Information Fusion 102 (2024) 102046

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

A multi-modal spatial–temporal model for accurate motion forecasting with visual fusion

Xiaoding Wang [a,b], Jianmin Liu [a,b], Hui Lin [a,b,*], Sahil Garg [c,*], Mubarak Alrashoud [d]

[a] College of Computer and Cyber Security, Fujian Normal University, No. 8 Xuefu South Road, Fuzhou, 350117, Fujian, China
[b] Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, No. 8 Xuefu South Road, Fuzhou, 350117, Fujian, China
[c] Electrical Engineering Department, École de technologie supérieure, Montreal, QC, H3C 1K3, Canada
[d] Department of Software Engineering (SWE), College of Computer and Information Sciences (CCIS), King Saud University, Riyadh, 11543, Saudi Arabia

ARTICLE INFO

Keywords:
Motion forecasting
Intelligent transportation
Spatial–temporal cross attention
Multi-source visual fusion

ABSTRACT

The multi-source visual information from ring cameras and stereo cameras provides a direct observation of the road, traffic conditions, and vehicle behavior. However, relying solely on visual information may not provide a complete environmental understanding. It is crucial for intelligent transportation systems to effectively utilize multi-source, multi-modal data to accurately predict the future motion trajectory of vehicles accurately. Therefore, this paper presents a new model for predicting multi-modal trajectories by integrating multi-source visual feature. A spatial–temporal cross attention fusion module is developed to capture the spatiotemporal interactions among vehicles, while leveraging the road's geographic structure to improve prediction accuracy. The experimental results on the realistic dataset Argoverse 2 demonstrate that, in comparison to other methods, ours improves the metrics of minADE (Minimum Average Displacement Error), minFDE (Minimum Final Displacement Error), and MR (Miss Rate) by 1.08%, 3.15%, and 2.14% , respectively, in unimodal prediction. In multimodal prediction, the improvements are 5.47%, 4.46%, and 6.50%. Our method effectively captures the temporal and spatial characteristics of vehicle movement trajectories, making it suitable for autonomous driving applications.

1. Introduction

With the popularization of artificial intelligence technology, fields such as autonomous driving, intelligent transportation, robotic control, internet of things (IoT) and intelligent logistics are thriving [1]. In intelligent transportation, there are several main tasks, including obstacle recognition [2], demand forecasting [3], traffic flow prediction [4,5], and trajectory prediction, etc. Accurately predicting the motion trajectory of moving objects is an important issue in order to ensure traffic safety and efficiency [6,7]. To achieve this goal, visual information is often fused with other sensor data, such as lidar, to provide additional environmental information. These data often have different modalities and can comprehensively describe the traffic situation more comprehensively. Various sensors are used to collect data in the environment, as shown in Fig. 1.

For example, a panoramic camera with multiple lenses, such as a ring camera, can capture a 360-degree field of view, providing a panoramic image that covers a wider range. There are many manufacturers whose lenses meet the requirements, such as Hikvision or Sony. A stereoscopic camera, such as Karmin3 and SceneScan Pro, composed of two lenses simulates the binocular vision of human eyes, enabling more accurate acquisition of three-dimensional structure and distance information. These multi-source visual information is often fused and understood together with point cloud information from LiDAR, such as VLP-32C, and time-series data from other sensors, in order to further represent rich traffic information. Multi-source visual fusion and understanding (MSVFU) aims to transform data sensed from the environment into raw models of perceptual content and then build a broader understanding of the world.

In fields like smart logistics, accurate trajectory prediction plays a crucial role in achieving objectives such as intelligent route planning and optimizing logistics and transportation operations. Similarly, in autonomous driving and robot control, predicting the trajectories of other vehicles, pedestrians, and obstacles by deep learning [8] is essential for ensuring safe and efficient navigation [9] and control decisions. Consequently, the development of trajectory prediction tasks holds substantial application value and societal importance. This paper focuses on trajectory prediction methods in autonomous driving. Due to

* Corresponding authors.
E-mail addresses: linhui@fjnu.edu.cn (H. Lin), sahil.garg@ieee.org (S. Garg).

https://doi.org/10.1016/j.inffus.2023.102046
Received 18 July 2023; Received in revised form 12 September 2023; Accepted 25 September 2023
Available online 6 October 2023

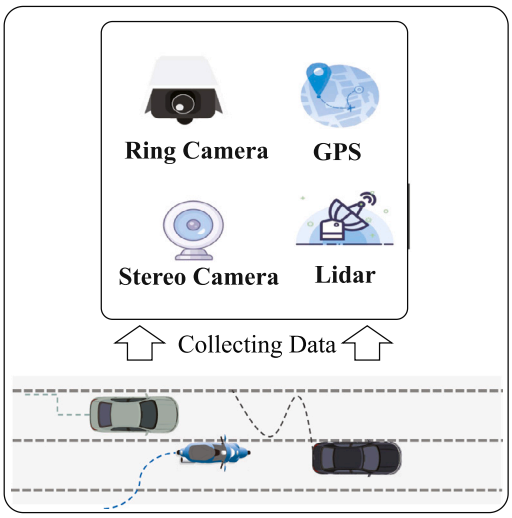


Fig. 1. Various sensors collect multi-source and multimodal data.

the high uncertainty of vehicle driving, it is common practice to predict multiple trajectories, and this paper is no exception.

In autonomous driving, the future trajectory of a vehicle is often influenced by three main factors: the road structure, the vehicle's historical movement trajectory, and the movement trajectories of other vehicles in its vicinity. The road structure sets limits on the vehicle's movement range, assuming it stays within the road boundaries. By analyzing the vehicle's historical movement trajectory, we can infer its driving intentions, such as going straight, turning, or changing lanes. Similarly, the movement trajectories of surrounding vehicles also impact the vehicle's trajectory, such as lane changes to avoid collisions or staying in place due to traffic congestion.

However, visual data is not always reliable. In scenarios such as nighttime, rainy weather, or heavy smog, the image quality captured by cameras can significantly deteriorate. Additionally, sudden changes in light intensity can temporarily blind the camera, such as when exiting a tunnel. These are currently unresolved issues in the field of vision. Furthermore, current trajectory prediction methods, such as [10–12], have certain limitations in modeling the spatial–temporal relationship and fail to fully exploit the potential spatial–temporal dependencies present in multimodal trajectory data. Moreover, another mothod [13] overlooks the constraints imposed by road geographical structures on vehicle trajectories, leading to subpar prediction performance. To tackle these challenges, this paper presents a novel multimodal spatial–temporal trajectory prediction model that incorporates multi-source visual fusion. The proposed method incorporates a cross-attention module to comprehensively model the spatial–temporal features and interaction relationships of vehicles' trajectories. Additionally, it integrates road geographical structures to enhance trajectory prediction accuracy. The key contributions of this paper are summarized as follows:

- A end-to-end trajectory prediction model has been proposed, which first embeds multi-source data and utilizes attention mechanism for fusion to explore latent trajectory features. Specifically, different embedding methods are applied to data from different sources, and deep fusion is achieved through cross-attention mechanism. By considering the positional relationship between vehicles at different time steps, a spatio-temporal graph is constructed to avoid the loss of temporal information and fully exploit the potential spatial dependencies in trajectory data.
- A Geographical Road Gating mechanism (GR Gating) is proposed, which utilizes a limited number of parameters to constrain the predicted trajectory coordinates within the road boundaries. This mechanism ensures that the predicted trajectory remains within the road, leading to more accurate predictions.

Table 1
Comparison of Trajectory Prediction Methods.

Method	HD-Map	Contrast
Kalman filter [14,15]	–	They are used for trajectory prediction of low-speed moving targets.
Markov chain model [16]	–	The prediction accuracy depends on the quality of the sample data. the computation of the state transition matrix is large and not suitable for real-time prediction.
RNN [17] LSTM [18–21]	–	The short-term task prediction accuracy has improved, but performs poorly in long-term prediction tasks.
Cart-pred [13]	–	It introduces crystal graph convolutional neural network to model the interaction between vehicle trajectories in mapless scenarios.
VectorNet [10]	✓	It integrates vectorized scene information and dynamic traffic participant information to achieve behavior prediction, and a simple prediction baseline is provided.
LANEGCN [22]	✓	LaneGCN considers the multi-order interaction of map nodes, and lane graph convolution is proposed.
LaPred [23] LaneRCNN [12]	✓	They aggregate the interaction information between vehicles based on road topology information.
TNT [11] DenseTNT [24]	✓	They predict through target-driven approach, which is inclined to predict driving intentions firstly.
mmTransform [25]	✓	It stacks multiple Transformers to model lanes, trajectories, and interaction relationships separately.
HOME [26] GoHome [27]	✓	They encode history and local context and decode to output heatmaps.
GANet [28]	✓	GANet proposes a goal area of interest operator to effectively extract semantic lane features in goal areas.
TPCN [29]	✓	TPCN converts trajectory position points and maps into point clouds for calculation.

