SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2026, Vol. 69, Iss. 3, 132202:1–132202:16
https://doi.org/10.1007/s11432-024-4538-3

Advanced trajectory prediction framework integrating diverse driving styles for autonomous vehicles

Juqi HU[1], Caini WANG[1], Subhash RAKHEJA[2], Youmin ZHANG[2], Changyin SUN[1,3], Hejia GAO[1*] & Darong HUANG[1]

[1]*School of Artificial Intelligence, Anhui University, Anhui 230601, China*
[2]*Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montreal QC H3G 1M8, Canada*
[3]*School of Automation, Southeast University, Nanjing 210096, China*

Received 23 October 2024/Revised 23 March 2025/Accepted 24 July 2025/Published online 23 October 2025

Abstract With the prevalence of autonomous vehicles, accurately predicting the motions of surrounding vehicles has become one of the key safety concerns for autonomous driving in mixed traffic. However, human drivers' personalized driving habits and necessary vehicle interactions complicate the motion prediction process. To address these challenges, this paper proposes an innovative personalized trajectory prediction framework that comprehensively considers individual driving styles and their interactions with surrounding vehicles. This framework comprises two main components: driving style classification and vehicle trajectory prediction. Initially, principal component analysis (PCA) and fuzzy c-means (FCM) algorithms are combined to effectively capture and categorize multiple driving styles. Based on the style information, a global-local enhanced attention module with residual bidirectional long short-term memory network (GLEA-ResBiLSTM) is constructed to integrate local and global information to obtain the predicted trajectory. Lastly, the next-generation simulation (NGSIM) dataset is used to validate the high performance of the proposed prediction framework. By comparing with the conventional methods, the proposed method reduces the root mean square error (RMSE) by 34.12% in 5 s predictions, which intensely shows the enhanced accuracy and innovative practicality of the proposed prediction framework for autonomous driving.

Keywords intelligent driving, vehicle trajectory prediction, personalized driving style, LSTM, attention module

Citation Hu J Q, Wang C N, Rakheja S, et al. Advanced trajectory prediction framework integrating diverse driving styles for autonomous vehicles. Sci China Inf Sci, 2026, 69(3): 132202, https://doi.org/10.1007/s11432-024-4538-3

1 Introduction

Serving as a critical pillar of a nation's economy, the automotive industry is an integral part of technological power [1]. As traffic congestion and accidents have become serious problems due to the increasing number of vehicles, the realization of safe, efficient, and reliable autonomous driving has emerged as an urgent demand [2]. However, the varying driving habits of traffic participants and their complex interactions with surrounding vehicles make accurately predicting future trajectories a significant challenge. It is noted that human drivers typically observe surrounding traffic participants and predict their future trajectories before performing appropriate driving maneuvers including accelerating, braking, or lane changing [3]. How to establish an efficient and accurate trajectory prediction system in highly complex driving environments has become one of the primary challenges faced by autonomous vehicles [4,5].

Many researchers have made significant contributions to vehicle trajectory prediction by proposing various effective methods. These methods can be categorized into physical model-based, classical machine learning-based, and deep learning-based methods. Methods based on physical models typically consider the kinematic and dynamic constraints of vehicles related to such as yaw rate, acceleration, and road surface friction coefficient. Typical approaches include Kalman filter (KF) methods [6–8], and Monte Carlo (MC) methods [9–11]. With combined utilization of vehicle dynamic and identification models, Gao et al. [8] employed the unscented Kalman filter (UKF) for filtering to achieve improved prediction performance. Chen et al. [10] combined deep learning concepts with MC algorithms to predict potential hazards in the current road conditions, enabling vehicles to choose more rational paths. While these methods can achieve short-term trajectory prediction, they do not consider the influence of driving styles on trajectories. To address this issue, MC algorithms are employed in [12] to assess collision probabilities. This method

* Corresponding author (email: hjgao@ahu.edu.cn)

© Science China Press 2025

selects the most suitable trajectory based on probability thresholds corresponding to different driving styles, but it does not perform well in complex dynamic traffic scenarios. It can be observed that methods based on physical models can provide relatively accurate short-term trajectory predictions by considering the kinematic and dynamic constraints of the vehicle. However, these methods are unable to fully handle the diversity of driving behaviors and the complexity of multi-vehicle interactions, which limits their prediction capability in dynamic traffic environments and makes them less effective in addressing complex and constantly changing driving scenarios.

Classical machine learning methods use data-driven approaches for trajectory prediction. The primary methods comprise support vector machines (SVMs) [13,14], hidden markov models (HMMs) [8,15,16], and dynamic Bayesian networks (DBNs) [17–19]. SVMs can learn and recognize driver operations in complex environments. In [13], the nonlinear learning and pattern recognition capabilities of SVM were used to analyze and predict lateral movement trajectories of vehicles based on the NGSIM dataset. HMMs are one of the most classic machine learning-based methods for trajectory prediction. In [15], HMM was employed for trajectory prediction and traffic risk analysis, and the obtained results were integrated into decision-making systems to achieve real-time prediction of road traffic risks. By considering interaction awareness, Wang et al. [16] proposed a vehicle maneuver prediction model by combining game theory with HMM. It is also noted that the DBNs have shown excellent performance in trajectory prediction. In [17], a DBN-based probabilistic method for predicting vehicle trajectories was proposed by integrating driving style information and vehicle dynamics models to achieve accurate long-term trajectory prediction. Compared to traditional physics-based models, classical machine learning methods improve prediction accuracy via data-driven approaches, they often rely on manual feature extraction, which limits their generalization ability. Furthermore, their difficulty in capturing complex interactions and long-term temporal dependencies hinders their adaptability and accuracy in dynamic traffic scenarios [20, 21].

In recent years, significant progress in vehicle trajectory prediction has been made through the application of deep learning methods [22–25]. These approaches leverage extensive data for training and learning, enabling comprehensive consideration of road environment structures and interactions among surrounding traffic participants. Originally employed for text translation, recurrent neural networks (RNNs) have subsequently been introduced into autonomous driving trajectory prediction tasks, facilitating the processing of sequential inputs and outputs. In [26], RNNs were proposed to extract features from historical trajectory data, followed by the use of graph neural networks (GNNs) for encoding to facilitate inter-vehicle perception and trajectory prediction [27, 28]. However, due to the necessity of continually storing historical information to address sequential inputs, RNNs encounter challenges such as vanishing or exploding gradients. Researchers addressed this issue by introducing gate mechanisms, with long short-term memory networks (LSTMs) emerging as the most notable solution. In [29], a personalized joint time sequences modeling approach based on LSTMs was proposed to efficiently predict the trajectories of preceding vehicles. Recognizing the strong spatiotemporal continuity inherent in trajectory prediction tasks, some researchers integrated convolutional neural networks (CNNs) into LSTMs to accurately capture spatiotemporal features within trajectory data [30]. This novel approach significantly improves the performance of vehicle trajectory prediction [14, 31]. Moreover, to enhance prediction accuracy, the attention mechanism is further integrated with LSTMs to focus on crucial information during the learning process [32]. In [33], Lin et al. effectively improved prediction precision by combining spatiotemporal attention mechanisms with LSTMs. However, they did not consider the impact of driving styles on vehicle trajectories. Building upon this foundation, Chen et al. [4] integrated the attention mechanism with the prediction network. They considered both driving styles and vehicle dynamics factors, thereby further enhancing prediction capabilities. However, it only considered simple lane change behaviors in high-speed environments.

In summary, deep learning-based prediction has enabled the effective capture of trajectory features in both temporal and spatial domains, resulting in enhanced prediction accuracy and robustness. One of the main challenges in trajectory prediction tasks is the diversity and uncertainty of vehicle driving behaviors, especially in complex traffic environments. The impact of different driving styles on trajectory prediction is significant, yet traditional methods often fail to fully account for these factors, resulting in lower prediction accuracy and adaptability. Moreover, vehicle trajectories are not only influenced by the current state but also by historical behaviors and interactions with surrounding traffic. Therefore, effectively extracting long-term dependency features and handling these complex interactions has become a critical issue in trajectory prediction. With all the above in mind, this paper proposes a new advanced vehicle trajectory prediction framework aiming at improving the accuracy of trajectory predictions in complex and dynamic traffic environments, as illustrated in Figure 1. The framework consists of a driving style classification module and a vehicle trajectory prediction module. It integrates the vehicle's historical behavior, driving style, and dynamic interactions with surrounding vehicles. The main contributions of this framework are briefly summarized as follows.

