69	0.61	0.56	0.57	**0.54**
	2	1.78	1.56	1.51	1.27	1.22	1.32	**1.15**
	3	3.13	2.75	2.55	2.09	**2.02**	2.22	2.14
	4	4.78	4.24	3.65	3.10	3.03	3.26	**2.87**
	5	6.68	5.99	4.71	4.37	4.30	4.40	**4.08**

multihead attention interaction module in extracting global interaction features, we removed this module from our model for ablation experiments.

5.5. Experimental Results and Analysis

5.5.1. Comparison between Different Models. Table 1 shows the RMSE results of different models under different prediction horizons. In Table 1, each row represents different prediction horizon, and each column shows the RMSE of different models in their respective prediction perspectives. Smaller RMSE values indicate better performance, and the text highlighted in bold black represents the optimal value for each prediction horizon, corresponding to the best-performing model. We found that the first three models showed greater RMSE in both short-term and long-term predictions. The last four models obtain lower RMSE by taking into account the information or interaction information of other surrounding vehicles. As CS-LSTM and NLS-LSTM are models based on the LSTM framework, their performance differences are not significant. S-GAN combines social rules in GAN to generate multimodal results, so its performance is slightly inferior to LSTM-based frameworks. Our proposed model utilizes TCN to capture temporal dependencies, using convolutional social pooling layers and multihead attention to capture global and local interaction features between vehicles, respectively. In most cases, our model's performance is significantly better than the baseline model.

5.5.2. Comparison between Different Ablation Experiments. We conducted three sets of ablation experiments: removing the social convolutional pooling layer, removing the attention interaction layer, and removing both the attention interaction layer and the social convolutional pooling layer. Table 2 shows our results.

When completely removing the interaction module (i.e., simultaneously removing both the attention interaction module and the convolutional pooling module), the model degrades into a TCN-based Seq2seq model, that is, the interaction information between vehicles is not used, under various prediction horizons, the model's performance significantly deteriorates, and it also shows the importance of interaction characteristics between vehicles from the side. However, when only removing either the social pooling module or the attention interaction module individually, the model's performance improves significantly compared to the Seq2seq model. We believe that in the case of using only the social pooling module or only the attention interaction module, the single interaction feature obtained (local interaction feature or global interaction feature) is insufficient to reflect the true interaction dynamics between vehicles. It is only when combining local and global interaction features that better performance is achieved. This also indirectly indicates that utilizing more auxiliary information is advantageous for improving the predictive model's performance. Furthermore, it suggests that there is a complementary effect between local and global interaction features.

5.5.3. Visualization of Prediction Results. As shown in Figure 7, we randomly selected one sample and visualized its results under different prediction horizons. The predicted horizons represented from top to bottom are 1 s, 3 s, 4 s, and 5 s, respectively. As the prediction horizon increases, we can observe that the longitudinal error of the predicted trajectory gradually becomes larger (the gap between the green line and the black line widens). This is a normal phenomenon because trajectory prediction is a complex nonlinear temporal prediction problem, and as the prediction horizon increases, the prediction error also gradually accumulates and increases.

As shown in Figure 8, the three columns from left to right represent three lateral maneuvers: lane keeping, right lane changing, and left lane changing. The three rows from top to bottom represent three longitudinal maneuvers: uniform speed driving, deceleration driving, and acceleration driving. Three lateral maneuvers and three longitudinal maneuvers can form a total of nine possible future driving maneuvers. With regard to lane-keeping scenarios, we observed that our prediction model demonstrates relatively good performance, regardless of whether the vehicle is accelerating, decelerating, or maintaining a constant speed. In lane-changing scenarios, we have observed that both lateral and longitudinal errors are larger in comparison to lane-keeping scenarios. This is because our prediction only involves single-modal prediction. In scenarios such as lane changes to the left or right, our model may give a result that tends towards the average mode, resulting in slightly larger errors in these two scenarios. Overall, compared to CS-LSTM, our model can provide results that are closer to the real trajectories.

As for other models that perform worse or similar to CS-LSTM, we did not provide visualizations because our data are directly sourced from the respective method's papers, following the same approach as the original CS-LSTM paper [44]. The premise for doing so is that all methods were experimented under fair conditions.

