Mech. Sci., 16, 87–97, 2025
https://doi.org/10.5194/ms-16-87-2025

Mechanical Sciences
Open Access

Interactive trajectory prediction for autonomous driving based on Transformer

Rui Xu[1], **Jun Li**[1], **Shiyi Zhang**[2], **Lei Li**[3], **Hulin Li**[4], **Guiying Ren**[3], **and Xinglong Tang**[3]

[1]School of Mechanotronics and Vehicle Engineering, Chongqing Jiaotong University,
Chongqing, 400074, China
[2]School of Shipping and Naval Architecture, Chongqing Jiaotong University, Chongqing, 400074, China
[3]Institute of Agricultural Machinery, Chongqing Academy of Agricultural Sciences, Chongqing, 400074, China
[4]School of Traffic and Transportation, Chongqing Jiaotong University, Chongqing, 400074, China

Correspondence: Jun Li (cqleejun@163.com)

Received: 4 May 2023 – Revised: 28 July 2024 – Accepted: 6 August 2024 – Published: 6 February 2025

Abstract. Trajectory planning has undergone remarkable strides in recent times, especially in the behavior prediction of traffic participants. Given that strong coupling conditions such as pedestrians, vehicles, and roads restrict the interactive behavior of autonomous vehicles and other traffic participants, it has become critical to design a trajectory prediction algorithm based on traffic scenarios for autonomous-driving technology. In this paper, we propose a novel trajectory prediction algorithm based on Transformer networks, a data-driven method that ingeniously harnesses dual-input channels. The rationale underlying this approach lies in its seamless fusion of scene context modeling and multi-modal prediction within a neural network architecture. At the heart of this innovative framework resides the multi-headed attention mechanism, ingeniously deployed in both the agent attention layer and the scene attention layer. This finessing not only captures the profound interdependence between agents and their surroundings but also imbues the algorithm with a better real-time predictive prowess, enhancing computational efficiency. Eventually, substantial experiments with the Argoverse dataset will demonstrate improved trajectory accuracy, with the minimum average displacement error (MADE) and minimum final displacement error (MFDE) being reduced by 12 % and 31 %, respectively.

1 Introduction

Intelligent transportation systems for traffic control and management are implemented to manage unexpected traffic situations while boosting road safety. Specifically, effective trajectory planning of autonomous vehicles requires the ability to forecast the potential behavior of traffic participants with high accuracy in order to make safe decisions. The question of how to improve the prediction accuracy of drivers' behaviors has spurred widespread research efforts. The activities of traffic agents are invariably multi-modal (Yao et al., 2021), predominantly affected by two kinds of factors: external environmental factors, such as road constraints and rules, and the driver's behavior intention, such as overtaking and changing lanes. It should be noted that the predicted trajectories exhibit strong nonlinearity over an extended period with regard to the uncertainty in individual behaviors (Min et al., 2020). Accurate long-term prediction, a promising strategy for autonomous vehicles, enables autonomous vehicles to promptly judge the target behavior, react to potential changes in the surrounding environment, and boost overall safety and comfort.

Model-based methods (Neto et al., 2018) can ensure short-term prediction accuracy. However, for long-term prediction of complex scenarios, if there is insufficient consideration of road scenarios or a lack of prior knowledge regarding driving, the accuracy of predictions will be significantly reduced, and the adaptability to different scenarios will also be affected. In long-term prediction, the behavior intention, and the dynamic behavior of the target body may change over time. At this point, the behavior of the target body is greatly influenced by its intention and surrounding environmental information (Jeong et al., 2020). The temporal correlation between current and previous time steps weakens in data, leading to an increased error rate in long-term prediction. This poses a major challenge in achieving accurate predictions.

A more accurate prediction made possible by an improved model serves as a significant indicator of system performance. In light of these challenges, we concentrate on the Transformer and VectorNet models (Gao et al., 2020), which is adept at capturing long-term interactions and scene context modeling. The improved prediction model is illustrated in Fig. 1. Drawing from map data and participant trajectories, it adeptly captures the interactions between the subject and the road, achieving high-precision prediction of the subject's future movement trajectory.