(1) First-time introducing the concept of 'soft clustering and hard labels', this paper applies the fuzzy c-means

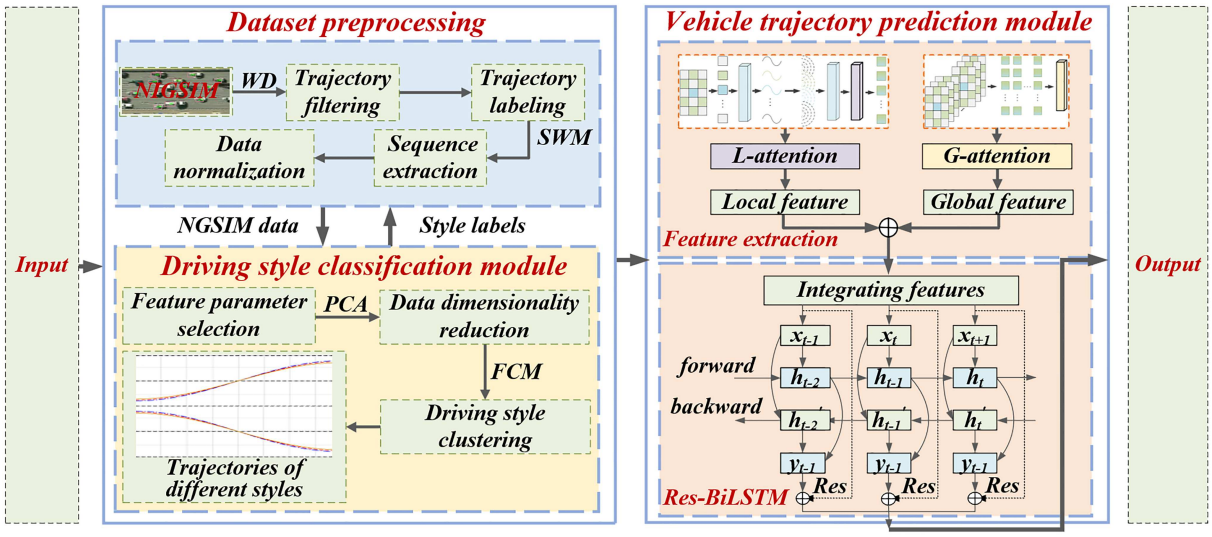

Figure 1 (Color online) Structure of the trajectory prediction framework. WD: wavelet denoising; SWM: sliding window method; PCA: principal component analysis; FCM: fuzzy c-means clustering; Res: residual structure.

(FCM) algorithm to classify diverse driving styles and transforms the clustering results into hard labels, integrating the fuzziness of soft clustering with the clarity of hard labels.

(2) To effectively extract long-term sequence features, this paper innovatively models vehicle behaviors as waveforms, which allows for capturing subtle variations in trajectories, unlike traditional trajectory modeling methods. Furthermore, a novel attention mechanism is designed to aggregate both global and local features, ensuring effective and appropriate utilization of the features.

(3) The trajectory prediction model in the proposed framework nicely integrates the LSTM encoder-decoder architecture with residual structures, thereby simplifying the training process and significantly enhancing prediction stability and accuracy.

2 Problem formulation

Trajectory prediction aims to forecast the spatial coordinates of various participants on the road (such as cars, animals, and pedestrians) over different periods, including short-term (1–3 s) and long-term (3–5 s) intervals [32]. Both the short-term and long-term predictions collectively form a complete trajectory prediction for the 1–5 s period. This problem can be viewed as a sequential task aimed at inferring the coordinate changes of the target object a time intervals.

In complex traffic environments, accurately predicting the future trajectory of a target vehicle requires leveraging surrounding environmental information while simultaneously considering the target vehicle's operational states. To ensure the effective extraction of latent features from real traffic data, input data should encompass parameters such as the target vehicle's type, operational status, and spatiotemporal relationships with surrounding obstacle vehicles. Therefore, this paper defines vehicle trajectory prediction as the problem of estimating a set of future states for the target vehicle. To simplify the computation, we focus on six obstacle vehicles whose longitudinal displacements fall within the range of $[-100, 100]$ m, as shown in Figure 2. Specifically, The input feature vector I_m^{tar} can be represented as follows:

$$\begin{aligned} I_m^{\text{tar}} &= \left(L, W, X_m^{\text{tar}}, Y_m^{\text{tar}}, V_m^{\text{tar}}, a_m^{\text{tar}}, D_{mn}^{\text{tar}}, T_{mn}^{\text{tar}}, S_m^{\text{tar}}\right), \\ m &= 1, 2, 3 \ldots, T_{\text{obs}}, \\ n &= 1, 2, 3 \ldots, 6, \end{aligned} \tag{1}$$

where I_m^{tar} is the input status of the target vehicle within the observation time domain T_{obs}, L and W represent the length and width of the predicted target vehicle, respectively. X_m^{tar} and Y_m^{tar} represent the lateral and longitudinal coordinates of the predicted target, and S_m^{tar} indicates its driving style label. V_m^{tar} and a_m^{tar} correspond to the velocity and acceleration of the predicted target. D_{mn}^{tar} represents the headway distance between the predicted target vehicle and the n-th obstacle vehicle, and T_{mn}^{tar} represents the headway time between them.

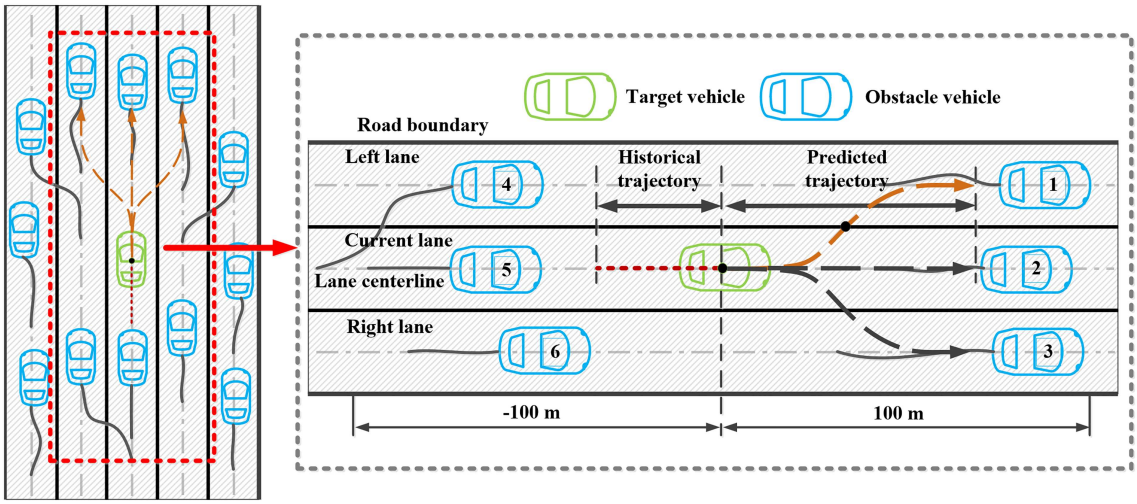

Figure 2 (Color online) Schematic diagram of the vehicle trajectory prediction scenario.

The output of the prediction is the future trajectory of the target vehicle within the prediction time domain:

$$O^{\text{tar}}_{T_{\text{obs}}+m} = \left(X^{\text{tar}}_{T_{\text{obs}}+m}, Y^{\text{tra}}_{T_{\text{obs}}+m}\right), \quad m = 1, 2, 3 \ldots, T_{\text{pred}}, \tag{2}$$

where $X_{T_{\text{obs}}+m}$ and $Y_{T_{\text{obs}}+m}$ denote the lateral and longitudinal coordinates of the predicted trajectory of the target vehicle from the current time to the prediction time domain T_{pred}.

By comprehensively considering the vehicle's physical attributes, environmental information, and driving habits, this paper transforms the problem of vehicle trajectory prediction into the estimation of future states for the target vehicle using the input from (1) to produce the output described in (2).

3 Driving style classification module based on fuzzy c-means

After extracting trajectory data from the dataset, this section categorizes the trajectories based on various driving styles. The original dataset typically contains numerous features, but not all contribute significantly to the model's predictive capability. By selecting the most relevant or representative features, the model's complexity can be reduced while improving its generalization ability. Therefore, this section applies principal component analysis (PCA) is employed to reduce the dataset's dimensionality [34]. This process helps the model better understand the data's structure and patterns.

The driving-behavior dataset was first processed, and 19 driving-behavior features were selected as shown in Table 1. These features cover four aspects of indicators: speed dynamics, lateral motion, acceleration characteristics, and following behavior. Next, PCA was applied to reduce the dimensionality of the data. The goal of PCA is to map the data from a high-dimensional space to a lower-dimensional space through linear transformation, extracting the main components of the data. For PCA analysis, the data were first standardized using the Z-score normalization method to normalize all features by the following formula:

$$X_{\text{norm}} = \frac{X - \lambda}{\sigma}, \tag{3}$$

where λ is the mean of the feature, and σ is the standard deviation. Through the KMO test (KMO = 0.86) and Bartlett's sphericity test ($\chi^2 = 632.7, p < 0.001$), it was confirmed that the data are suitable for PCA dimensionality reduction.

Through the eigendecomposition of the covariance matrix, the first five principal components listed in Table 2 were extracted. The cumulative variance contribution of these components reached 99.36%, far exceeding the empirical threshold of 90%. According to the Kaiser criterion, principal components with eigenvalues greater than 1 were retained, ensuring that the main information in the data was preserved while reducing redundant components.