Table 2: Experimental results of ablation.

Metric	Prediction horizon (s)	Without social pooling	Without attention interaction	Seq2seq (TCN)	Complete model
RMSE (m)	1	0.55	0.57	0.64	**0.54**
	2	1.18	1.17	1.56	**1.15**
	3	2.38	2.14	2.97	**2.14**
	4	2.94	2.89	4.28	**2.87**
	5	4.16	4.13	6.03	**4.08**

Bold values indicate the best result within each predicted horizon.

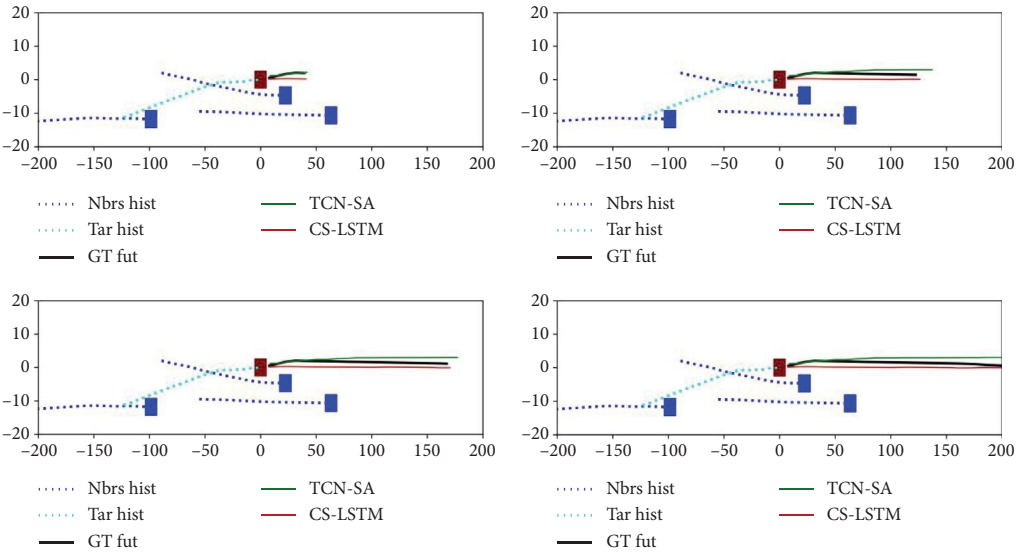

Figure 7: The prediction effects with different predicted horizons.