The proposed algorithm has been rigorously assessed utilizing the Argoverse dataset, which encapsulates an extensive array of intricate traffic scenarios, each representing real-world driving conditions. Experimental results demonstrate enhanced trajectory accuracy compared to existing models, with a 12 % reduction in the minimum average displacement error (MADE) and a 31 % reduction in the minimum final displacement error (MFDE).

The main contributions of the paper are as follows:

1. A novel neural network framework, equipped with scene context information and multi-modal prediction information, is introduced to grasp the temporal and spatial interactions among agents.

2. A VectorNet-based map simplification method is implemented by extracting the primary map structure and collecting fine-grained data to optimize prediction accuracy.

3. A multi-head attention mechanism that integrates vehicle and scene attention layers in parallel is developed to enhance computational efficiency and real-time prediction capabilities.

The rest of this paper is organized as follows: Sect. 2 discusses the current literature and outlines the approach adopted in the paper, and the structure of the proposed model is described in Sect. 3. Section 4 presents simulation results and a performance comparison of the model. The main conclusions and potential avenues for future work are summarized in Sect. 5.

2 Related work

In accordance with the expression and architecture of the existing work, this section presents the latest progress in the modeling of scene contexts and multi-modal predictions, which constitute the key issues in the prediction tasks and currently active areas of research.

2.1 Context representation

The rasterized map depicts the agent scenario and the road geometry environment in a most universal and straightforward manner. CoverNet (Phan-Minh et al., 2020) and RasterNet (Djuric et al., 2020) simplify the prediction tasks by rendering different traffic entities and rasterizing high-definition maps and surroundings into a bird's-eye-view image. However, rasterized images tend to be overly complex representations of the environment. Graph neural networks and graph convolutional networks (Cui et al., 2020; Wu et al., 2020) are becoming increasingly essential in processing unstructured data as a means to circumvent this onerous process. VectorNet (Gao et al., 2020) vectorizes map information and participant trajectories to construct polyline graphs fed into graph neural networks for vehicle trajectory prediction, thereby reducing data loss. In the LaneGCN (Li et al., 2020), the complex topology and long-range correlation of the lane map are captured through multiple adjacency matrices and multi-scale expansion convolution on the raw map data. Both of these demonstrate strong capabilities in extracting spatially local information.

2.2 Trajectory prediction

Simple physical models provide a valuable means to correlate some control inputs and external conditions with the evolution of the vehicle state (Wang et al., 2019; Xie et al., 2017) for short-term prediction windows of 1 s or less. Behavior models based on Kalman filtering (Keller and Gavrila, 2013), hidden Markov models (Neto et al., 2018; Ye et al., 2015), and their variants (Hubmann et al., 2018) have been employed for intensive investigation to gain a deeper understanding of vehicle motion and behavior.

However, road participants exhibit diverse dynamic behaviors, which can be associated with either aggressive or conservative driving styles. Consequently, the neural network models based on data have gained wider acceptance in order to enhance the accuracy of vehicle movement analysis and to identify road user behaviors more effectively. In contrast, the deep neural network model, based on big data and used to learn the vehicle trajectory data that cover all complexities, demonstrates enhanced expressiveness and yields superior results over a long prediction range of a few seconds. Thereafter, the subject has been extensively explored and continues to be under investigation in terms of its methodological aspects and concrete applications.