It can be seen that PC1 primarily reflects the vehicle's longitudinal motion intensity, particularly during acceleration and deceleration, contributing 82.14% to the data variance. PC2 is related to the vehicle's lateral control stability, reflecting the driver's performance in steering and lane-keeping, with a variance contribution of 16.33%.

Table 1 Selection of driving behavior feature parameters. 'Maximum' is shortened to 'Max'. 'Average' is shortened to 'Avg'. 'Standard deviation' is shortened to 'Std dev'. 'Acceleration' is shortened to 'Acc'. 'Distance' is shortened to 'Dist'. 'Lateral' is shortened to 'Lat'.

No.	Parameter (unit)	No.	Parameter (unit)	No.	Parameter (unit)
1	Max speed (m/s)	8	Mean abs acc (m/s^2)	15	Std dev follow dist (m)
2	Avg speed (m/s)	9	Acc std dev (m/s^2)	16	Min headway time (s)
3	Speed std dev (m/s)	10	Max abs lat acc (m/s^2)	17	Avg headway time (s)
4	Max lat speed (m/s)	11	Avg abs lat acc (m/s^2)	18	Headway time std dev (s)
5	Mean lat speed (m/s)	12	Std dev lat acc (m/s^2)	19	Travel dist (m)
6	Lat speed std dev (m/s)	13	Min following dist (m)	–	–
7	Max abs acc (m/s^2)	14	Avg following dist (m)	–	–

Table 2 Physical interpretation and core loadings of principal components.

Principal component	High-load features ($\|r\| > 0.7$)	Variance contribution (%)
PC1	Max speed (0.92), Avg speed (0.89)	82.14
PC2	Std dev of lat acc (0.78)	16.33
PC3	Std dev of headway time (0.88)	0.92
PC4	Max abs acc (0.84)	0.53
PC5	Min headway time (0.77)	0.04

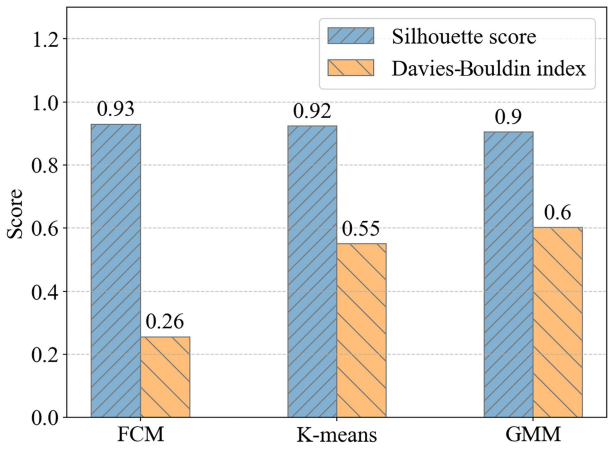

Figure 3 (Color online) Comparison of silhouette scores and the Davies-Bouldin index among the three algorithms.

PC3, PC4, and PC5 represent the dispersion of the following behavior, emergency braking tendency, and risk of tailgating, respectively. Although their variance contributions are small (0.92%, 0.53%, and 0.04%), they still provide effective information about driving style and should not be neglected.

Next, cluster analysis is conducted using the dimensionality-reduced data. However, driving style represents a fuzzy concept where boundaries between different driving behaviors are unclear. Traditional hard clustering algorithms, such as k-means [35] and Gaussian mixture models (GMM) [4], face challenges in handling the uncertainty of driving styles. In contrast, the fuzzy c-means (FCM) clustering algorithm [36] allows trajectories to belong to multiple clusters simultaneously. It reflects the probability of a trajectory belonging to different categories through membership degrees. To compare the clustering performance of FCM, k-means, and GMM for driving style classification tasks, this paper sets the number of driving style categories to three [4]. The silhouette score [37] and Davies-Bouldin index [38] are used as evaluation metrics. The evaluation results are shown in Figure 3.

The silhouette score close to the value of 1 indicates better clustering performance, with high similarity within clusters and significant differences between clusters. A lower Davies-Bouldin index signifies better clustering quality and stronger separation between clusters. FCM outperforms both k-means and GMM in these metrics, indicating that it generates the most distinct and well-separated clusters. While k-means also shows good clustering performance, it does not achieve the effectiveness of FCM. GMM performs the worst, reflecting the poorest clustering results. Therefore, FCM is the most suitable method for driving style classification tasks, offering the best cluster quality and separation among the three methods.

In the FCM process, fuzzy cluster centers are determined by minimizing an objective function, which can be

described as

$$J_M^{\mathrm{FCM}} = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^M |x_i - c_j|_2^2 = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^M \sqrt{\sum_{k=1}^{d}(x_{ik} - c_{jk})^2}, \tag{4}$$

where M is the fuzzy coefficient that controls the degree of fuzziness in the clustering process, affecting how membership values are assigned to each driving style for the trajectory samples ($1 \leqslant M < \infty$). N denotes the number of vehicle trajectory samples, and C stands for the number of driving style categories, which equates to the number of cluster centers. c_j represents the j-th cluster center, and u_{ij} indicates the probability that sample x_i belongs to cluster center c_j. d represents the dimensionality of the samples, x_{ik} denotes the value of sample x_i in the k-th dimension, and c_{jk} represents the value of cluster center c_j in the k-th dimension. $|\cdot|_2$ represents the Euclidean norm used for measuring the distance between driving samples and cluster centers.

The fuzzy partition coefficient (FPC) is an important metric in fuzzy clustering algorithms used to evaluate the quality of clustering results. It helps determine whether the chosen number of clusters C is appropriate and can be used to select the optimal number of clusters. By calculating the FPC values for different numbers of clusters C, the effectiveness of the clustering can be assessed, allowing the selection of the number of clusters that optimizes the FPC value. The FPC is obtained by averaging the maximum membership degrees of all data points

$$\mathrm{FPC} = \frac{1}{n}\sum_{i=1}^{n} u_{ij}^{\max} \tag{5}$$

in the given equation, i represents the index of data points in the dataset. Specifically, it is used to summarize data points, from 1 to n, where n is the total number of data points. The FPC value closer to 1 indicates a better clustering effectiveness.

Next, the algorithm iteratively obtains the membership distribution of each sample across various driving styles, achieving soft clustering of driving styles. The termination conditions for iteration are as follows:

$$\max_{ij}\left\{\left|u_{ij}^{(t+1)} - u_{ij}^{(t)}\right|\right\} < \varepsilon, \tag{6}$$

where t represents the number of iterations, ε is a small constant indicating the error threshold. In other words, iteration stops when the maximum change between consecutive iterations of u_{ij} does not exceed the ε.

Then, to effectively utilize the results of soft clustering, this paper introduces the concept of 'soft clustering to hard labels'. Specifically, the FCM algorithm is first used to compute the membership degree of each data point across all clusters. Next, based on the membership degrees of each data point in all clusters, the cluster with the highest membership degree is selected as the final assigned cluster for the data point. By doing so, the data point is assigned to the cluster center with the highest membership degree, effectively converting soft clustering into hard labels. The conversion process can be expressed by the following equation:

$$\mathrm{label}(x_i) = \arg\max_{c_j} u_{ij}, \tag{7}$$

where $\mathrm{label}(x_i)$ denotes the hard label of sample x_i. This approach not only preserves the flexibility of soft clustering but also translates it into clear hard labels. This approach provides more precise personalized driving style information, thereby enhancing the accuracy and applicability of the vehicle trajectory prediction framework.

4 Global-local enhanced attention module with residual bidirectional long short-term memory network

Utilizing the labels obtained from the driving style classification module, the vehicle trajectory prediction module benefits from precise personalized feature inputs. This section will design a new network architecture for vehicle trajectory prediction called GLEA-ResBiLSTM, as shown in Figure 4. This architecture integrates global and local attention mechanisms to provide richer feature information when capturing complex in-sequence data. Furthermore, residual connections are introduced into the BiLSTM network, enhancing the stability and training efficiency of the prediction model by preserving the original input information.

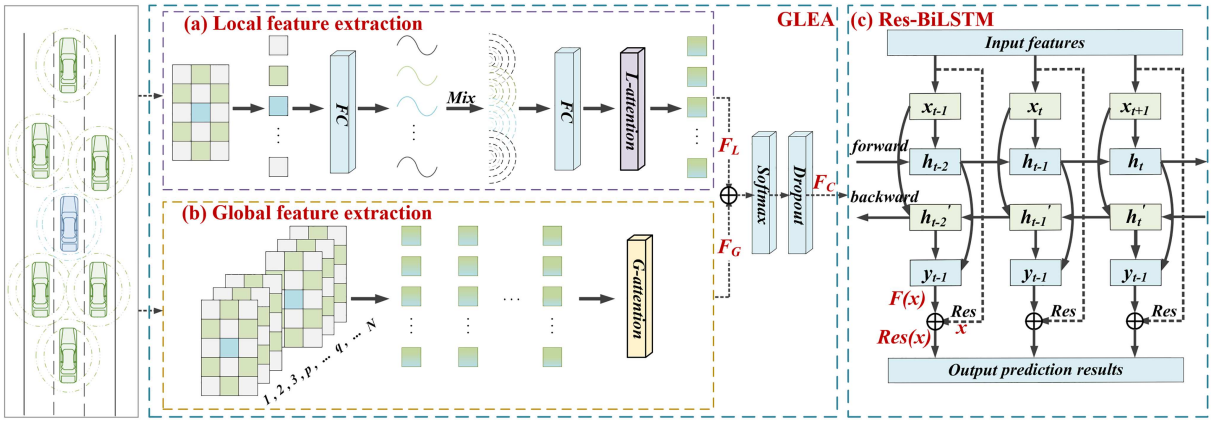

Figure 4 (Color online) The diagram of GLEA-ResBiLSTM. (a) Local feature extraction module; (b) global feature extraction module; (c) Res-BiLSTM vehicle trajectory prediction network.