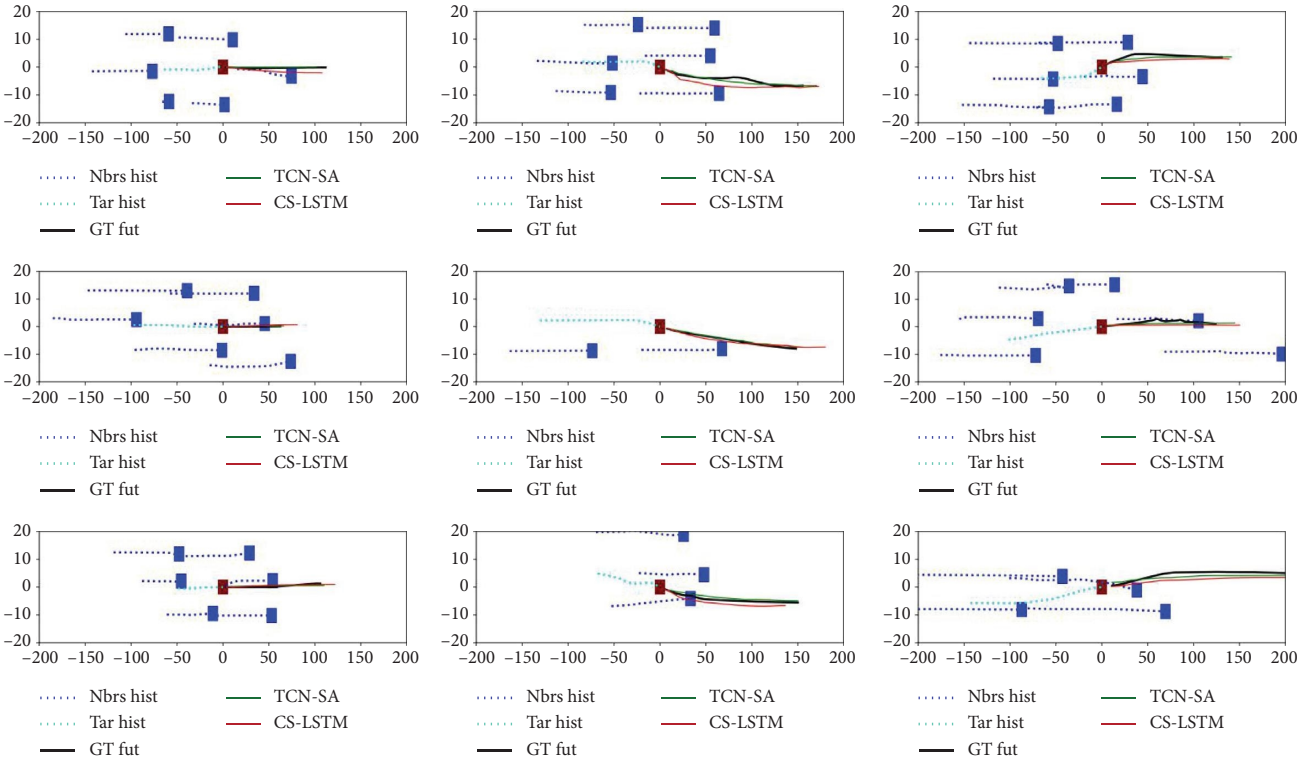

Figure 8: Visualization of prediction results under different driving maneuvers.

6. Conclusions

This paper proposes a hybrid deep learning vehicle trajectory prediction model based on the TCN encoder-decoder structure, which can not only capture the temporal dependencies between historical data but also capture the interaction features between vehicles. Compared with classical models, the effectiveness of our proposed model has been verified, achieving good results in both long-term and short-term predictions. To capture the interaction features between vehicles, we used two modules: the social convolutional pooling module and the multihead attention interaction module, where the social convolutional pooling module can capture the local interaction features between vehicles, and the multihead attention interaction module can capture the global interaction features between vehicles. Through ablative experiments, it has been verified that these two modules are indispensable. Using only one of them cannot fully capture the intervehicle interaction features. Only when both modules are used in conjunction can the model achieve its maximum effectiveness. We also found that adding interaction information between vehicles significantly improved the prediction performance. However, the utilization of surrounding vehicle information is just a simple exploration. Different positions and quantities of surrounding vehicles may have different effects on the experimental results. In future work, we will explore how to select and utilize surrounding vehicle information to improve prediction performance. Furthermore, this paper did not investigate the model's performance in complex road scenarios. In future work, we will select appropriate datasets for further experiments to explore the model's robustness in more complex terrains or traffic conditions.

Data Availability

Researchers for the Next Generation Simulation (NGSIM) program collected detailed vehicle trajectory data on the specified freeway segments of US-101 and I-80, as well as the specified arterial segments of Lankershim Boulevard and Peachtree Street. Data were collected through a network of synchronized digital video cameras. NGVIDEO, a customized software application developed for the NGSIM program, transcribed the vehicle trajectory data from the videos. These vehicle trajectory data provide the precise location of each vehicle within the study area every one-tenth of a second, resulting in detailed lane positions and locations relative to other vehicles. This dataset is widely applied in the field of vehicle prediction. The dataset can be obtained from https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. N. Azadani and A. Boukerche, "Driving behavior analysis guidelines for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6027–6045, 2021.

[2] Y. Wang, H. Tan, Y. Wu, and J. Peng, "Hybrid electric vehicle energy management with computer vision and deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 3857–3868, 2021.

[3] Q. Li, H. Tan, Z. Jiang, Y. Wu, and L. Ye, "Nonrecurrent traffic congestion detection with a coupled scalable bayesian robust tensor factorization model," *Neurocomputing*, vol. 430, pp. 138–149, 2021.