Given the vanishing-gradient issue of the recurrent neural network (RNN) (Yeon et al., 2019), the long short-term memory (LSTM) network has been the dominant method of trajectory prediction. In Deo et al. (2018), the authors present an LSTM-based prediction model for drivers' intentions and verify the precision of each trajectory prediction clue to generate the optimal trajectory. However, a motion model based on intention-guided motion cannot explain

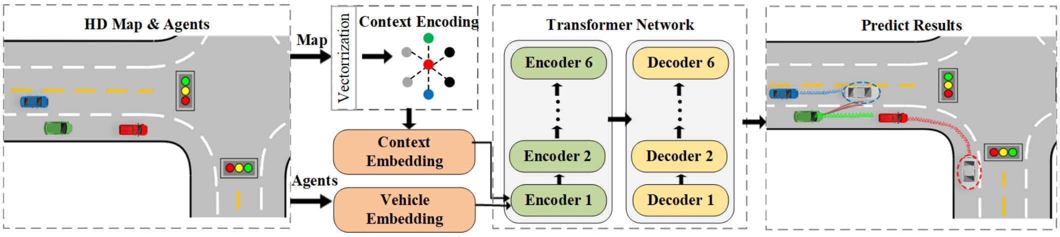

Figure 1. Proposed trajectory prediction architecture. The inputs of the prediction model on the left are the historical trajectories of surrounding vehicles and the scene context. The output is the distribution of the future trajectories.

drivers' behaviors because of the interdependencies amongst vehicles (Mozaffari et al., 2020). Social LSTM is introduced in Alahi et al. (2016), where participant interactions are connected through a social pooling layer to complete the prediction. Furthermore, in Hou et al. (2019), a structural LSTM network is proposed, where LSTM models are assigned to individual vehicles and the model automatically learns the dependencies among multiple interactive vehicles by sharing unit states and hidden states. These approaches factor in the interactions between multiple vehicles and result in a more accurate risk assessment. Nonetheless, the LSTM also has certain limitations caused by its time-dependent structure for the extraction of the sequence and memory mechanism (Dai et al., 2019). It is challenging to train long-term sequences, resulting in lower long-term prediction accuracy.

Efforts to resolve this dilemma have resulted in the combination of various neural networks. The convolutional neural network (CNN) has attracted attention for its capacity to extract spatial features. The CNN-LSTM (Mo et al., 2020) necessitates feeding the historical trajectories of vehicles and the interaction extracted between adjacent vehicles by employing the CNN in the LSTM decoder to generate the predicted future trajectory. To improve the robustness of the social pool layer, in Deo and Trivedi (2018), the authors exploit the convolutional social pool layer to learn the spatial correlation of the trajectory. Finally, in Lee et al. (2017), multiple predicted trajectories are generated on the basis of the RNN codec, which combines the scene context features extracted by the CNN for sorting and refining to construct the final trajectory.

Although numerous trajectory prediction methods have been developed in recent years, challenges persist, including computational complexity, reliance on specific environments or datasets, and a poor knowledge of vehicle interactions. There is a paucity of interactive prediction models in existing research that can properly account for both the relevant hazards in traffic scenarios and the degree of influence from adjacent vehicles.

2.3 Our approach

The evolution of trajectory patterns, with their intrinsic laws and external influencing factors being intertwined, exhibits a high degree of dynamism and complexity, posing a severe challenge to accurate prediction. On the one hand, we observe that certain trajectory features exist independently of the road framework, and the subtle correlation with road construction is often overlooked. On the other hand, some prediction algorithms solely concentrate on the trajectory dynamics of a single subject vehicle. Even when considering multi-vehicle trajectories to a limited extent, the impact of other vehicles on the target vehicle is often simplified. Therefore, it remains a formidable task to guarantee high prediction accuracy in such a complex and dynamic road environment.

Our task is to employ neural networks as a pivotal tool to integrate the dual challenges of trajectory prediction, thereby constructing an interactive model capable of profoundly capturing and deciphering the intricate spatiotemporal interactions among the various entities involved. In the context of transportation scenarios characterized by multiple vehicles operating simultaneously and in complex environments, we are convinced that leveraging multi-faceted input data and fully integrating the trajectories of multiple vehicles with environmental information for prediction make us more likely to achieve more precise predictions of vehicle positions in the long term.