- Extensive experiments were conducted on the publicly available Argoverse 2 motion forecasting dataset, demonstrating that the trajectory prediction method proposed in this paper has advanced performance.

The remainder of this paper is structured as follows. Section 2 provides an overview of the related works in the field. The system model and problem statement are presented in Section 3. The implementation details of the proposed model are elaborated upon in Section 4. The experimental methods and analysis are discussed in Section 5. Finally, Section 6 concludes this paper.

2. Related works

In recent years, there has been a surge of research and development efforts in the field of trajectory prediction, encompassing both academia and industry, which is shown in Table 1. These approaches can be broadly classified into two categories based on the data utilized: research methods based on historical trajectory data and research methods that incorporate high-precision maps.

Research methods based on historical trajectory data. They can be further divided into two main categories: those based on dynamic analysis models and mathematical statistics, and those based on deep learning.

In the realm of dynamic analysis models and mathematical statistics, trajectory prediction methods such as Kalman filter or Bayesian filter are commonly employed to predict trajectories, primarily for low-speed moving targets. For instance, Houenou et al. [14] utilize models that consider the instantaneous path of vehicles, road shape, and vehicle maneuverability to perform short-term trajectory prediction. Abbas MT et al. [15] employ multimodal Kalman filters to handle a wider range of possible trajectory scenarios. In order to accommodate the

requirements of trajectory prediction in nonlinear systems, Markov chain models have been introduced. However, the predictive accuracy of such models heavily relies on the quality of the sample data and necessitates high-quality data support. Additionally, the computation of state transition probability matrices is computationally intensive, rendering it unsuitable for real-time prediction.

Deep learning has significantly advanced trajectory prediction, particularly with the utilization of recurrent neural networks (RNNs), as demonstrated by Min et al. [17]. However, RNNs have shown limitations in long-term prediction tasks. To address this, Altché et al. [18] and others introduce the long short-term memory (LSTM) neural network, which has demonstrated success in predicting future longitudinal and lateral trajectories of vehicles on highways. Deo et al. [19] further improve the LSTM model by incorporating convolutional social pooling layers to more robustly capture the mutual dependencies of vehicle motion. Furthermore, Pecher et al. [16] improve the accuracy of trajectory prediction by increasing the complexity of both Markov models and neural network models. LSTM is widely used in trajectory prediction within deep learning techniques due to its strong performance in handling sequence data. However, it still has limitations in capturing spatial interactions between vehicles and incorporating semantic data. To address this, Mercat et al. [20] propose a prediction model that combines the LSTM encoding-decoding structure with a multi-head attention mechanism. Notably, they intentionally excluded the use of map information to explore how to predict the movement trajectories of other vehicles in the absence of high-precision maps. Another approach, CRAT-Pred [30], draws inspiration from materials science and employs crystal graph convolutional neural networks to model the interaction relationships between vehicle trajectories in scenarios where maps are not available.

Research methods for fusion of HD Maps. Research methods for integrating high definition maps have gained prominence in recent years, thanks to the ongoing advancements in high definition map technology. Advanced methods for integrating high definition maps often involve merging map information into a graph structure to enable a unified modeling approach. One such method is VectorNet [10], which treats road segments and vehicles as equal nodes and constructs a global interaction graph for trajectory prediction. LaneGCN [22], on the other hand, considers multi-level interactions among map nodes and propagates the vehicle trajectories features over the traffic network, incorporating attention mechanisms for data fusion between vehicle nodes and lane graphs to capture complex topological structure features. Other methods, such as LaPred [23] and LaneRCNN [12], aggregate interaction information among vehicles based on road topology, but this approach may result in reduced interpretability of vehicle interaction relationship modeling. In addition, the mmTransformer [25] stacks three Transformer modules to model the map, historical trajectories, and interaction information, respectively. And it proposes a region-based training strategy to emphasize the local environment of vehicles. TNT [11] introduces a trajectory prediction framework based on end points, sampling anchor points evenly along road centerlines, and predicting an offset as an end point. The trajectory is then completed by filling points according to the end point, followed by scoring and filtering. Densetnt [24] avoids the heuristic selection of end points by employing an anchor-free method. GANet [28] enhances future interaction information by designing a small-scale information fusion near the end point. HOME [26] encodes historical and local context, decodes output heatmaps using metric metrics, and employs a fully connected layer for trajectory sampling, ultimately decoding the final position of the trajectory. These mentioned methods primarily focus on trajectory prediction by employing various decoding techniques. In contrast to these approaches, our method places emphasis on the fusion of multi-source and multi-modal data.

Table 2
Explanation of Abbreviations.

Abbreviation	Explanation
ADE	Average Displacement Error
FDE	Final Displacement Error
HD Maps	high precision maps or high definition maps
LiDAR	laser imaging, detection, and ranging
MR	Miss Rate
MSVFU	Multi-source Visual Fusion and Understanding

3. Problem definition

In the field of autonomous driving, vehicle trajectory prediction involves predicting the future motion trajectory of a vehicle based on its historical movement track and MSVFU information of surrounding environment. In this task, the driving route of an autonomous vehicle is influenced by factors such as the road layout and the presence of other moving entities like pedestrians, bicycles, and other autonomous vehicles. The inputs for trajectory prediction typically include the vehicle's historical track data and a high definition map that provides semantic information.

The historical track, denoted as $J_{his} = J_T, J_{T-1}, \ldots, J_1$, represents the sequence of the past T time steps of the autonomous vehicle's historical tracks. Each J_t ($J_t = \{P, H, V, ts\}$) consists of four-tuples that describe the position and state of the autonomous vehicle. In particular, $P = \{x, y\}$ represents the current location coordinates of the vehicle. It is important to note that, in our experiments, we utilize relative position displacement with respect to the prediction point, rather than absolute coordinates. We take the current position coordinates as the origin, and calculate the displacement between the positions at other moments and the current moment. Then two displacement distances x and y were normalized separately. H represents the current direction of the vehicle, indicating its heading or orientation. $V = (V_x, V_y)$ represents the instantaneous velocity of the vehicle along the X and Y directions. The ts represents the timestamp of the current moment, providing temporal information about the data.

The semantic information of the high definition map refers to the map representation of the current scene surrounding the vehicle. It includes details such as the position coordinates of the road's center line, identification of intersections, lane directions, and the presence of traffic lights. This semantic information is denoted as M. Therefore, the task of trajectory prediction can be formulated as learning a mapping function f that takes into account the historical track information of the target vehicle and other moving agents in the scenario. This function aims to predict a set of possible trajectories for the target vehicle within a specific time horizon of N time steps. These predicted trajectories consist of k sequences. The function f can be expressed as:

$$J^k_{pred} = f(J_{his}, M) = \{J^k_{t+1}, \ldots, J^k_{t+N}\} \tag{1}$$

where J_{his} represents the historical track information, M represents the high definition map semantic information, k denotes the number of predicted multimodal trajectories, pre indicates the predicted time step. The goal of the model is to obtain more accurate trajectory sequences, that is, to minimize the difference between the predicted trajectory J^k_{pred} and the ground truth.

An explanation of some of the abbreviations that appear in the article is shown in the following Table 2. And the definitions of the symbols used are explained as Table 3.