4.1 Global-local enhanced attention module

To further enhance the accuracy of trajectory prediction, this section introduces a novel global-local enhanced attention module (GLEA). The module comprises three components: local feature extraction, global feature extraction, and feature integration. Global feature extraction captures overall feature relationships within the entire traffic scene, such as traffic flow trends or the influence of distant vehicles, aiding in trajectory predictions over longer time frames. In contrast, local feature extraction focuses on small-scale changes, quickly responding to sudden events like speed variations or emergency lane changes, helping the model adapt to real-time dynamic environments. Feature extraction is primarily achieved through collaboration between the global attention module (G-attention) and the local attention module (L-attention). The G-attention module focuses on capturing feature relationships across the entire scene, and the L-attention module specializes in capturing detailed interactions among nearby vehicles. By integrating these features, the model achieves a comprehensive understanding of the overall traffic scene and accurately captures subtle variations between individual vehicles.

Local feature extraction is performed first, as shown in Figure 4(a). According to the principles of quantum mechanics, all matter exhibits wave-like behavior, which is a phenomenon known as matter wave [39, 40]. Inspired by this, this section models every vehicle in complex traffic scenarios as a wave source with specific amplitude and phase. Each vehicle generates a wave based on its motion state (speed, acceleration, position). The speed of the vehicle determines the wave's amplitude, with higher speeds resulting in stronger amplitudes. Acceleration affects the frequency of the wave, a higher acceleration corresponds to a higher frequency (i.e., the rate of phase change), making the wave's variation more pronounced. The vehicle's position determines the phase of the wave, reflecting its relative position within the traffic flow. The waves generated by their motion can superimpose with those from other vehicles, thereby transforming the complex traffic situation into the superposition effect of waves.

When waves from two or more vehicles intersect at the same point, they adhere to the principle of wave superposition. The superposition effect of vehicle waves at this point is represented by the combined vector sum of their respective displacements. This effect can be expressed by the following formula:

$$\tilde{z}_h = |z_h| \odot \mathrm{e}^{\mathrm{i}_m \theta_h}, h = 1, 2, \ldots, n, \tag{8}$$

where i_m is the imaginary unit. $|\cdot|$ denotes the absolute value operation, and $\odot$ denotes element-wise multiplication. $|z_h|$ represents the amplitude, and θ_h represents the phase. For vehicle waves, differences in vehicle speeds affect the amplitude of the waves and the phase of the waves signifies the position of the vehicles. When two vehicle waves exhibit identical displacement in the same direction at a certain point, constructive interference occurs with a phase difference of

$$\theta_h = \theta_{\mathrm{i}_m} + 2\eta\pi, \ \eta \in [0, \pm 2, \pm 4, \ldots]. \tag{9}$$

Conversely, when two waves have opposite displacements at a certain point, the destructive interference takes place with a phase difference of

$$\theta_h = \theta_{\mathrm{i}_m} + \eta\pi, \ \eta \in [0, \pm 1, \pm 3, \ldots]. \tag{10}$$

Therefore, when interference occurs between vehicles with different speeds, the superposition effect of waves will reflect how waves with different amplitudes and phases interfere with each other. By analyzing the superposition

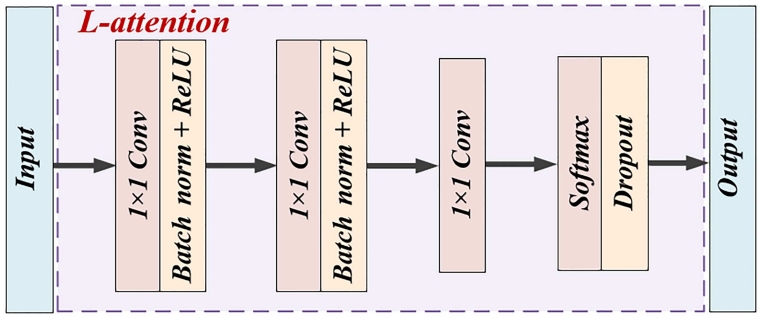

Figure 5 (Color online) Diagram of the L-attention mechanism.

effect of vehicle waves, one can infer the interactions between different vehicles and further comprehend the dynamic behavior in traffic scenarios. The amplitude z_h and phase θ_h of vehicle waves are determined using the following equations:

$$\begin{cases} z_h = \mathrm{FC}(f_g, \omega_z), \\ \theta_h = \mathrm{FC}(f_g, \omega_\theta), \\ h = 1, 2, 3 \ldots, n, \end{cases} \tag{11}$$

where 'FC' denotes a standard fully connected layer and ω_z and ω_θ represent learning parameters. These parameters can be adjusted based on input features f_g to acquire appropriate weights for obtaining amplitude and phase embeddings. Referring to (8), the phase and amplitude of vehicle waves are represented in the complex domain

$$\tilde{z}_h = |\boldsymbol{z}_h| \odot \cos\theta_h + i|\boldsymbol{z}_h| \odot \sin\theta_h, \ h = 1, 2, \ldots, n. \tag{12}$$

For the interaction among all vehicles in a driving scenario, n vehicle waves are represented as $\tilde{Z} = [\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{n-1}, \tilde{z}_n]$, and the aggregation among these vehicle waves can be represented as

$$\tilde{\boldsymbol{o}}_h = \mathrm{FC}(\tilde{Z}, \omega^o)_h, \ h = 1, 2, \ldots, n, \tag{13}$$

where $\tilde{\boldsymbol{o}}_h$ represents the complex domain representation after aggregating vehicle waves. Real-valued output $\boldsymbol{o}_h$ is obtained through weighted summation:

$$\boldsymbol{o}_h = \sum_q \omega^o_{hq} \boldsymbol{z}_q \odot \cos\theta_q + \omega^s_{hq} \boldsymbol{z}_q \odot \sin\theta_q, \quad h = 1, 2, \ldots, n, \tag{14}$$

where ω^o, ω^s are both learning weights.

To enhance these local features, this paper designs the L-attention mechanism, as shown in Figure 5. The L-attention mechanism first applies a 1×1 convolution to the input features to perform the initial dimensional transformation. Then, using an adaptive convolution layer and weighting mechanism, the feature map is element-wise multiplied by the corresponding attention weights, thereby enhancing the local features. To generate the attention weights α_t, the similarity between the current time step feature x_t and its adjacent time step features x_{t-1} and x_{t+1} is calculated. The attention weight for the local feature α_t can be expressed as

$$\alpha_t = \frac{\exp\left(\mathrm{sigmoid}\left(f\left(x_t, x_{t-1}, x_{t+1}\right)\right)\right)}{\sum_{t=1}^{T} \exp\left(\mathrm{sigmoid}\left(f\left(x_t, x_{t-1}, x_{t+1}\right)\right)\right)}, \tag{15}$$

where $f(x_t, x_{t-1}, x_{t+1})$ is used to calculate the similarity between the features of time step t and its adjacent time steps, and the sigmoid function provides a nonlinear mapping. The normalization ensures the sum of weights is 1. Finally, the feature map is weighted by these attention weights to obtain the enhanced local feature representation F_L.

As shown in Figure 4(b), the global feature extraction occurs simultaneously with local feature processing. This module integrates the features from each local time window, aiming to capture global structural information by considering the relationship between the current time step and all other time steps. For a given time step p and q, the input feature tensors T_p and T_q are mapped to a shared feature space using a feature mapping function $f(\cdot)$ and a fully connected layer

$$r_{p,q} = f(T_p, T_q). \tag{16}$$

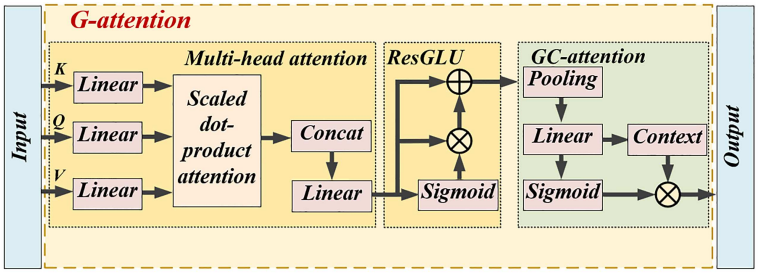

Figure 6 (Color online) Diagram of the G-attention mechanism.