In recent years, Transformer networks, originally successful in natural language processing (Vaswani et al., 2023), have also shown promise in vehicle trajectory prediction (Xu et al., 2021). In Quintanar et al. (2021), the author maintains the original structure of the Transformer and enhances prediction performance by incorporating the agent's orientation. However, this is a single-agent method that ignores context factors and interactions with other agents. As suggested by Li et al. (2020), a joint approach for target detection and motion prediction is built on a Transformer model, which extracts participants' states and spatial information from a bird's-eye view (BEV) feature map. The computation of shared tasks ensures optimal memory utilization, but it also raises prediction complexity.

Compared with the above methods, our approach involves a simplified spatial feature extraction method that cleverly transforms complex traffic scenes into ordered and semantically rich vector sequences using VectorNet technology. This facilitates efficient computation while achieving deep characterization and efficient encoding of scene context information. In addition, the VectorNet framework can flexibly handle different levels of road network data, demonstrating excellent scalability and providing strong support for large-scale trajectory prediction tasks.

Additionally, a comparison experiment is constructed to validate the model's prediction performance under identical settings. On outdoor datasets, it outperforms dual-LSTM models in terms of accuracy and shows closer prediction to the ground truth, with ADE and FDE being lowered by an average of 10 % and 11 %, respectively.

3 Prediction model

In this section, we introduce the traffic scene context encoding method, the model input and output, the encoder–decoder Transformer structure, and implementation details.

3.1 Traffic scene context modeling

Modeling the scene context information to capture the agent–road interaction is of paramount importance for trajectory prediction. VectorNet is adopted for its superior ability in extracting spatial locality. The core idea of VectorNet is to realize high-precision and low-redundancy expression of scene context information by abstracting map elements into vector sequences. The key to this transformation process lies in its ability to preserve the geometric structure and attribute information of the original map data while effectively addressing the limitations of traditional methods in handling irregular shapes and topological relationships.

Specifically, most elements in high-definition maps that pertain to traffic scenes, such as splines, closed-shaped polygons, and points, come with additional attribute information. All of these geographic entities and their properties can be approximated as vector sequences. We select a starting point and direction and then uniformly sample the key points along the splines with a consistent spatial distance. Vectorization processing is mainly divided into two layers:

1. Each lane line is represented by several polylines.
2. Each line is a piece of semantic information composed of multiple vectors.

We extract specific lane line features following vectorization steps, embed the sorting information into the vector, and constrain subgraph connectivity in accordance with the polyline grouping; the corresponding expressions are as follows:

$$\boldsymbol{v}_i^{\mathrm{t}} = \left[d_i^{\mathrm{s}}, d_i^{\mathrm{e}}, a_i, \mathrm{id}_{\mathrm{p}}\right], \tag{1}$$
$$P^{\mathrm{t}} = \left[\boldsymbol{v}_1^{\mathrm{t}}, \boldsymbol{v}_2^{\mathrm{t}}, \ldots, \boldsymbol{v}_i^{\mathrm{t}}\right], \tag{2}$$
$$V^{\mathrm{t}} = \left[P_1^{\mathrm{t}}, P_2^{\mathrm{t}}, \ldots, P_i^{\mathrm{t}}\right], \tag{3}$$

where $\boldsymbol{v}_i^{\mathrm{t}}$ represents each vector; d_i^{s} and d_i^{e} are the start and end coordinates of the vector; $\boldsymbol{a}_i$ represent the feature vector, which can include feature types such as road features and lane speed limits; and id_{p} signifies the index of the polyline to which the vector belongs. P^{t} is a set of lines representing lane line features, composed of multiple sets of vectors v_i^{t}. V^{t} is the vectorized result of the one-to-one correspondence of map labels.

To address the issue of varying lengths of lane lines, we standardize their lengths by segmenting roads longer than 100 m into shorter segments, thereby mitigating the issue of large vector length discrepancies that can lead to extraction errors. Consequently, each polyline contains a maximum of 100 vectors, and for those containing fewer than 100 vectors, zeros are padded to ensure a consistent data structure and to facilitate subsequent processing.