4. Proposed model

How to achieve MSVFU in a large amount of data, and fully explore the spatiotemporal characteristics of vehicle trajectories, is a key consideration factor for traffic trajectory prediction models. This article presents a trajectory prediction model that incorporates a spatial–temporal cross attention mechanism. The model comprises three main

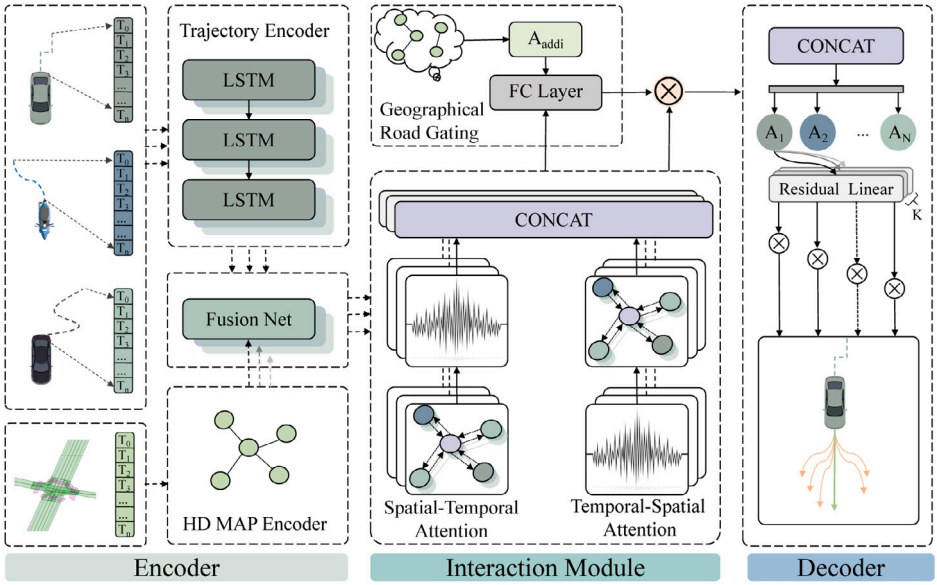


Fig. 2. The diagram of proposal framework. (It includes three parts: encoder, interaction module, and decoder. (a) Encoder. It encodes the tracks and maps separately and integrates the features through feature fusion. (b) Interaction module. It extracts the spatial–temporal interaction behavior between vehicles' tracks using cross-attention and outputs through a gating mechanism. (c) Decoder. It uses a linear residual layer to generate multi-modal tracks.).

Table 3
Explanation of Symbols.

Symbol	Explanation
J_{his}	Historical trajectories of the vehicle
P	Position of the vehicle, which is consists of location coordinates (x, y)
V	Velocity of the vehicle
H	Heading of the vehicle
ts	Timestamp of moment
M	Semantic information of the map
N	Time steps to prediction
k	Number of predicted multimodal trajectories
σ	Non-linear activation function
$\Delta_{i,j}$	The euclidean distance between node i and j
$\otimes$	Matrix multiplication

components: the input encoder module, the spatial–temporal interaction module, and the trajectory decoder module.

Encoder encodes the tracks and maps separately and integrates the features through feature fusion. The tracks and maps are processed individually to capture their respective information. The features extracted from both sources are then fused together to create a comprehensive representation. Interaction module focuses on extracting the spatial–temporal interaction behavior between vehicles' tracks. It accomplishes this by employing a spatial–temporal cross attention mechanism, which enables the model to attend to relevant information from different tracks. The outputs of the interaction module are further refined through a gating mechanism. The decoder utilizes a linear residual layer to generate multi-modal tracks. It takes the refined outputs from the interaction module and predicts multiple possible trajectories for the vehicles. The linear residual layer helps in capturing the residual information and refining the predictions. The schematic diagram of the model's structure is depicted in Fig. 2. Next, we will provide a comprehensive explanation of the functioning of each module.

4.1. Encoder

The function of the encoder is to separately encode and embed the historical trajectory sequence of the vehicle and the high-precision map information. By separately encoding and embedding the historical trajectory sequence and the high-precision map information, the encoder module enables the model to capture the distinctive characteristics of each data source. This facilitates the subsequent stages of the trajectory prediction model to effectively utilize the encoded information for generating accurate and context-aware predictions.

Trajectory Encoder. To embed the trajectory sequence features of vehicles, a stacked LSTM network layer is utilized, which could be represented by Eq. (2). Each historical trajectory of an agent, denoted as $J_{his} = \{J_T, J_{T-1}, \ldots, J_1\}$, is composed of quadruples $J_t = \{P, H, V, ts\}$ representing position coordinates, vehicle heading, instantaneous velocity, and timestamp at different time slices. The network parameters for embedding all trajectories are shared, resulting in a feature vector H_t^n for each agent node.

$$H_t^n = LSTM(J_{his}^n), \tag{2}$$

HD Map Encoder. The high definition map encompasses a wealth of semantic information, including lane positions, lane adjacency relationships, and intersection identification. To capture the adjacency relationships between lanes, four directed graphs are constructed denoted as $G(V, suc, pre, left, right)$, where V represents the node of center point of the lane. These graphs represent the lane node topology structure and are characterized by four types of edge connections: predecessor (pre), successor (suc), left neighbor (left), and right neighbor (right).

The LaneGCN (lane graph convolution) method was employed to aggregate node information. Given the uncertainty and low-order nature of the left and right neighbors, the convolution operation is performed only once. To account for the road's continuity in preceding and succeeding nodes, dilated convolution is introduced to expand the receptive field in the lane direction. The convolution operation is repeated k times to effectively capture long-range dependencies along the lane direction. The LaneGCN formula can be expressed as follows:

$$Y = XW_0 + \sum_{i \in left,right} A_i X W_i + \sum_{k=1}^{K} (A_{pre}^k X W_{pre,k} + A_{suc}^k X W_{suc,k}), \tag{3}$$

Where X represents node features; A_i represents the adjacency matrix for type i relationships; A_{pre}^k represents the kth power of the matrix A_{pre}; W is a trainable parameter; K represents the order of dilation convolution. Please note that inflation only occurs in predecessors and successors.

Fusion Net. Taking inspiration from the LaneGCN [22], a more general attention mechanism was employed for feature fusion. This

feature fusion aimed to propagate vehicle features through interactions with lanes, including vehicle-lane, lane-lane, lane-vehicle and vehicle-vehicle interactions. The ultimate goal was to achieve comprehensive feature fusion among vehicles. LaneGCN is employed specifically for lane-lane feature fusion. On the other hand, attention fusion is utilized for the fusion of vehicle-lane, lane-vehicle, and vehicle-vehicle features.

In the interaction between vehicles and roads, considering that our trajectory prediction task is in seconds, we introduce the concept of interaction range, as vehicles do not affect the traffic environment beyond 1 kilometer within a few seconds. The interaction ranges for vehicle-lane, lane-vehicle, and vehicle-vehicle are set to 10 m, 10 m, and 100 m, respectively, indicating that we will only consider the vehicle nodes and lane nodes within the interaction range. At each moment, the vehicle nodes and the surrounding lane nodes form an interaction graph. The formula (4) indicates that the features of a node at the next time step need to aggregate its own current information, the information from neighboring lane nodes, and the positional differences in their interaction information. For example, for vehicle node i, we can aggregate the features from its context lane node j, according to the following Eq. (4).

$$y_i = x_i W_0 + \sum_j \sigma(concat(x_i, \Delta_{i,j}, x_j)W_1)W_2, \tag{4}$$

where x_i is the feature of the ith node. W_0, W_1, W_2 is the trainable parameter. Specifically, W_0 represents how much of the original features are retained. The concat operation causes a change in dimension, and W_1 parameter restores the feature dimension, and W_2 represents the attention for each interaction node. $\Delta_{i,j}$ is the euclidean distance between node i and node j. Only nodes with $\Delta \leq threshold$ are considered, and the thresholds for vehicle-lane, lane-vehicle and vehicle-vehicle are set to 10, 10, and 100 meters respectively. And σ is the operation of normalization and non-linear activation function.

During the propagation process, both lane information and vehicle information were fully integrated and transmitted. This integration allowed for the effective combination of features from both sources, enabling a more holistic understanding of the interactions between vehicles and lanes.

4.2. Interaction module

4.2.1. Spatial attention module

In trajectory prediction tasks, it is essential to take into account the diverse interaction relationships between vehicles and others. To effectively capture and extract spatial dependencies among neighboring vehicles, spatial attention mechanisms are employed. The spatial attention mechanism calculates attention weights between different vehicles. These weights are then used to adjust the importance of each vehicle's information, better reflecting the relationships between them. By incorporating this mechanism, the model can more accurately predict the motion trajectories of each vehicle, resulting in improved overall prediction performance.