Subsequently, using $r_{p,q}$ to represent the relationship between the features at time step p and time step q. For the feature T_p at time step p, its relationship vector $\boldsymbol{r}_p$ is defined as

$$\boldsymbol{r}_p = [r_{p,1}, r_{p,2}, \ldots, r_{p,q}, r_{p,N}, r_{1,p}, r_{2,p}, \ldots, , r_{q,p}, r_{N,p}]. \tag{17}$$

Then, combine the time step features with the relationship vectors to form paired representations. The global features are expressed as $F_G^p = [T_p, \boldsymbol{r}_p]$. Similarly, the global features at time step q can also be obtained.

The G-attention mechanism illustrated in Figure 6 is used to enhance the extracted global features, integrating the strengths of the improved multi-head self-attention (ResGLU-attention) and the global context attention (GC-attention). It is noted that the new ResGLU-attention mechanism introduces gating mechanisms (GLU) and residual connections. By activating the GLU, the network can selectively enhance the input features or diminish their bad influence. Moreover, residual connections facilitate efficient information transmission and integration across different attention heads. By concurrently processing multiple attention heads, the model effectively captures feature relationships at each time step within the time series. The GC-attention aggregates global contextual information and then provides a comprehensive understanding of interdependencies among features across all-time series.

During the weight parameter optimization process, all weights in the attention module are optimized using the backpropagation algorithm. By computing the gradients of the loss function with respect to the weight parameters and using the Adam optimization algorithm to update the weights, the model ensures efficient and accurate learning of the important features.

Finally, the global and local features are integrated as follows:

$$F_C = F_L + F_G, \tag{18}$$

where F_C represents the results of integrating global and local features. F_L represents the results of local feature extraction. F_G is the result of global feature extraction.

4.2 The design of residual bidirectional long short-term memory network

After acquiring rich global and local features, this section will design a network named residual bidirectional long short-term memory network (Res-BiLSTM) to utilize these features. This network builds upon the foundational framework of LSTM. It is well-known that LSTM is renowned for its widespread use in vehicle trajectory prediction tasks, characterized by memory cells and three types of gates: forget gate, input gate, and output gate. The specific architecture of LSTM is depicted in Figure 7. The relationships among variables within LSTM units are described by the following equations:

$$\begin{cases} f_t = \sigma(W_f\,[h_{t-1}, X_t] + b_f), \\ i_t = \sigma(W_i\,[h_{t-1}, X_t] + b_i), \\ \tilde{C}_t = \tanh\left(W_{\tilde{c}}\,[h_{t-1}, X_t] + b_{\tilde{c}}\right), \\ o_t = \sigma(W_o\,[h_{t-1}, X_t] + b_o), \\ C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ h_t = o_t \cdot \tanh\left(C_t\right), \end{cases} \tag{19}$$

where σ is the sigmoid activation function, and tanh is the hyperbolic tangent activation function. h_{t-1} is the hidden state vector at time step $t-1$, containing the accumulated memory and information from previous time steps. X_t is the input vector at the current time step t, derived from F_C. The symbol '$\cdot$' denotes element-wise multiplication. The weight matrices W_f, W_i, $W_{\tilde{c}}$, W_o and their corresponding bias terms b_f, b_i, $b_{\tilde{c}}$, b_o are used for the calculations of the forget gate, input gate, candidate cell state, and output gate, respectively.

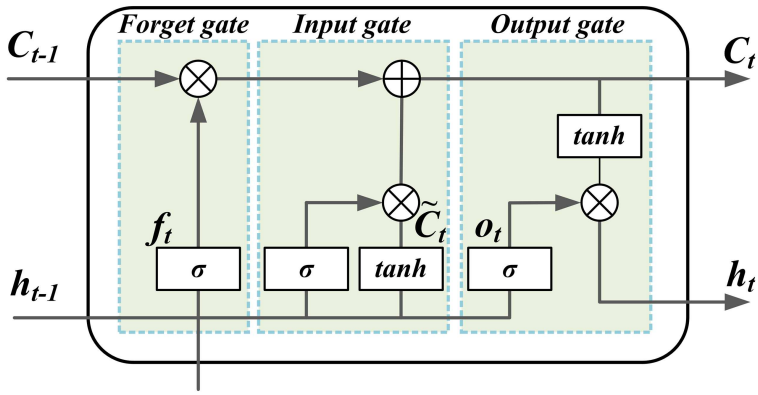

Figure 7 (Color online) Long short-term memory unit.

Compared to the LSTM network, the BiLSTM network incorporates an additional backward recurrent layer, enabling the model to access both forward and backward time steps of sequential data simultaneously. This bidirectional recurrent structure effectively integrates past and future information, allowing the model to more accurately capture long-term dependencies within sequence data. In this study, a residual structure is introduced into the BiLSTM network, designed as depicted in Figure 4(c) as the Res-BiLSTM network. It utilizes skip connections to construct residual blocks, enabling the network to leverage both the original input data from the previous layer and residual information concurrently, thereby enhancing model performance. The computational formula for a residual block is as follows:

$$\mathrm{Res}(x) = F(x) + x, \tag{20}$$

where x is the input sequence, and $F(x)$ is the output processed by the network. Finally, the Res-BiLSTM network delivers more accurate vehicle trajectory predictions based on rich global and local features.

5 Data pre-processing and label annotation results

After constructing the framework in this paper, an appropriate dataset will be selected and preprocessed to enable effective input into the framework for experiments.

5.1 Dataset selection

The NGSIM dataset [41] is a publicly available traffic safety dataset initiated by the U.S. Federal Highway Administration. Widely adopted by scholars in the field of transportation, it is utilized for research on driving behaviors such as vehicle following and lane-changing, vehicle motion trajectory prediction, and decision-planning tasks. The dataset has a collection frequency of 10 Hz. Based on the research scenarios, the datasets for the US101 and I-80 segments were selected and exported in .csv format for subsequent use.

5.2 Data preparations

Due to some errors in the data collection process of the NGSIM dataset, especially for lateral speed errors. It is necessary to employ filtering algorithms for data smoothing. This study utilizes a five-iteration wavelet transform denoising algorithm to effectively filter noise in the trajectory data from the original dataset. Next, we will balance the denoised dataset based on trajectory labels and annotate it with driving style labels, aiming to enhance the framework's adaptability to diverse driving behaviors.

5.2.1 *Trajectory labeling*

The NGSIM dataset primarily focuses on lane-keeping trajectories, with relatively fewer trajectories for left and right lane changes. It is crucial to balance the dataset through trajectory label annotation. In the domain of trajectory annotation tasks, various factors such as position information and turn signals are used in [42, 43]. In this study, we adopt the heading angle θ_{head} as the metric for trajectory labeling as follows:

$$\theta_{\mathrm{head}} = \arctan \frac{x_{t+1} - x_t}{y_{t+1} - y_t}, \tag{21}$$

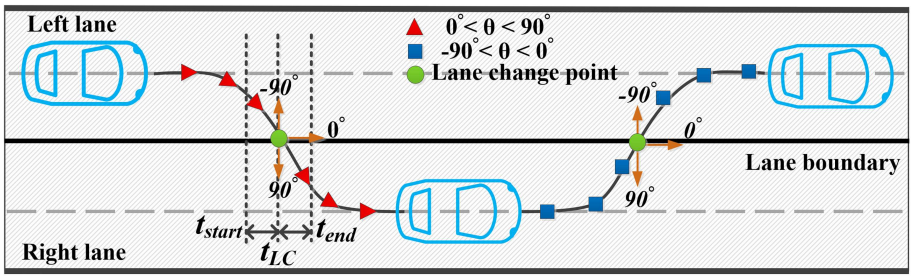

Figure 8 (Color online) Vehicle trajectory labeling diagram.

Algorithm 1 NGSIM dataset trajectory labeling algorithm.

Input: NGSIM dataset.
Output: Trajectory labels.
for each vehicle ID in the dataset **do**
  Update vehicle ID;
  Determine lane change points based on lane ID changes;
  Calculate heading angle θ_{head} for the changing vehicle using (21);
  Sample continuously to obtain heading angles;
  if for three consecutive samples, $0^\circ < \theta_{\text{head}} < 90^\circ$ **then**
    Label as a right lane change;
  else if for three consecutive samples, $-90^\circ < \theta_{\text{head}} < 0^\circ$ **then**
    Label as a left lane change;
  else
    Label as lane-keeping;
  end if
end for

Table 3 Balanced dataset results.

Label categories	Left lane change	Lane-keeping	Right lane change
Original quantity	149204	1205647	38827
Balanced quantity	38827	38827	38827

where x and y represent the lateral and longitudinal positions of the trajectory points. t and $t+1$ denote the current and the next time steps, respectively.