3.2 Problem formulation

In addressing the complex task of vehicle trajectory prediction, we opt for a systematic simplification by reframing it as a sequence prediction problem. To achieve this, we establish a multi-agent framework, wherein each agent, representing a distinct vehicle in the traffic scene, can process its current and previous positions (observation or motion history) through a Transformer network to output the predicted future positions.

Assuming that N agents are involved in a given traffic scene, the trajectory of agents is represented as $X^{\mathrm{t}} = \left[X_1^{\mathrm{t}}, X_2^{\mathrm{t}}, \ldots, X_N^{\mathrm{t}}\right]$, where $X_i^{\mathrm{t}} = \left(x_i^{\mathrm{t}}, y_i^{\mathrm{t}}\right)$ indicates the position of the vehicle from time 1 to t_{obs} in the top-down view map. Both the past trajectory X^{t} and the map information V^{t} serve as inputs to the model. The future trajectories based on ground truth from time $t_{\mathrm{obs}+1}$ to t_{pred} can be expressed as $Y^{\mathrm{t}} = \left[Y_1^{\mathrm{t}}, Y_2^{\mathrm{t}}, \ldots, Y_N^{\mathrm{t}}\right]$, while the output sequence from our model can be denoted as $\hat{Y}^{\mathrm{t}} = \left[\hat{Y}_1^{\mathrm{t}}, \hat{Y}_2^{\mathrm{t}}, \ldots, \hat{Y}_N^{\mathrm{t}}\right]$.

3.3 Transformer network

Transformer has dominated the field of capturing global dependencies between inputs and outputs, especially for long sequences, with the adoption of the classic encoder–decoder architecture. The encoder creates a representation of the interactions between map information and vehicles to enhance the memorability of the model, while the decoder is in charge of generating the future trajectory positions. Modules such as the embedding layers, position encoding (PE), self-attention layers, and fully connected feed-forward layers constitute the core of this algorithm.

Before entering the encoder, the input sequences are converted to vectors through the embedding layer, and the position-encoding blocks subsequently obtain positional information about the order of the input elements with the same dimension d_model as the embedding. The two components are then summed to form the final embedding as the input for the encoder–decoder. The position-encoding calculation

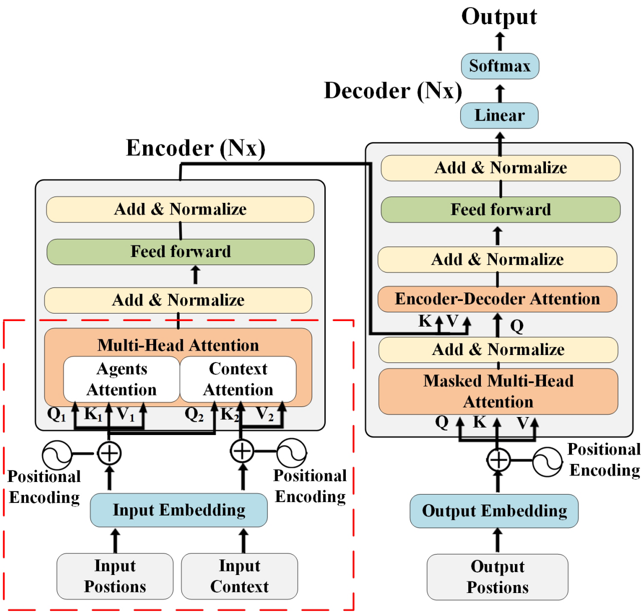

Figure 2. The modified Transformer architecture.

formula (Vaswani et al., 2023) is expressed by Eqs. (4)–(5):

$$\mathrm{PE}_{(\mathrm{pos},2i)} = \sin\left(\frac{\mathrm{pos}}{10\,000^{\frac{2i}{d_{\mathrm{model}}}}}\right), \tag{4}$$

$$\mathrm{PE}_{(\mathrm{pos},2i+1)} = \cos\left(\frac{\mathrm{pos}}{10\,000^{\frac{2i}{d_{\mathrm{model}}}}}\right), \tag{5}$$

where pos denotes the position, and i represents the dimension. Each position obtains the value combination of sine and cosine functions, with varying periods in the embedding dimension, thereby conveying unique positional information.