The interaction relationship between vehicles in the environment can be effectively represented by constructing a spatiotemporal graph, denoted as $G(V, E, A)$ and shown in Fig. 3. V comprises various entities such as vehicles and pedestrians, with each intelligent agent represented as a node. E indicates whether there exists an interaction between different intelligent agents. The interaction is determined based on the Euclidean distance between the agents' coordinate positions. If the distance is below a certain threshold, it signifies an interaction between the two vehicles or intelligent agents.

Spatial attention mechanism performs solely on interacting vehicles, rather than considering all vehicles. This approach reduces the number of parameters and enhances computational efficiency. Each subgraph assumes an interaction between all vehicles within it, resulting in the construction of a bidirectional fully connected graph. As time progresses, the positions of intelligent agents change, leading to variations in their interaction relationships. This process generates a sequence of spatial graphs that exhibit potential temporal relationships.

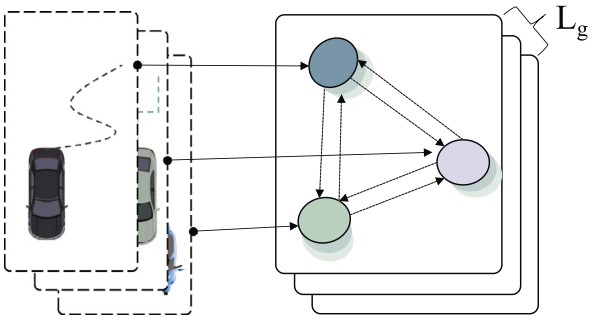


Fig. 3. Spatial interaction graph.

The spatial attention module employs graph convolutional network to effectively model the spatial relationships between vehicles. The computation method of graph convolutional network can be described as follows:

$$H^{(1+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(1)}W^{(1)}) \tag{5}$$

where $\tilde{A} = A + I_N$, A, I_N are the adjacency and identity matrix. $\tilde{D}$ is the degree matrix of $\tilde{A}$. W is a parameter used for training. The spatial attention module stacks multiple layers of GCN to explore multi-level interactions between vehicles. L_g denotes the layer of GCN and we set the parameter to 2. Non-linear activation function ReLU is used to connect each layer of GCN.

4.2.2. Temporal attention module

In the context of trajectory prediction for autonomous vehicles, existing methods primarily focus on considering the spatial structure of the lane and the spatial relationships between vehicles. However, they often lack a comprehensive modeling of temporal sequence features. When the speeds of vehicles are similar, the relative spatial positions tend to change less, leading to approximately or even identical graph structures. However, different time periods in trajectories may exhibit distinct features. For instance, traffic flow and driving speed can significantly differ during peak and off-peak periods, resulting in distinct characteristics observed in trajectories.

To alleviate this issue, we proposes the incorporation of a temporal attention module, aiming to enhance the contextual information and facilitate a comprehensive understanding of trajectory behavior, ultimately leading to improved prediction accuracy. By integrating temporal attention, the model can dynamically focus on the trajectory information from different time periods and make predictions based on their temporal significance. The vehicle features obtained from the upper layer are passed through a multi-head attention layer for dynamic encoding. This approach enhances the effectiveness and accuracy of the model while mitigating the risk of gradient vanishing or exploding.

The temporal attention mechanism, as shown in Fig. 4, takes as input a sequence of trajectories $x = (x_1, \ldots, x_n)$. The temporal attention mechanism calculates the correlation between each moment x_i and any other moment, and utilizes the temporal relevance to perform feature fusion. Finally, it outputs the attention-based result sequence $z = (z_1, \ldots, z_n)$. Each z_i can be calculated using the following Eq. (6):

$$z_i = \sum_{j=1}^{n} \alpha_{i,j}(x_j W^V), \tag{6}$$

where $\alpha_{i,j}$ denotes the attention coefficient between different moments, its calculation process can be described as Eq. (7):

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{n}\exp(e_{i,k})}, \tag{7}$$

and $e_{i,j}$ can be obtained by Eq. (8).

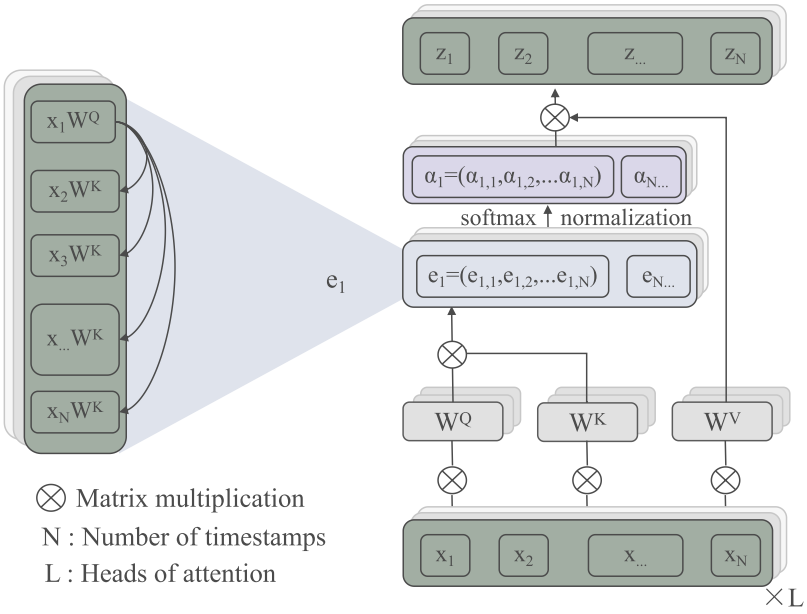


Fig. 4. Temporal attention module.

$$e_{i,j} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_k}}, \tag{8}$$

where W^Q, W^K, W^V are the three sets of trainable parameters.

At this moment, the output sequence z of the attention module takes into account the features from other arbitrary time steps and performs attention aggregation, enabling the capture of more comprehensive temporal features.

According to the model architecture diagram, it incorporates spatial–temporal cross attention to obtain two sets of spatiotemporal feature sequences, which $F_{st} = F_t(F_s(X))$ and $F_{ts} = F_s(F_t(X))$ are respectively represented as spatial-first and temporal-first features. The spatiotemporal interaction module takes these features as input and produces the fusion feature as the final output.

4.2.3. Geographical road gating

Designing gating parameters: By introducing the concept of macroscopic connectivity, it becomes possible to derive the adjacency matrix, denoted as A, for the road network.

$$A_{i,j} = \begin{cases} 1, & \text{if lane i, j exists interaction,} \\ 0, & \text{else.} \end{cases} \tag{9}$$

The element $A_{i,j}$ in the adjacency matrix indicates a connection between lane i and lane j. This connection is not limited to a specific relationship such as successor, predecessor, left, or right lane.

To quantitatively measure the strength of the association between two nodes, the parameter matrix W_{topy} is introduced, where $W \in \mathbb{R}^{N*N}$. Here, N represents the number of lanes in the road network. Each element $W_{i,j}$ in the matrix represents the possibility or possibility of transitioning from lane i to lane j. We expect

$$W_{i,j} = \begin{cases} \geq 0, & \text{if } A_{i,j} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Then, we employed a Rectified Linear Unit (ReLU) activation function and utilized the lane connection relationship matrix A as a mask. This means that during the computation, only the elements in the feature matrix that correspond to the connected lanes in the adjacency matrix A are considered. This helps to restrict the model's attention to the relevant lanes and improves the efficiency of information propagation.

$$W = ReLU(W_0) \cdot A \tag{11}$$

where W_0 is the parameter matrix of the gating mechanism, and symbol $\cdot$ refers to element-wise multiplication.

Design of gating function: The gating vector $g = G(Wy')$, where G is a mapping function and $\hat{y}$ is the result of spatiotemporal attention module. Since W directly reflects the strength of association between two lanes, where larger values indicate stronger associations. In order to ensure that the output of g is in the range [0,1], we define the G function as a sigmoid function.

$$g = G(x) = \frac{1}{1 + e^{-x}} \tag{12}$$

Therefore, by using g to correct the result of spatiotemporal attention module, we obtain y, the output result of the interaction module.

$$y = \hat{y} \cdot g \tag{13}$$

where $\hat{y}$ is the result of spatiotemporal attention module, and symbol $\cdot$ refers to element-wise multiplication. The gating mechanism is used to constrain trajectories in order to limit the variation trend of the trajectory.