In this study, the lane-changing process is considered as a continuous action of crossing lane boundaries. Firstly, we filter the target vehicle's trajectory characteristics by its vehicle ID. As illustrated in Figure 8, the lane changing is determined by the lane ID and defines the point t_{LC} as where the trajectory crosses the lane boundary. Subsequently, the heading angles are computed at these intersection points. If the heading angles of three consecutive points fall between 0 and 90 degrees, the first point satisfying this condition is marked as the starting point t_{start} of a right lane change, and the ending point t_{end} is similarly determined. All trajectory points between t_{start} and t_{end} are labeled as part of a right lane change. Similarly, labels are assigned for left lane changes. Points not identified as lane changes are labeled as lane-keeping [44]. The trajectory labeling process for the NGSIM dataset is outlined in Algorithm 1. To meet the model's requirement for data sequence length, a sliding window method was utilized to process trajectory sequences in this study. Considering that this model predicts future trajectories based on the preceding 3 s of trajectory data for the next 5 s, each trajectory segment is labeled according to the marked status at the time of 3 s.

The final labeled dataset is presented in the first row of Table 3, there is a relatively small number of samples for right lane changes. To ensure data balance, random sampling was performed on lane-keeping and left lane change data to equalize the sample size with that of right lane change data. The resulting balanced dataset is presented in the second row of Table 3.

5.2.2 *Driving style labeling*

The driving style classification module will then be used to extract driving style information for each vehicle. The main objective of this module is to cluster the data into different categories and evaluate them using the FPC mentioned in Section 3. This study evaluates how different number of clusters affects clustering performance. The results show that when the number of clusters is 3, the FPC value reaches its highest (0.9851), surpassing other cluster numbers (2, 4, 5, and 6), with corresponding FPC values of 0.9821, 0.9823, 0.7659, and 0.7628, respectively.

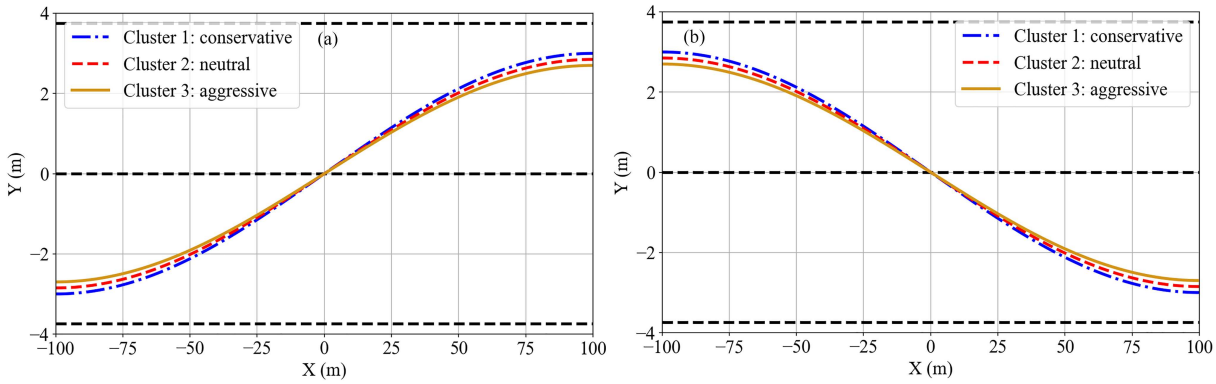

Figure 9 (Color online) Comparison of different types of driver lane-change trajectories. (a) Right lane-change trajectory; (b) left lane-change trajectory.

It is evident from the results that clustering achieves optimal performance with three categories. To assign precise style labels, each data point is allocated to the cluster center corresponding to its maximum membership degree. Transforming fuzzy clustering results into precise hard labels using the concept of 'soft clustering and hard labels', which are subsequently recorded in the dataset.

Figure 9 illustrates the comparison of the average lane-change trajectories from three different driving style categories. The trajectory curve of cluster 1 is relatively smooth with a small lane-change angle, categorizing it as a conservative driving style. Cluster 2 exhibits a moderate lane-change angle and medium trajectory length, characterized by a neutral driving style. Cluster 3 demonstrates rapid lane-change actions and a short trajectory length, identified as an aggressive driving style.

Finally, the data was divided into training, testing, and validation sets by a 6:2:2 ratio. To enhance the model's predictive ability and training efficiency, data differencing and feature scaling regularization methods were both employed to reduce noise and fluctuations in the sequence.

6 Experiments and analysis

Experiments using the processed dataset will be conducted to validate the framework's predictive capability. In the following, we will comprehensively outline the experimental setup of this study, including the selection of model parameters and the definition of evaluation metrics. Both the comparative and ablation experimental results are also presented, together with their analysis and discussion.

6.1 Experimental setup

6.1.1 *Experimental details*

The experiments were conducted on an NVIDIA GeForce RTX 4060 Ti GPU, using the PyTorch deep learning framework. It involved 500 training iterations with a batch size of 1024. The mean squared error (MSE) was used as the loss function, with the Adam optimizer for model training. Hyperparameter tuning was performed using grid search in combination with cross-validation to enhance model performance. The grid search evaluated various hyperparameter configurations, selecting the optimal ones based on MSE results from the validation set. This process identified the best hyperparameters: a learning rate of 0.001 and a weight decay of 0.0001. Additionally, both the encoder and decoder utilized Res-BiLSTM modules with identical architectures, consisting of 2 layers, a hidden dimension of 128, and a dropout rate of 0.2.

6.1.2 *Evaluation metrics*

This paper uses a 3-s historical trajectory to predict the trajectory for the next 5 s, with the evaluation metric being the root mean square error (RMSE) of the predicted trajectory from 1 to 5 s. RMSE assesses the prediction accuracy of the entire trajectory sequence by calculating the square root of the mean square error between the predicted and the actual values, reflecting the overall accuracy of the model. The formula for RMSE is as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)}, \tag{22}$$

Table 4 Comparison results of different methods. G-RB is the proposed GLEA-ResBiLSTM. The minimum values of different predicted periods are marked in bold.

Model	Predicted endpoint root mean square error (m)				
	1 s	2 s	3 s	4 s	5 s
GSTCN	0.42	0.81	1.29	1.97	2.95
STDAN	0.42	1.01	1.69	2.56	3.57
FHIFN	0.40	0.98	1.66	2.52	3.63
IHN	0.45	0.81	1.11	1.50	1.86
MTF-LSTM	0.47	0.85	1.05	1.09	1.22
IMM	**0.07**	**0.22**	0.46	0.66	0.85
G-RB	0.20	0.29	**0.38**	**0.47**	**0.56**

where n denotes the number of samples, (x_i, y_i) are the coordinates of the i-th point of the true trajectory, $(\hat{x}_i, \hat{y}_i)$ are the coordinates of the i-th point of the predicted trajectory.

6.2 Comparative experiments

Next, to assess the performance of the proposed model, we will compare it with several advanced models proposed in recent years. These models include the following.

(1) **GSTCN** (graph spatial-temporal convolutional network). This approach combines graph convolutional network (GCN) and CNN for temporal features [45].

(2) **STDAN** (space-time dynamic attention network). This method utilizes self-attention for multi-dimensional feature extraction, capturing both temporal and spatial interaction features [46].

(3) **FHIFN** (fusion of historical interactive features network). This approach employs a multi-head attention mechanism for interaction feature encoding between target and neighboring vehicles [19].

(4) **IHN** (intent-aware hybrid convolutional and attention network). This method utilizes the intent convolutional social pooling module and the hybrid attention mechanism module to consider the periodicity of trajectories influenced by road geometry constraints [47].

(5) **MTF-LSTM** (mixed teaching force long short-term memory network). A hybrid imitation learning decoding method for vehicle trajectory prediction based on LSTM networks [44].

(6) **IMM** (interactive multi-model). This approach combines LSTM behavior recognition with Gaussian process motion modeling for precise lane change trajectory prediction [4].

(7) **GLEA-ResBiLSTM** (residual bidirectional long short-term memory network). The vehicle trajectory prediction model is proposed in this paper.

The comparison results are illustrated in Table 4. It can be seen that the GLEA-ResBiLSTM model demonstrates outstanding prediction capability, particularly in medium to long-term predictions. The RMSE results of the seven methods are shown in Figure 10(a). In terms of medium to long-term predictions, GSTCN, STDAN, and FHIFN methods have shown significant errors, with RMSE values of 2.95, 3.57, and 3.63 m, respectively, for the 5-s prediction. In comparison, the IHN and MTF-LSTM methods achieve much better performance. While the IMM method shows a slight advantage in the first 2-s prediction (with an RMSE of less than 0.22 m), the proposed framework demonstrates very stable prediction accuracy in the short term. As the prediction horizon extends, the prediction accuracy of the proposed GLEA-ResBiLSTM framework becomes increasingly evident, ultimately keeping the 5-s RMSE within 0.56 m. This result clearly illustrates the robustness of the proposed GLEA-ResBiLSTM framework in handling trajectory prediction for complex scenarios.