The forecasting model relies heavily on two indispensable inputs: the historical trajectory data and the contextual map, both of which serve as vital information sources. Initially, each encoder's input undergoes meticulous processing by a self-attention layer, designed to uncover long-term dependencies within by calculating self-attention across embeddings at various time points. As depicted in Fig. 2, these self-attention layers are intricately bifurcated into two distinct parts: the agent attention layer and the scene attention layer.

The agent attention layer specifically captures the intricate interactions between the historical trajectories of individual agents, facilitating a deeper understanding of their movement patterns. In parallel, the scene attention layer encodes the map data, intertwining it with the agents' historical trajectories. This integration provides crucial environmental cues that not only enhance prediction accuracy but also profoundly influence the agents' decision-making processes and subsequent movements, underscoring its significance in refining our forecasting capabilities.

The self-attention layer maps a set of queries, $\mathbf{Q}$; a set of keys, $\mathbf{K}$; and a set of values, $\mathbf{V}$, into an output vector. $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ represent the query matrix, key matrix, and value matrix, respectively. These matrices, denoted as $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, are derived from the input vector $\boldsymbol{x}$ through three different weight matrices. The attention matrix can be obtained by the dot product of the query and the key vector by $1/\sqrt{d_k}$. When the d_k value is large, a scaling operation is performed to prevent convergence issues. The result is then passed through the softmax function, represented by the formula of Eq. (6):

$$\text{attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}. \tag{6}$$

The multi-head attention mechanism is used to establish multiple subspaces, allowing the model to focus on various aspects of information and to obtain subspace information from different locations through multiple calculations. First, after linear transformation, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are executed in parallel h times in relation to the scaled dot product attention mechanism, which is called the multi-head h. Then, the results of h times in relation to the scaled dot product attention are connected and transformed into the expected dimension through linear transformation to obtain the final value of the multi-head attention, corresponding to the following formulas in Eqs. (7)–(8):

$$\text{multi-head}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{concat}(\text{head}_1,\ldots,\text{head}_\text{h})\mathbf{W}^\text{O}, \tag{7}$$

$$\text{head}_i = \text{attention}\left(\mathbf{Q}\mathbf{W}_i^\text{Q}\mathbf{K}\mathbf{W}_i^\text{K}\mathbf{V}\mathbf{W}_i^\text{V}\right), \tag{8}$$

where $\mathbf{W}_i^\text{Q}$, $\mathbf{W}_i^\text{K}$, $\mathbf{W}_i^\text{V}$, and $\mathbf{W}^\text{O}$ represent the query, key, value, and output weight matrices.

The distinction between the scene attention layer and the agent attention layer stems from their respective input vectors. Specifically, the agent attention layer incorporates vehicle position embeddings for $\mathbf{Q}_1$, $\mathbf{K}_1$, and $\mathbf{V}_1$, whereas the scene attention layer utilizes vehicle position embeddings for $\mathbf{Q}_2$ and integrates scene information embeddings for $\mathbf{K}_2$ and $\mathbf{V}_2$. In the intricate web of multi-vehicle interactions, each vehicle's paramount objective is to distill decisive feature signatures from the ambient environment and the myriad of participants. This intricate feature extraction process can be meticulously segmented into four phases: gathering distinctive features, precisely delineating feature sets, efficiently querying identifiers, and consolidating results. Each attention head leverages input linear projections to generate diverse feature selections, ultimately constructing $\mathbf{V}$. To pinpoint these salient features, $\mathbf{K}$ is associated with each value $\mathbf{V}$, fostering a profound comprehension and enabling precise responses to complex traffic scenarios.

The structure of the decoder bears a resemblance to that of the encoder. This encoder–decoder attention works similarly to the multi-head attention, except for the fact that the former creates a query $\mathbf{Q}$ through the layer beneath it and obtains the key $\mathbf{K}$ and value $\mathbf{V}$ from the output of the encoder.

The masked attention layer performs a mask operation in response to attention from subsequent positions.