4.3. Decoder

In real-world scenarios, vehicles exhibit diverse driving intentions. For instance, at an intersection, vehicles have the option to go straight, turn left, turn right, or make a U-turn. These choices lead to distinct trajectories and multiple potential destinations. Moreover, vehicles traveling towards the same destination may take different paths and change lanes at various positions, resulting in different sequences of trajectories. To address the challenge of multimodal trajectory sequence prediction, the spatiotemporal features of each vehicle are simultaneously fed into multiple residual linear layers. These layers generate multiple prediction results, allowing for the representation of different possible outcomes.

By employing Eq. (14), the encoded spatiotemporal features of trajectories are decoded to generate multimodal predicted trajectory sequences. The network architecture comprises multiple linear layers,

normalization layers, and non-linear activation layers. Each residual connection connects the input and the intermediate output of the model.

$$Output = Residual\,Decoder(y_{Fusion}) \tag{14}$$

Indeed, the residual structure in neural networks facilitates the learning of nonlinear transformations across multiple layers. This structure helps to alleviate the issues of gradient vanishing and exploding, enhances the expressive power of neural networks.

4.4. Loss function

The proposed end-to-end model in this paper utilizes the Smooth-L1 loss function, which is a regression loss function. In comparison to commonly used loss functions like Mean Absolute Error (MAE) and Mean Squared Error (MSE), the Smooth-L1 loss function offers stronger robustness and computational efficiency for regression models. It also provides better handling of outliers in the dataset. The expression for the Smooth-L1 loss function is as follows:

$$Smooth_{L1}(x) = \begin{cases} \frac{1}{2}x^2, & \text{if } |x| < 1, \\ |x| - \frac{1}{2}, & \text{otherwise.} \end{cases} \tag{15}$$

where x refers to the Euclidean distance error ($\|\widetilde{p} - p^*\|$) between the predicted value $\widetilde{p}$ and the true value p^* at a specific moment.

Therefore, the loss function of the entire model is defined as follows:

$$\text{loss} = \frac{1}{T}\sum_{t=1}^{T} \text{Smooth}_{L1}\left(\widetilde{p} - p^*\right) \tag{16}$$

In Eq. (16), T represents the each moment of each position in the trajectory sequence.

5. Experimental results and analysis

5.1. Dataset

Argoverse 1 motion forecasting [31] is a large-scale motion forecasting dataset with high-definition maps and sensor data,which consists of 324,557 real-world driving scenarios collected in Pittsburgh and Miami. Each scenario is 5 s long, for training and validation, while each test scenario presents only 2 s to the model, and another 3 s are withheld for the leaderboard evaluation. Each scenario contains the 2D, birds-eye-view centroid of each tracked object sampled at 10 Hz.

Argoverse 2 motion forecasting [32] consists of 250,000 scenes, with 224,896 scenes used for training and validation. Each scene is sampled at a frequency of 10 Hz, with a time length of 11 s (110 frames), and includes 2D positions, bird's-eye view center points, and orientations of tracked objects. During the experiment, we utilized the historical trajectories from the preceding 5 s to train the model. Subsequently, we employed this trained model to predict the future trajectories for the subsequent 6 s.

Compared to the Argoverse 1 motion forecasting dataset, the scenes in Argoverse 2 are approximately twice as long and more diverse. The dataset includes some complex social interaction behaviors, such as buses crossing multi-lane intersections, vehicles yielding to pedestrians at crosswalks, and cyclists sharing crowded city streets. These behaviors highlight unusual situations in terms of kinematics and social interactions, especially the behaviors exhibited by actors related to vehicle decision-making processes.

5.2. Parameter settings and evaluation metrics

The experiment was conducted on a server running Ubuntu 20.04 LTS. The server was equipped with an NVIDIA GeForce RTX 3090 GPU and 96 GB of RAM. For training purposes, we utilized the PyTorch 2.0 deep learning framework. The model training employed a batch size of 4 and the neural network consisted of 64 hidden units. The default vehicle interaction range was set to 100 m. The training process consisted of 72 epochs.

The Argoverse 1 and 2 test datasets only contain historical data from the previous 2 s and 5 s, respectively. Therefore, to evaluate and obtain online test results, it is necessary to submit the model's predicted future trajectory results to the Eval AI platform (website1) and (website2). EvalAI is an open-source platform for large-scale evaluation and comparison of machine learning (ML) and artificial intelligence (AI) algorithms. It is used for experimental evaluation of various conferences, including CVPR and ICCV.

This article evaluates the experimental results fundamental basic prediction parameters and evaluation metrics, focusing on two prediction results: unimodal prediction (K=1) and multi-modal prediction (K=6). During the prediction process, the data from the preceding 5 s is utilized to forecast the subsequent 6 s of motion trajectory. The standard evaluation metrics encompass the minimum average displacement error (minADEk), minimum final displacement error (minFDEk), and miss rate (MRk). The specific details of each metric are as follows:

(1) Average Displacement Error (ADE)
The minimum average displacement error (minADEk) metric represents the average Euclidean distance between the predicted value and the ground truth at each point during the prediction phase. It quantifies the average accuracy of the predicted trajectory within the prediction time T. The formula for ADE is as follows:

$$ADE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(\overline{y_i} - y_i\right)^2 + \left(\overline{x_i} - x_i\right)^2} \tag{17}$$

In the equation, $(\overline{x_i}, \overline{y_i})$ represents the position coordinates of the predicted trajectory, (x_i, y_i) represents the position coordinates of the actual trajectory, and n represents the number of samples within the predicted time period, which is 60 in this paper. When predicting multimodal trajectories, multiple trajectories correspond to multiple average displacement errors. This paper adopts the minimum average displacement error as the evaluation criterion.

(2) Final Displacement Error (FDE)
The final displacement error (minFDEk) metric represents the Euclidean distance between the predicted trajectory value and the true value at the last moment of the prediction phase. It quantifies the accuracy of the predicted trajectory at the final time step of the prediction period.

$$FDE = \frac{1}{m}\sum_{k=1}^{m}\sqrt{\left(\overline{y_n^k} - y_n^k\right)^2 + \left(\overline{x_n^k} - x_n^k\right)^2} \tag{18}$$

Among them, m represents the total number of trajectories, and this article takes either 1 or 6; k represents the kth trajectory; $(\overline{x_n^k}, \overline{y_n^k})$ represents the endpoint coordinate of the predicted trajectory; (x_n^k, y_n^k) represents the endpoint coordinate of the actual trajectory.

(3) Miss Rate (MR)
The miss rate (MRk) metric represents the proportion of failed predictions when the distance between the predicted trajectory and all true trajectories exceeds a certain threshold. In this study, the threshold size is set at 2.0 m. It measures the percentage of predictions that do not meet the specified accuracy criteria.

(4) Brier Minimum Final Displacement Error (brier-minFDE)
Brier-minFDE is similar to minFDE. It adds a penalty term, $(1.0 - p)^2$, to the L2 distance error towards the endpoints, which p corresponds to the predicted probabilities of the best predicted trajectory.

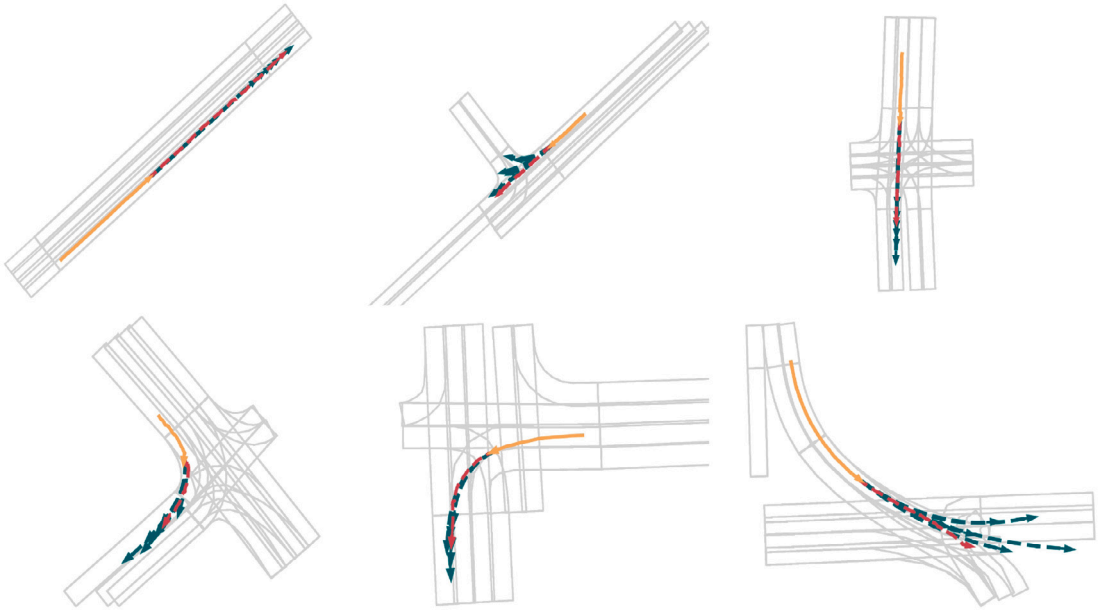


Fig. 5. Visualization of prediction results on Argoverse 1.