[4] Q. Li, H. Tan, Y. Wu, L. Ye, and F. Ding, "Traffic flow prediction with missing data imputed by tensor completion methods," *IEEE Access*, vol. 8, pp. 63188–63201, 2020.

[5] U. Fattore, M. Liebsch, B. Brik, and A. Ksentini, "Automec: lstm-based user mobility prediction for service management in distributed mec resources," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 155–159, Alicante, Spain, November 2020.

[6] J. Wu, Z. Huang, Z. Hu, and C. Lv, "Toward human-in-the-loop ai: enhancing deep reinforcement learning via real-time human guidance for autonomous driving," *Engineering*, vol. 21, pp. 75–91, 2023.

[7] Q. Li, X. Yang, Y. Wang, Y. Wu, and D. He, "Spatial-temporal traffic modeling with a fusion graph reconstructed by tensor decomposition," 2022, https://arxiv.org/abs/2212.05653.

[8] J. Wu, Z. Huang, W. Huang, and C. Lv, "Prioritized experience-based reinforcement learning with human guidance for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, pp. 1–15, 2022.

[9] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[10] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Attention based vehicle trajectory prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 175–185, 2021.

[11] J. Scharcanski, A. B. de Oliveira, P. G. Cavalcanti, and Y. Yari, "A particle-filtering approach for vehicular tracking adaptive to occlusions," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 381–389, 2011.

[12] R. Jiang, H. Xu, G. Gong, Y. Kuang, and Z. Liu, "Spatial-temporal attentive lstm for vehicle-trajectory prediction," *ISPRS International Journal of Geo-Information*, vol. 11, no. 7, p. 354, 2022.

[13] S. A. Goli, B. H. Far, and A. O. Fapojuwo, "Vehicle trajectory prediction with Gaussian process regression in connected vehicle environment," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 550–555, Changshu, China, December 2018.

[14] R. Chandra, T. Guan, S. Panuganti et al., "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-lstms," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4882–4890, 2020.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[18] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, http://arxiv.org/abs/1803.01271.

[19] G. Guo and W. Yuan, "Short-term traffic speed forecasting based on graph attention temporal convolutional networks," *Neurocomputing*, vol. 410, pp. 387–393, 2020.

[20] S.-J. Li, Y. A. Farha, Y. Liu, M.-M. Cheng, and J. Gall, "Ms-tcn++: multi-stage temporal convolutional network for action segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6647–6658, 2023.

[21] D. Li, Y. Li, C. Wang, M. Chen, and Q. Wu, "Forecasting carbon prices based on real-time decomposition and causal temporal convolutional networks," *Applied Energy*, vol. 331, Article ID 120452, 2023.

[22] Z. Gan, C. Li, J. Zhou, and G. Tang, "Temporal convolutional networks interval prediction model for wind speed forecasting," *Electric Power Systems Research*, vol. 191, Article ID 106865, 2021.

[23] D. Li, F. Jiang, M. Chen, and T. Qian, "Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks," *Energy*, vol. 238, Article ID 121981, 2022.

[24] G. Yating, W. Wu, L. Qiongbin, C. Fenghuang, and C. Qinqin, "Fault diagnosis for power converters based on optimized temporal convolutional network," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[25] M. Khakzar, A. Rakotonirainy, A. Bond, and S. G. Dehkordi, "A dual learning model for vehicle trajectory prediction," *IEEE Access*, vol. 8, pp. 21897–21908, 2020.

[26] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1341–1352, 2020.

[27] H. Woo, Y. Ji, H. Kono et al., "Lane-change detection based on vehicle-trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1109–1116, 2017.

[28] C. Fei, X. He, and X. Ji, "Multi-modal vehicle trajectory prediction based on mutual information," *IET Intelligent Transport Systems*, vol. 14, no. 3, pp. 148–153, 2020.

[29] S. Choi, J. Kim, and H. Yeo, "Attention-based recurrent neural network for urban vehicle trajectory prediction," *Procedia Computer Science*, vol. 151, pp. 327–334, 2019.