To further visually demonstrate the predictive performance of the GLEA-ResBiLSTM model across different driving behaviors, we selected trajectory samples of various driving behaviors from the balanced dataset as shown in Table 3. There are the prediction results for left lane change (Figure 10(b)), right lane change (Figure 10(c)), and lane keeping behaviors (Figure 10(d)). It can be observed again that the model proposed in this paper demonstrates excellent performance in predicting these behaviors, exhibiting high accuracy and stability. It is worth noting that the slight sawtooth pattern observed in the historical trajectory in Figure 10(c) is a result of vehicle movement in dynamic environments, especially for high-speed lane changes. However, such a small magnitude of fluctuation in X and Y positions does not affect the overall accuracy of trajectory prediction.

6.3 Ablation experiments

To evaluate the effectiveness of the model structure proposed in this paper, we have also conducted ablation experiments to assess the individual contributions of various components to the overall performance. These results

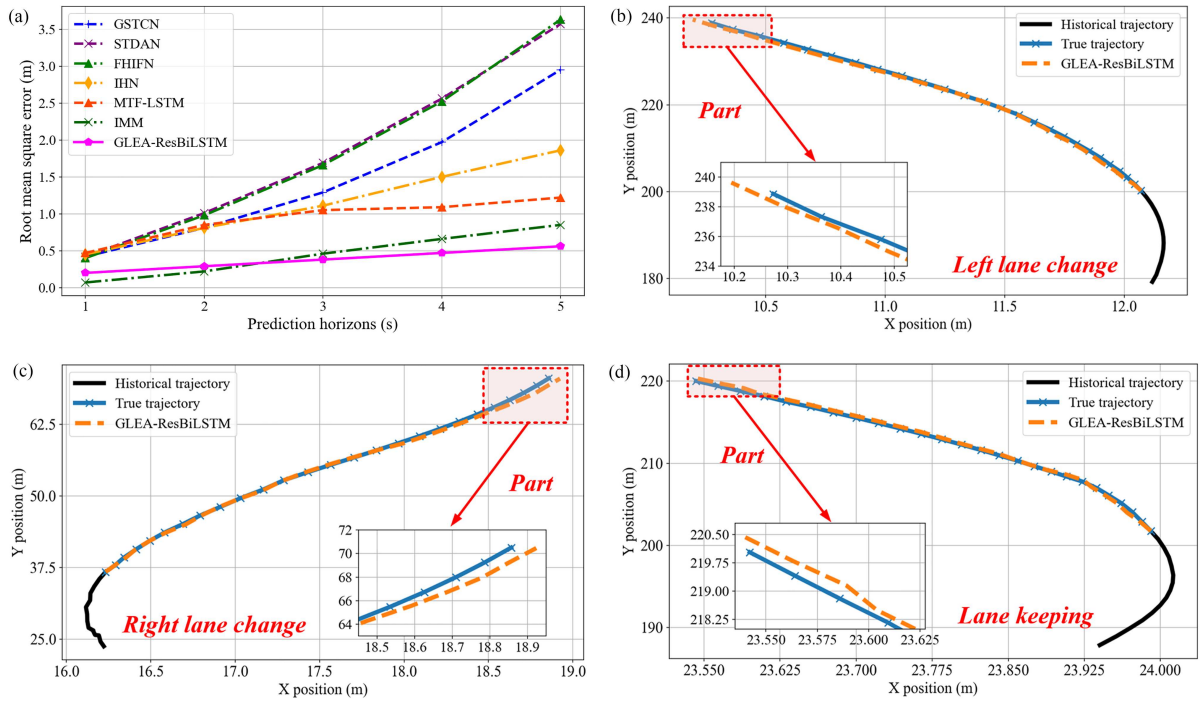

Figure 10 (Color online) Results of the comparison and prediction experiments. (a) The results of RMSE comparison for each algorithm; (b) the prediction results for left lane change trajectories; (c) the prediction results for right lane change trajectories; (d) the prediction results for lane keeping trajectories.

Table 5 Comparison results of ablation experiments.

Model	Predicted endpoint root mean square error (m)				
	1 s	2 s	3 s	4 s	5 s
Baseline	0.20	0.29	0.38	0.47	0.56
Nostyle	0.39	0.61	0.77	0.88	1.02
Noattention	0.23	0.35	0.46	0.58	0.69
Noresidual	0.61	0.96	1.29	1.81	2.63

will provide deeper insights into the significance of different model elements in the trajectory prediction task and their impact on predictive performance.

(1) **Baseline.** The complete model incorporating all components was trained and evaluated, serving as the baseline for subsequent comparisons.

(2) **Nostyle.** The driving style labels were removed, and the model was retrained to assess the significance of driving style labels in trajectory prediction tasks.

(3) **Noattention.** The attention mechanism module was removed, and the model was retrained to validate the role of attention mechanisms in capturing vehicle interactions.

(4) **Noresidual.** The residual connection structure was removed, and the model was retrained to verify the contribution of residual connections to model training and feature capture.

Table 5 presents the results of ablation experiments, validating the effectiveness of the proposed model. The baseline model performs best across the 5-s prediction window, with predicted endpoint RMSE of 0.20, 0.29, 0.38, 0.47 and 0.56 m for every 1 s interval. As shown in Figure 11, removing driver-style labels leads to much larger prediction errors, indicating the importance of driver-style information in prediction accuracy. Similarly, although the removal of the attention mechanism results in a relatively small increase in prediction errors, it still emphasizes its role in capturing vehicle features. Furthermore, removing residual connections has the most significant impact on model performance. This is because residual connections effectively preserve the original information, enabling the network to accurately learn key trajectory features even as its depth increases.

Figures 12(a) and (b) show the predicted results for a lane-keeping sample. It is evident that each module plays an indispensable role in improving model performance. All the elements as a whole contribute to the development of an efficient, stable, and accurate trajectory prediction framework.

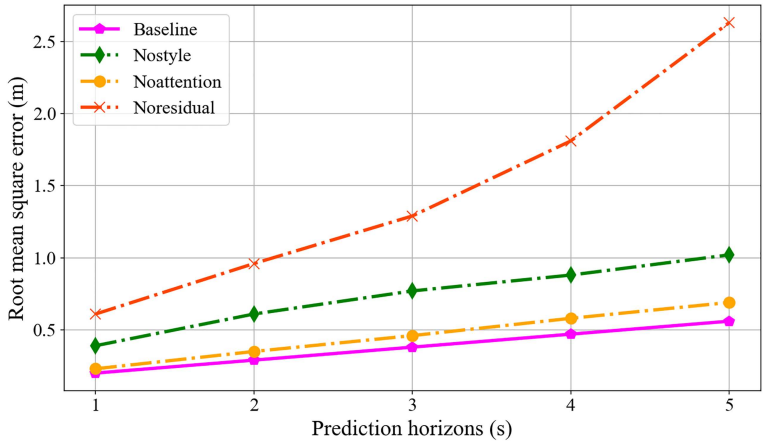

Figure 11 (Color online) RMSE comparison results of the ablation experiments.

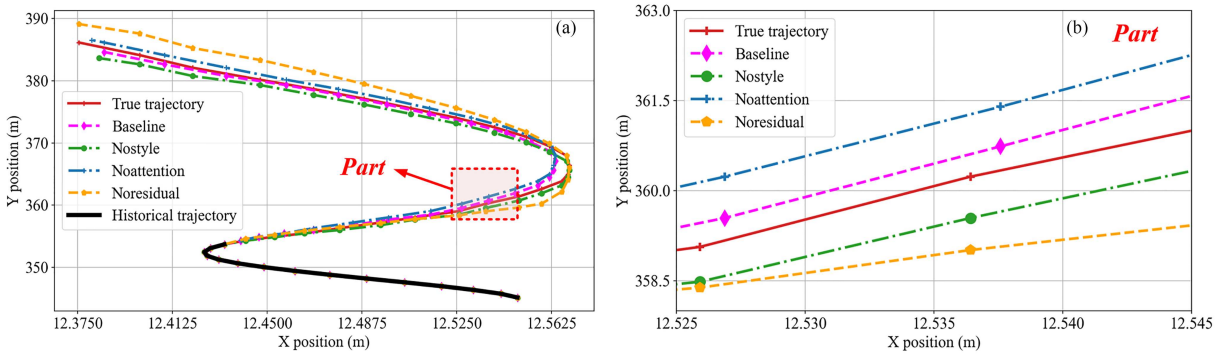

Figure 12 (Color online) Prediction results of ablation experiment. (a) The trajectory prediction ablation experiment results; (b) the local magnification results of the ablation experiment.

7 Conclusion and future work

With integrated consideration of driver styles and environmental uncertainties, this paper proposes a personalized trajectory prediction framework to enhance the safety and reliability of autonomous vehicles in complex traffic scenarios. To accommodate diverse driving behaviors, a flexible and accurate driver-style classification algorithm is initially designed to capture personalized behavioral characteristics effectively. In terms of model architecture, innovative integration of BiLSTM with residual structures resolves the gradient vanishing issue in deep networks, thereby improving model stability and accuracy. Moreover, the proposed GLEA-ResBiLSTM network leverages the complementary advantages of global and local attention mechanisms, further enhancing prediction performance. Experimental results demonstrate a significant reduction in root mean square error for 5 s prediction tasks, indicating higher prediction accuracy and stability of the proposed prediction framework.