Another sublayer of the encoder and decoder is a fully connected feed-forward network, allowing for nonlinear transformations. This includes two layers of linear transformations with a rectified linear unit (ReLU) activation in between. Each sub-module of the main framework is accompanied by a residual connection and normalization layer, which effectively ameliorates the issue of gradient vanishing in deep models and accelerates convergence. Finally, the predicted results are outputted through linear transformation and the softmax function.

3.4 Implementation details

In this study, the Transformer network consists of eight attention heads, six layers, and $d_{\text{model}} = 512$. The layer normalization sets eps $= 1 \times 10^{-5}$ for the output value of multi-head attention. To mitigate over-fitting during the training process, a dropout rate of 0.2 is set for the residual network. Considering the fact that trajectory prediction constitutes a regression problem, the loss function is defined by the L2 loss between the predicted output trajectory and the ground truth trajectory. Backpropagation training of the network is conducted using the Adam optimizer, with a learning rate set to 0.0001 and a batch size set to 128. The network training and evaluation are facilitated using PyTorch.

4 Experiments

In this section, our approach is evaluated based on the publicly available Argoverse dataset, a large-scale autonomous driving dataset containing 320 h of data and rich map information (Chang et al., 2019). The recorded vehicles are mainly distributed across complex traffic made up of intersections, turns, and lane changes, with a total of 324 557 5 s sequences. For trajectory length, we set $t_{\text{obs}} = 2$ s and $t_{\text{pred}} = 3$ s, with a time interval of 0.1 s. The training, test, and validation sets contain 205 942, 78 143, and 39 472 sequences, respectively. Quintessentially, the "agent" in each sequence is a vehicle, but other tracked objects can be vehicles, pedestrians, or bicycles. Figure 3 illustrates sampling locations and the distribution of these sequences.

4.1 Evaluation metrics and baselines

In terms of the multi-modal prediction based on Argoverse, four separate metrics are introduced for evaluation. The common metrics in the literature are as follows:

> *Final displacement error (FDE).* This metric measures the distance between the final predicted position and the corresponding ground truth position at that specific time point without a consideration of the prediction errors that occur in other time steps in the prediction horizon.
>
> *Average displacement error (ADE).* ADE measures the average discrepancy between the predicted position at each time step and the ground truth position for that particular step.
>
> *MADE and MFDE.* These metrics refer to the minimum ADE and FDE among multiple predictions.

ADE and FDE measurements are standard assessment metrics in trajectory prediction. They clearly measure the distance error between predicted and ground truth locations; the following observations are made based on these metrics:

1. These indicators are comparatively intuitive and enable us to directly quantify the performance of the prediction algorithm by measuring ADE at a particular time interval and FDE at a defined time; therefore, they are commonly used to evaluate the performance of prior techniques.
2. Standard ADE and FDE metrics are applicable to any interaction scenario and do not take into consideration any form of prediction confidence, such as the possibility or rank of each trajectory.

For multi-modal evaluation, we also introduce MADE and MFDE to evaluate the single best prediction. A series of experiments are conducted to compare our model with a wide range of baselines, which include the following:

> *TNT (Zhao et al., 2020).* This model encodes the map context to capture the agent–road and agent–agent interactions while using multi-layer perception (MLP) to predict the future trajectory of the agent.
>
> *LaneRCNN (Zeng et al., 2021).* LaneRCNN proposes LaneRoI, which models participant interactions with a graph-based interactor and predicts the final position in a fully convolutional manner.
>
> *Trajformer (Bhat et al., 2020).* Trajformer unifies the Transformer encoder structure with the normalized flow-based decoder structure for multi-modal trajectory prediction.
>
> *Dual-LSTM models (Xin et al., 2018).* This approach employs two LSTM networks – one for intent identification of the sequential trajectory and another for future trajectory prediction.