Table 4
Results on the Argoverse 1 Test Set.

Method	minADE (k = 1)	minFDE (k = 1)	MR (k = 1)	minADE (k = 6)	minFDE (k = 6)	MR (k = 6)	brier-minFDE (k = 6)
LaneRCNN [12]	**1.685**	3.692	0.569	0.904	1.453	0.123	2.147
TNT [11]	2.174	4.959	0.710	0.910	1.446	0.166	2.140
DenseTNT [24]	1.703	3.696	0.599	0.911	1.381	0.103	2.076
LaneGCN [22]	1.706	3.779	0.591	0.868	1.364	0.163	2.059
mmTransformer [25]	1.774	4.003	0.618	0.844	1.338	0.154	2.033
GOHOME [27]	1.689	3.647	0.572	0.943	1.450	0.105	**1.983**
HOME [26]	1.73	3.73	0.584	0.94	1.45	**0.102**	–
Ours	1.714	**3.6091**	**0.5574**	**0.8274**	**1.3297**	0.107	2.012

Table 5
Results on the Argoverse 2 Test Set.

Method	minADE (k = 1)	minFDE (k = 1)	MR (k = 1)	minADE (k = 6)	minFDE (k = 6)	MR (k = 6)	brier-minFDE (k = 6)
ConsVehocity [21]	7.7215	17.4607	0.8973	–	–	–	18.1552
LSTM-ED [21]	6.0257	16.8254	0.9389	1.6572	3.7231	0.5787	4.8773
CradPred [13]	4.9376	13.3025	0.9229	1.2088	2.8206	0.425	3.6843
furtherAI [33]	4.10	**10.26**	**0.90**	2.60	6.10	0.67	6.81
Narsis [33]	4.63	12.85	0.94	1.25	2.85	0.44	3.72
Ours	4.5798	12.4447	0.9199	**1.1816**	**2.7228**	**0.4114**	**3.582**

5.3. Experimental results

The comparison results will be listed as follows for analysis and evaluation. Table 4 shows the test results on Argoverse 1.1, while Table 5 shows the test results on Argoverse 2.

To assess the predictive performance of the model, we visually present the multimodal prediction results, as depicted in Figs. 5 and 6. In the figure, the orange rectangles represent the focused vehicles, while the light rectangles represent other vehicles. The burnt orange line represents the historical trajectory, the red line represents the ground truth, and the dark green line represents the multimodal predicted trajectory. This figure showcases real-life situations in different scenarios, such as traffic congestion, turning, lane changing, oncoming traffic, straight road driving, and intersection driving. It provides a visual representation of the model's ability to accurately predict and handle complex driving scenarios.

According to the experimental results, the trajectory prediction model proposed in this paper demonstrates accurate predictions of vehicle trajectories in various scenarios, such as straight driving, intersection turning, and traffic congestion. In these scenarios, at least one predicted trajectory completely overlaps with the real trajectory, indicating the model's capability to predict the uncertainty of vehicle driving intentions through spatiotemporal interactions between vehicles. This approach enables accurate trajectory prediction. By observing Fig. 6(a) traffic congestion and Fig. 6(d) straight driving at intersections, it is evident that the predicted trajectories have different lengths. On straight roads without traffic congestion, the vehicle tend to have higher speed, resulting in longer predicted trajectories. Conversely, on congested roads, the slower movement of vehicles leads to shorter predicted trajectories. This demonstrates that the model proposed in this paper can accurately predict the trajectory of a vehicle

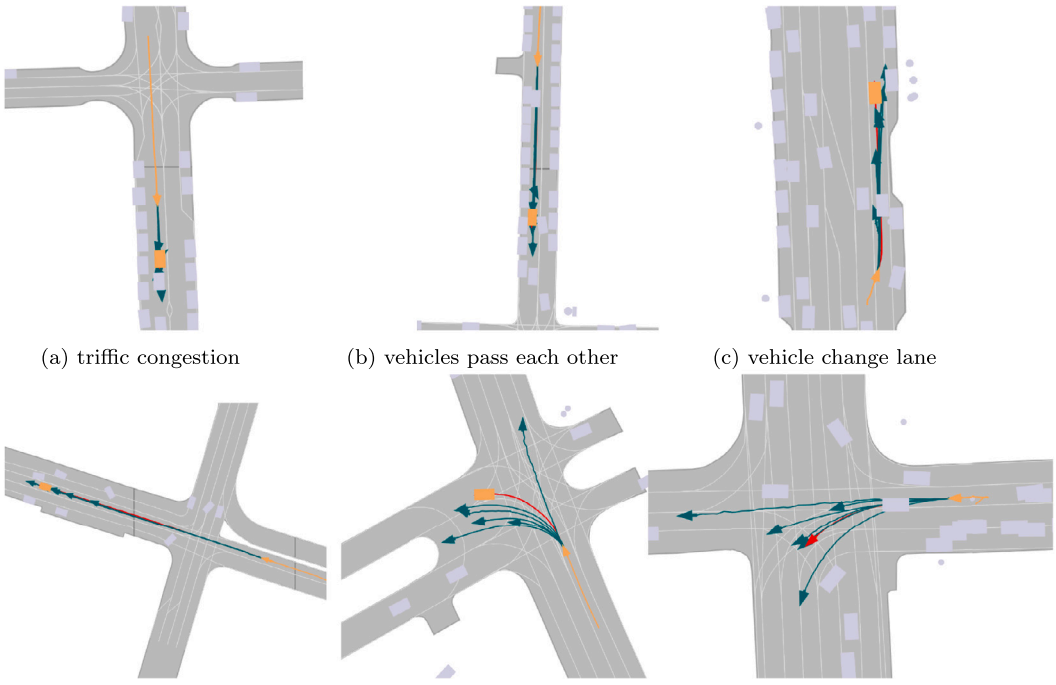


Fig. 6. Visualization of prediction results on Argoverse 2.((a) It displays the trajectory of vehicles during traffic congestion. (b) It shows the trajectory of vehicles during encounters. (c) It demonstrates the lane-changing behavior of vehicles. (d) Vehicles proceeding straight at intersections. (e) Vehicles making turns at intersections. (f) Motorcycles making turns at intersections.).

by predicting its driving speed. Additionally, by analyzing Fig. 6(c) and Fig. 6(e), it can be observed that the model can accurately predict future behaviors such as lane changes and turns. In Fig. 6(b), the orientation of the vehicle indicates that it has made the appropriate choice to avoid oncoming traffic, demonstrating that the model can effectively capture the interaction between vehicles and make correct decisions. Furthermore, upon examining the final Fig. 6(f), it is evident that the model is not only applicable to predicting vehicle trajectories but also to other traffic participants, such as motorcyclists or pedestrians.