Future research will focus on further enhancing the model's adaptability to complex traffic scenarios in daily driving. To achieve this, we plan to expand the dataset to include urban roads, highways, intersections, and pedestrian-dense areas to comprehensively evaluate the model's general robustness. Moreover, efforts will be directed toward optimizing the model's training and evaluation strategies to enhance its reliability under complex traffic scenarios.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62388101, 62303011, 62303010, 62273065), Anhui Provincial Key Research Program of Universities (Grant No. 2022AH050087), and University Synergy Innovation Program of Anhui Province (Grant No. GXXT-2023-039).

References

1 Liu J, Wu C, Le Vine S, et al. Panel data analysis of Chinese households' car ownership and expenditure patterns. Transp Res Part D-Transp Environ, 2023, 123: 103915
2 Qin Y, Wang H. Stabilizing mixed cooperative adaptive cruise control traffic flow to balance capacity using car-following model. J Intell Transp Syst, 2023, 27: 57–79
3 Huang Y, Du J, Yang Z, et al. A survey on trajectory-prediction methods for autonomous driving. IEEE Trans Intell Veh, 2022, 7: 652–674
4 Chen G, Gao Z, Hua M, et al. Lane change trajectory prediction considering driving style uncertainty for autonomous vehicles. Mech Syst Signal Process, 2024, 206: 110854

5 Huang Z, Wu J, Lv C. Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning. IEEE Trans Intell Transp Syst, 2021, 23: 10239–10251
6 Ju C, Wang Z, Long C, et al. Interaction-aware Kalman neural networks for trajectory prediction. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Las Vegas, 2020. 1793–1800
7 Abbas M T, Jibran M A, Afaq M, et al. An adaptive approach to vehicle trajectory prediction using multimodel Kalman filter. Trans Emerging Tel Tech, 2020, 31: e3734
8 Gao H, Qin Y, Hu C, et al. An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario. IEEE Trans Neural Netw Learn Syst, 2021, 34: 6468–6479
9 Wang Y, Liu Z, Zuo Z, et al. Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control. IEEE Trans Veh Technol, 2019, 68: 8546–8556
10 Chen J, Zhang C, Luo J, et al. Driving maneuvers prediction based autonomous driving control by deep Monte Carlo tree search. IEEE Trans Veh Technol, 2020, 69: 7146–7158
11 Mo S, Pei X, Wu C. Safe reinforcement learning for autonomous vehicle using Monte Carlo tree search. IEEE Trans Intell Transp Syst, 2022, 23: 6766–6773
12 Li P, Pei X, Chen Z, et al. Human-like motion planning of autonomous vehicle based on probabilistic trajectory prediction. Appl Soft Comput, 2022, 118: 108499
13 Feng Y Y, Yan X L. Support vector machine based lane-changing behavior recognition and lateral trajectory prediction. Comput Intell Neurosci, 2022, 2022: 1–10
14 Xie G, Shangguan A Q, Fei R, et al. Motion trajectory prediction based on a CNN-LSTM sequential model. Sci China Inf Sci, 2020, 63: 212207
15 Zheng X, Zhang D, Gao H, et al. A novel framework for road traffic risk assessment with HMM-based prediction model. Sensors, 2018, 18: 4313
16 Wang Y, Wang C, Zhao W, et al. Decision-making and planning method for autonomous vehicles based on motivation and risk assessment. IEEE Trans Veh Technol, 2021, 70: 107–120
17 Jiang Y, Zhu B, Yang S, et al. Vehicle trajectory prediction considering driver uncertainty and vehicle dynamics based on dynamic Bayesian network. IEEE Trans Syst Man Cybern Syst, 2022, 53: 689–703
18 Gao H B, Su H, Cai Y F, et al. Trajectory prediction of cyclist based on dynamic Bayesian network and long short-term memory model at unsignalized intersections. Sci China Inf Sci, 2021, 64: 172207
19 Zuo Z, Wang X, Guo S, et al. Trajectory prediction network of autonomous vehicles with fusion of historical interactive features. IEEE Trans Intell Veh, 2023, 9: 2171–2183
20 Rudenko A, Palmieri L, Herman M, et al. Human motion trajectory prediction: a survey. Int J Robotics Res, 2020, 39: 895–935
21 Messaoud K, Yahiaoui I, Verroust-Blondet A, et al. Attention based vehicle trajectory prediction. IEEE Trans Intell Veh, 2020, 6: 175–185
22 Xing Y, Lv C, Mo X, et al. Toward safe and smart mobility: energy-aware deep learning for driving behavior analysis and prediction of connected vehicles. IEEE Trans Intell Transp Syst, 2021, 22: 4267–4280
23 Katariya V, Baharani M, Morris N, et al. Deeptrack: lightweight deep learning for vehicle trajectory prediction in highways. IEEE Trans Intell Transp Syst, 2022, 23: 18927–18936
24 Gao X, Si J, Huang H. Reinforcement learning control with knowledge shaping. IEEE Trans Neural Netw Learn Syst, 2023, 35: 3156–3167
25 Liu X, Wang Y, Zhou Z, et al. Trajectory prediction of preceding target vehicles based on lane crossing and final points generation model considering driving styles. IEEE Trans Veh Technol, 2021, 70: 8720–8730
26 Mo X Y, Xing Y, Lv C. Graph and recurrent neural network-based vehicle trajectory prediction for highway driving. In: Proceedings of the IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, 2021. 1934–1939
27 Zhou H, Ren D, Xia H, et al. AST-GNN: an attention-based spatio-temporal graph neural network for Interaction-aware pedestrian trajectory prediction. Neurocomputing, 2021, 445: 298–308
28 Lu Y, Wang W, Hu X, et al. Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks. IEEE Trans Intell Transp Syst, 2022, 24: 8452–8464
29 Xing Y, Lv C, Cao D. Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles. IEEE Trans Veh Technol, 2019, 69: 1341–1352
30 Zamboni S, Kefato Z T, Girdzijauskas S, et al. Pedestrian trajectory prediction with convolutional neural networks. Pattern Recogn, 2022, 121: 108252
31 Bogaerts T, Masegosa A D, Angarita-Zapata J S, et al. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transp Res Part C-Emerging Technol, 2020, 112: 62–77
32 Peng Z, Yan J, Yin H, et al. Efficient interaction-aware trajectory prediction model based on multi-head attention. Automot Innov, 2024, 7: 258–270
33 Lin L, Li W, Bi H, et al. Vehicle trajectory prediction using LSTMs with spatial-temporal attention mechanisms. IEEE Intell Transp Syst Mag, 2022, 14: 197–208
34 Greenacre M, Groenen P J F, Hastie T, et al. Principal component analysis. Nat Rev Methods Primers, 2022, 2: 100
35 Ikotun A M, Ezugwu A E, Abualigah L, et al. K-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data. Inf Sci, 2023, 622: 178–210
36 De Rango F, Tropea M, Serianni A, et al. Fuzzy inference system design for promoting an eco-friendly driving style in IoV domain. Vehicular Commun, 2022, 34: 100415
37 Shahapure K R, Nicholas C. Cluster quality analysis using silhouette score. In: Proceedings of the 7th International Conference on Data Science and Advanced Analytics (DSAA), 2020. 747–748
38 Ros F, Riad R, Guillaume S. PDBI: a partitioning Davies-Bouldin index for clustering evaluation. Neurocomputing, 2023, 528: 178–199
39 Thomson G P, Reid A. Diffraction of cathode rays by a thin film. Nature, 1927, 119: 890
40 Tang Y, Han K, Guo J Y, et al. An image patch is a wave: phase-aware vision MLP. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, 2022. 10935–10944
41 Coifman B, Li L. A critical evaluation of the next generation simulation (NGSIM) vehicle trajectory dataset. Transp Res Part B-Meth, 2017, 105: 362–377
42 Cheng J, Zhang X, Luo P, et al. An unsupervised approach for semantic place annotation of trajectories based on the prior probability. Inf Sci, 2022, 607: 1311–1327
43 Zheng Y, Hansen J H L. Lane-change detection from steering signal using spectral segmentation and learning-based classification. IEEE Trans Intell Veh, 2017, 2: 14–24
44 Fang H Z, Liu L, Xiao X F, et al. Vehicle trajectory prediction based on mixed teaching force long short-term memory. J Transp Syst Eng Inf Technol, 2023, 23: 80–87
45 Sheng Z, Xu Y, Xue S, et al. Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving. IEEE Trans Intell Transp Syst, 2022, 23: 17654–17665
46 Chen X, Zhang H, Zhao F, et al. Intention-aware vehicle trajectory prediction based on spatial-temporal dynamic attention network for Internet of vehicles. IEEE Trans Intell Transp Syst, 2022, 23: 19471–19483
47 Li C, Liu Z, Lin S, et al. Intention-convolution and hybrid-attention network for vehicle trajectory prediction. Expert Syst Appl, 2024, 236: 121412