[30] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified lstm models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38287–38296, 2019.

[31] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, 2010.

[32] R. Pepy, A. Lambert, and H. Mounier, "Reducing navigation errors by planning with realistic vehicle model," in *Proceedings of the 2006 IEEE Intelligent Vehicles Symposium*, pp. 300–307, Tokyo, Japan, July 2006.

[33] W. Xiao, L. Zhang, and D. Meng, "Vehicle trajectory prediction based on motion model and maneuver model fusion with interactive multiple models," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 2, no. 6, pp. 3060–3071, 2020.

[34] C.-F. Lin, A. G. Ulsoy, and D. J. LeBlanc, "Vehicle dynamics and external disturbance estimation for vehicle path prediction," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 508–518, 2000.

[35] M. Althoff and A. Mergel, "Comparison of Markov chain abstraction and Monte Carlo simulation for the safety assessment of autonomous cars," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1237–1247, 2011.

[36] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1484–1489, Redondo Beach, CA, USA, February 2017.

[37] C. Ding, W. Wang, X. Wang, and M. Baumann, "A neural network model for driver's lane-changing trajectory prediction in urban traffic flow," *Mathematical Problems in Engineering*, vol. 2013, Article ID 967358, 8 pages, 2013.

[38] M. Bahram, C. Hubmann, A. Lawitzky, M. Aeberhard, and D. Wollherr, "A combined model-and learning-based framework for interaction-aware maneuver prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1538–1550, 2016.

[39] X. Peng, Y. L. Murphey, R. Liu, and Y. Li, "Driving maneuver early detection via sequence learning from vehicle signals and video images," *Pattern Recognition*, vol. 103, Article ID 107276, 2020.

[40] Q. Liu, S. Xu, C. Lu, H. Yao, and H. Chen, "Early recognition of driving intention for lane change based on recurrent hidden semi-markov model," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10545–10557, 2020.

[41] Y. Zhang, J. Li, Y. Guo, C. Xu, J. Bao, and Y. Song, "Vehicle driving behavior recognition based on multi-view convolutional neural network with joint data augmentation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4223–4234, 2019.

[42] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.

[43] E. Bertolazzi, P. Bevilacqua, F. Biral, D. Fontanelli, M. Frego, and L. Palopoli, "Efficient re-planning for robotic cars," in *Proceedings of the 2018 European Control Conference (ECC)*, pp. 1068–1073, Limassol, Cyprus, June 2018.

[44] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1468–1476, Salt Lake City, UT, USA, June 2018.

[45] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: framework, algorithms, and verifications," *IEEE/ASME Transactions on mechatronics*, vol. 21, no. 2, pp. 740–753, 2016.

[46] H. Zhou, X. Yang, M. Fan, H. Huang, D. Ren, and H. Xia, "Static-dynamic global graph representation for pedestrian trajectory prediction," *Knowledge-Based Systems*, vol. 277, Article ID 110775, 2023.

[47] P. Xu, J.-B. Hayet, and I. Karamouzas, "Context-aware timewise vaes for real-time vehicle trajectory prediction," 2023, https://arxiv.org/abs/2302.10873.

[48] K. Shi, Y. Wu, H. Shi, Y. Zhou, and B. Ran, "An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network," *Physica A: Statistical Mechanics and Its Applications*, vol. 599, Article ID 127303, 2022.

[49] M. Abdel-Aty and O. Zheng, "Lane change intention recognition and vehicle status prediction for autonomous vehicles," 2023, https://arxiv.org/abs/2304.13732.

[50] Gsimulation, "Us highway 101 dataset," 2007, https://www.fhwa.dot.gov/publications/research/operations/07030/.

[51] N. Deo, A. Rangesh, and M. M. Trivedi, "How would surround vehicles move? a unified framework for maneuver

classification and motion prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 129–140, 2018.

[52] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 204–211, Redondo Beach, CA, USA, February 2017.

[53] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Non-local social pooling for vehicle trajectory prediction," in *Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 975–980, Paris, France, June 2019.

[54] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2255–2264, Salt Lake City, UT, USA, June 2018.