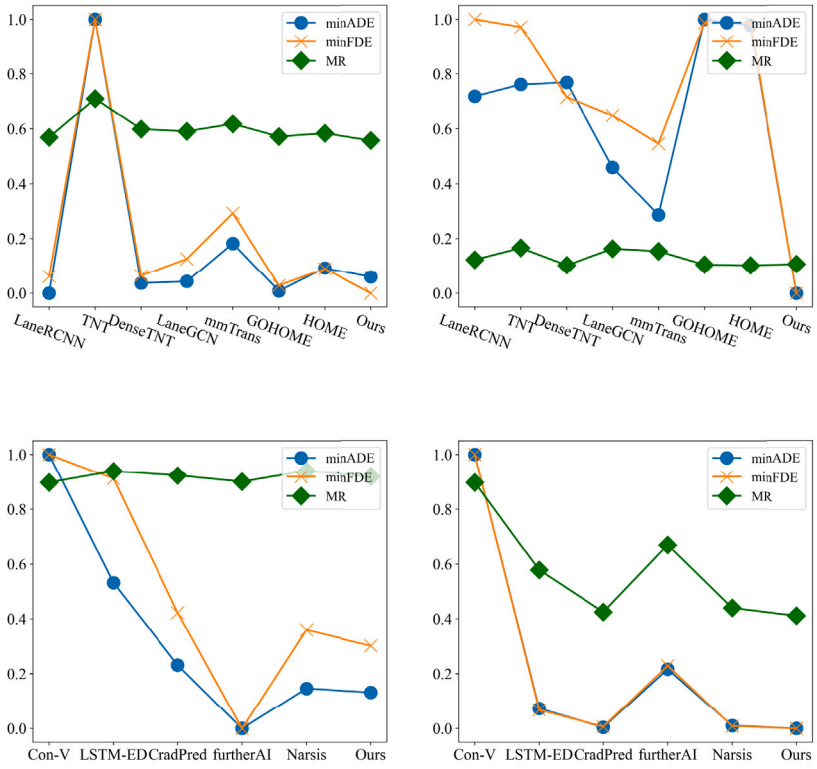
5.4. Experimental analysis

There are more methods implemented in the Argoverse 1 dataset as shown in Figs. 7(a) and 7(b). LaneGCN [22] proposes the LaneConv method to model the interaction between lanes and vehicles. Our method borrows the LaneConv operation and introduces spatio-temporal cross-attention mechanism and gate mechanism to further extract features and control outputs, thus achieving better performance. TNT [11] and DenseTNT [24] adopt target-driven decoding, where the selection of targets relies on heuristic knowledge, and the early trajectory feature modeling is not as sufficient as the LaneConv method. Thanks to the spatio-temporal modeling of the cross-attention mechanism, our method achieves better performance. Home [26] and GoHome [27] use heatmaps to predict target positions. Due to the lane-level heatmaps, the prediction results focus more on the range rather than the fine-grained position. LaneRCNN [12] uses global information when dealing with the interaction between lanes and vehicles, which introduces some noise, especially in multi-modal prediction, leading to performance degradation caused by accumulated errors. Similarly, the region training strategy of mmTransformer [25] does not consider the surrounding environment features during feature extraction modeling. To avoid this problem, we only consider the local environment within the influence range of the vehicles.

In order to further analyze the effectiveness of the method in this study, we compare the performance of various models under different indicators, as shown in Figs. 7(c) and 7(d). Based on the results, it can be observed that the method proposed in this paper performs better than other mainstream methods in trajectory prediction tasks. Compared with the constant velocity [21] baseline method, the minADE and minFDE indicators in the single-modal prediction of this paper's method have improved by 40.69% and 28.73% respectively. However, in single-modal prediction, the miss rate has increased by 2.52%, shown in subgraph 7(c). This may be because the neural network may get stuck in a local optimum during single-modal prediction. In contrast, in multi-modal prediction tasks, due to the simultaneous operation of multiple neural networks, it is not easy to get stuck in a local optimum, resulting in a significant decrease in the miss rate.

In comparison, prediction methods based on deep learning perform more ideally. Compared with the LSTM-ED method [21], the proposed method in this paper has improved the performance on various indicators in single-modal prediction tasks by 24.00%, 26.04%, and 2.02% respectively, and in multi-modal prediction tasks by 28.70%, 26.87%, and 28.91% respectively. The LSTM method only considers the temporal features of trajectories, while ignoring the spatial interaction effects between trajectories, thus resulting in lower prediction accuracy compared to the method proposed in this paper.

Compared to the CradPred [13], the proposed method has improved the performance on single-modal prediction by 7.25%, 6.45%, and 0.33% for different metrics, and on multi-modal prediction by 2.25%, 3.47%, and 3.20% for different metrics. Similar to the CradPred method, the method in this paper also considers the temporal sequence features among trajectories and introduces self-attention and graph networks to capture the spatial interaction relationships among trajectories, resulting in higher prediction accuracy compared to LSTM-based methods. However, this method further improves the prediction accuracy by incorporating map data and cross-attention to model spatial–temporal features, reducing the loss of different spatial–temporal domain features and enhancing the effectiveness of modeling long-term spatial–temporal features. The data for the Narsis and FurtherAI methods is sourced from the Eval AI leaderboard [33], but specific details are not provided. Therefore, no further analysis is conducted and it is only used for comparison with the performance of the model in this paper.



(c) Argoverse 2 (k=1) (d) Argoverse 2 (k=6)

Fig. 7. Improvement in experimental results.

Table 6
Ablation Experiment Results on Argoverse 1 Test set.

Method	minADE (k = 1)	minFDE (k = 1)	MR (k = 1)	minADE (k = 6)	minFDE (k = 6)	MR (k = 6)	b-minFDE (k = 6)
/o ST	1.8572	3.8655	0.5626	0.8327	1.3489	0.1232	2.1127
/o TS	1.8631	3.8729	0.5793	0.8384	1.3672	0.1198	2.1038
/o RG	1.9724	4.0172	0.5673	0.8431	1.4271	0.1211	2.1781
Ours	1.714	3.6091	0.5574	0.8274	1.3297	0.107	2.012

Table 7
Ablation Experiment Results on Argoverse 2 Test set.

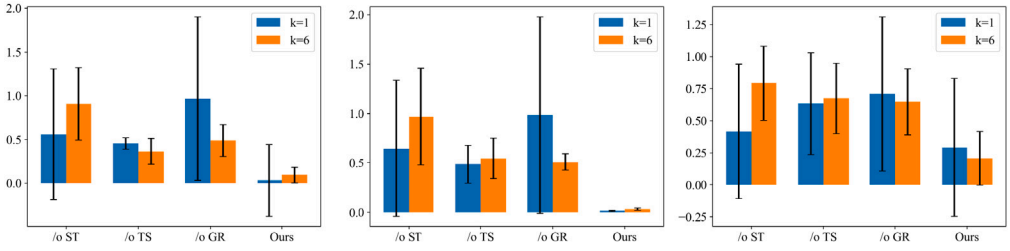
Method	minADE (k = 1)	minFDE (k = 1)	MR (k = 1)	minADE (k = 6)	minFDE (k = 6)	MR (k = 6)	b-minFDE (k = 6)
w/o ST	4.6912	12.7944	0.9216	1.2298	2.9944	0.4437	3.9690
w/o TS	4.7589	13.0096	0.9234	1.3535	3.2653	0.4516	4.4205
w/o GR	4.8943	13.2969	0.9387	1.2523	2.9128	0.4397	3.7813
Ours	4.5798	12.4447	0.9199	1.1816	2.7228	0.4114	3.582

5.5. Ablation experiment

To ensure the effectiveness of the proposed model, this paper conducted ablative experiments to analyze the model. These experiments removed the spatial–temporal (ST) attention branch, temporal–spatial (TS) attention branch, and the Geographical Road Gating (GR) module to evaluate the effectiveness of the spatiotemporal interaction module. The experiments were also conducted on the test set, and the Eval AI platform provided the results at Table 6 and Table 7:

Similarly, in order to further analyze the impact of each module of the model on its performance, we compared the ablation results on the two datasets, as shown in Fig. 8, which illustrates the mean and variance magnitudes on two datasets. It shows that after removing the spatial–temporal attention branch (ST) and the temporal–spatial attention branch (TS) respectively, the model's performance in both unimodal and multimodal prediction tasks is reduced to varying degrees. Specifically, after removing the temporal–spatial attention branch, the performance of the unimodal prediction task decreased by 2.43%, 2.81%, and 0.18% for each metric, and the performance of the multimodal task decreased by 4.08%, 9.98%, and 7.85% for each metric. Similarly, after removing the spatial–temporal attention branch, the performance of the unimodal prediction task decreased by 3.91%, 4.54%, and 0.38%, and the performance of the multimodal task decreased by 14.55%, 19.95%, and 9.77% for each metric. This indicates that these two branches play an important role in extracting spatiotemporal features for trajectory information, validating the effectiveness of the modules.

Whether it is the space–time attention branch or the time-space attention branch, both are exploring the changing trends of the trajectory's future position at a certain moment. However, different orders may result in the loss of different information, which has been verified



(a) minADE (b) minFDE (c) MR

Fig. 8. Improvement in ablation experimental results.

in the erasure experiment, manifested as varying degrees of performance decline. Therefore, integrating both orders can supplement some spatiotemporal information, reduce information loss, and thus improve the accuracy of prediction.

Furthermore, when the gating module (GR) is removed, the model's performance in unimodal prediction is reduced by 6.87%, 6.85%, and 2.04% for each metric, and the performance in multimodal prediction also decreases to varying degrees, by 5.98%, 6.98%, and 6.88% for each metric. The fused features can include fine-grained spatiotemporal characteristics, but vehicles travel on a macroscopic lane. Therefore, by utilizing the macroscopic road structure to constrain the output results, it is possible to avoid some unrealistic prediction scenarios, such as surmount or break the path.

Therefore, the experimental results fully demonstrate the indispensability of the sub-modules and the overall spatial–temporal interaction module in trajectory prediction tasks.

6. Conclusion

This paper proposes a new spatiotemporal trajectory prediction model that integrates and represents multimodal data from multiple sources of visual data and other sensors, thereby exploring the comprehensive value of the multimodal data. Employing the MSVFU technique enhances both the precision and dependability of trajectory prediction. The model is based on graph neural networks and spatial–temporal cross attention mechanisms, combined with high-precision map information, to extract and analyze the spatiotemporal features inherent in data that integrates visual information and semantic content. It utilizes a road gating mechanism to constrain the output results within the road network and finally predicts possible multimodal driving trajectories in the future using a residual network. Extensive testing was conducted on the publicly available real-world dataset Argoverse 2, and the results show that the proposed model achieves good performance. However, the current methods overly rely on high-definition maps, which are costly and time-consuming to collect and may not be available in some areas. Furthermore, target-driven methods primarily focus on predicting the vehicle's driving intention, which offers higher interpretability. Therefore, we will consider target-driven or map-free trajectory prediction methods.

CRediT authorship contribution statement

Xiaoding Wang: Conceptualization, Writing – review & editing, Supervision. **Jianmin Liu:** Methodology, Software, Formal analysis, Validation, Visualization, Writing – original draft. **Hui Lin:** Funding acquisition, Editing. **Sahil Garg:** Editing, Supervision, Project administration. **Mubarak Alrashoud:** Resources, Validation, Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Researchers Supporting Project Number (RSPD2023R800), King Saud University, Riyadh, Saudi Arabia.

References

[1] J. Mills, J. Hu, G. Min, Communication-efficient federated learning for wireless edge intelligence in IoT, IEEE Internet Things J. 7 (7) (2020) 5986–5994.
[2] K. Guo, Z. Wu, W. Wang, S. Ren, X. Zhou, T.R. Gadekallu, E. Luo, C. Liu, GRTR: Gradient rebalanced traffic sign recognition for autonomous vehicles, IEEE Trans. Autom. Sci. Eng. (2023) 1–13.
[3] S. Feng, J. Ke, H. Yang, J. Ye, A multi-task matrix factorized graph neural network for co-prediction of zone-based and OD-based ride-hailing demand, IEEE Trans. Intell. Transp. Syst. 23 (6) (2022) 5704–5716.
[4] J. Ou, J. Sun, Y. Zhu, H. Jin, Y. Liu, F. Zhang, J. Huang, X. Wang, STP-TrellisNets+: Spatial-temporal parallel TrellisNets for multi-step metro station passenger flow prediction, IEEE Trans. Knowl. Data Eng. 35 (7) (2023) 7526–7540.
[5] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, Y. Zheng, Spatio-Temporal meta learning for urban traffic prediction, IEEE Trans. Knowl. Data Eng. 34 (3) (2022) 1462–1476.
[6] L. Zhang, S. Li, A literature review on intelligent body trajectory prediction based on deep learning, Radio Eng. 53 (03) (2023) 644–656.
[7] Z. Chen, J. Hu, G. Min, A.Y. Zomaya, T. El-Ghazawi, Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning, IEEE Trans. Parallel Distrib. Syst. 31 (4) (2020) 923–934.
[8] J. Mills, J. Hu, G. Min, Multi-Task federated learning for personalised deep neural networks in edge computing, IEEE Trans. Parallel Distrib. Syst. 33 (3) (2022) 630–641.
[9] N. Kamath B, R. Fernandes, A.P. Rodrigues, M. Mahmud, P. Vijaya, T.R. Gadekallu, M.S. Kaiser, TAKEN: A traffic knowledge-based navigation system for connected and autonomous vehicles, Sensors 23 (2) (2023) 653.
[10] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, C. Schmid, Vectornet: Encoding hd maps and agent dynamics from vectorized representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11525–11533.
[11] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, et al., Tnt: Target-driven trajectory prediction, in: Conference on Robot Learning, PMLR, 2021, pp. 895–904.
[12] W. Zeng, M. Liang, R. Liao, R. Urtasun, Lanercnn: Distributed representations for graph-centric motion forecasting, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2021, pp. 532–539.
[13] J. Schmidt, J. Jordan, F. Gritschneder, K. Dietmayer, CRAT-pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention, in: 2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022, IEEE, 2022, pp. 7799–7805.
[14] A. Houenou, P. Bonnifait, V. Cherfaoui, W. Yao, Vehicle trajectory prediction based on motion model and maneuver recognition, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2013, pp. 4363–4369.
[15] M.T. Abbas, M.A. Jibran, M. Afaq, W.-C. Song, An adaptive approach to vehicle trajectory prediction using multimodel Kalman filter, Trans. Emerg. Telecommun. Technol. 31 (5) (2020) e3734.
[16] P. Pecher, M. Hunter, R. Fujimoto, Data-driven vehicle trajectory prediction, in: Proceedings of the 2016 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation, 2016, pp. 13–22.

[17] K. Min, D. Kim, J. Park, K. Huh, RNN-based path prediction of obstacle vehicles with deep ensemble, IEEE Trans. Veh. Technol. 68 (10) (2019) 10252–10256.
[18] F. Altché, A. de La Fortelle, An LSTM network for highway trajectory prediction, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2017, pp. 353–359.
[19] N. Deo, M.M. Trivedi, Convolutional social pooling for vehicle trajectory prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1468–1476.
[20] J. Mercat, T. Gilles, N. El Zoghby, G. Sandou, D. Beauvois, G.P. Gil, Multi-head attention for multi-modal joint vehicle motion forecasting, in: 2020 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2020, pp. 9638–9644.
[21] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, J. Hays, Argoverse: 3D tracking and forecasting with rich maps, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 8740–8749.
[22] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, R. Urtasun, Learning lane graph representations for motion forecasting, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, Springer, 2020, pp. 541–556.
[23] B. Kim, S. Park, S. Lee, E. Khoshimjonov, D. Kum, J. Kim, J. Kim, J. Choi, Lapred: Lane-aware prediction of multi-modal future trajectories of dynamic agents, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 14631–14640.
[24] J. Gu, C. Sun, H. Zhao, Densetnt: End-to-end trajectory prediction from dense goal sets, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15303–15312.
[25] Y. Liu, J. Zhang, L. Fang, Q. Jiang, B. Zhou, Multimodal motion prediction with stacked transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7577–7586.
[26] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, F. Moutarde, Home: Heatmap output for future motion estimation, in: 2021 IEEE International Intelligent Transportation Systems Conference, ITSC, IEEE, 2021, pp. 500–507.
[27] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, F. Moutarde, Gohome: Graph-oriented heatmap output for future motion estimation, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 9107–9114.
[28] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, W. Yang, Ganet: Goal area network for motion forecasting, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 1609–1615.
[29] M. Ye, T. Cao, Q. Chen, Tpcn: Temporal point cloud networks for motion forecasting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11318–11327.
[30] J. Schmidt, J. Jordan, F. Gritschneder, K. Dietmayer, Crat-pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 7799–7805.
[31] M.-F. Chang, J.W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, J. Hays, Argoverse: 3D tracking and forecasting with rich maps, in: Conference on Computer Vision and Pattern Recognition, CVPR, 2019.
[32] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J.K. Pontes, D. Ramanan, P. Carr, J. Hays, Argoverse 2: Next generation datasets for self-driving perception and forecasting, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, NeurIPS Datasets and Benchmarks 2021, 2021.
[33] ArgoAI-Argoverse, Argoverse 2: Motion forecasting competition. [Online], Available: https://eval.ai/web/challenges/challenge-page/1719/overview/.