4.2 Quantitative results and analyses

On the basis of this dataset, a comparison is made between our proposed approach and those presented in the previous section. The MADE and MFDE values of these models are presented in Table 1. The findings indicate that dual-LSTM models exhibit undisputed superiority in terms of lateral position error. On the other hand, the Trajformer model, with its

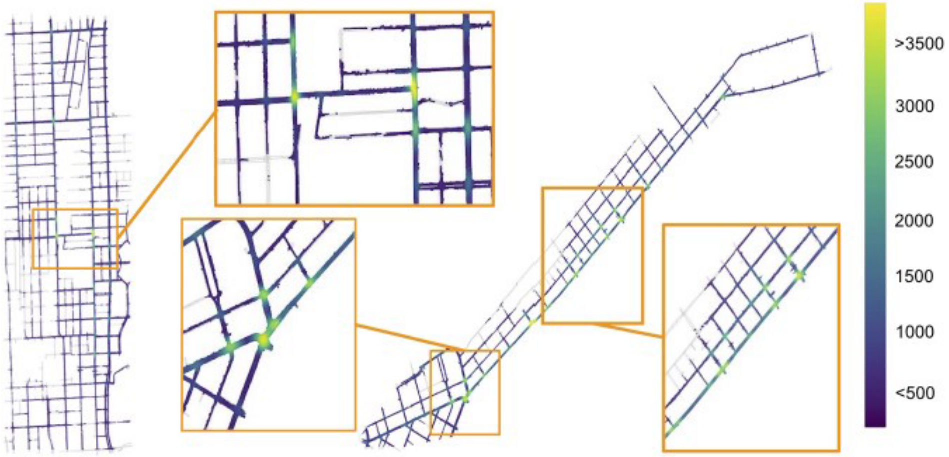

Figure 3. Distribution of trajectories. The colors indicate the number of trajectories on the maps of Miami (left) and Pittsburgh (right), focusing on junctions and busy roads.

Table 1. Comparison with baselines based on the Argoverse. The bold values represent the results corresponding to the model proposed in this paper.

Baselines	MADE (m)	MFDE (m)
TNT	0.73	1.29
LaneRCNN	0.77	1.19
Trajformer	0.62	0.71
Dual-LSTM models	0.58	0.74
Our approach	**0.58**	**0.62**

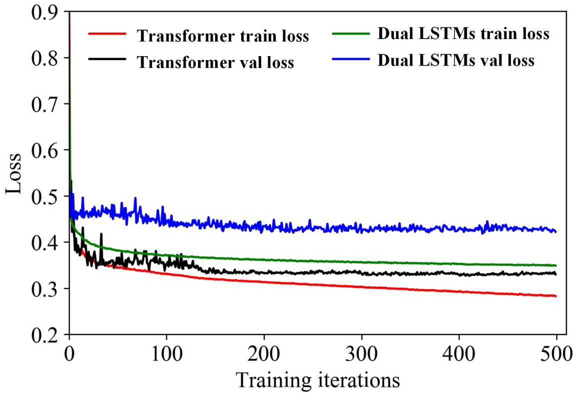

Figure 4. The loss variation of the two models.

unique Transformer encoding architecture, has also demonstrated considerable predictive performance, which is firmly in the forefront. However, it is worth mentioning that the interactive prediction model based on Transformer architecture proposed in this paper not only reduces MADE and MFDE by 12 % and 31 % on average but also reveals a profound insight: the consideration of surrounding vehicle and environmental scene information is indispensable when predicting vehicle trajectories, and the excellent ability of the Transformer model based on the multi-head attention mechanism to simulate complex motion interactions between vehicles provides a crucial reference for accurate prediction.

We employ the dual-LSTM model as a benchmark in one experiment. Apart from the internal model structure, all aspects related to the two models remain identical, such as inputs, outputs, the number of layers, and the training process. No human intervention occurs during the training process. The loss function values are plotted in Fig. 4. It can be observed that, when the loss starts to converge after about 150 iterations, the improved Transformer model has a smoother training process and can converge the loss value to a satisfactory low value. This phenomenon strongly confirms the superiority of the model in terms of training difficulty and convergence speed. In Fig. 5, most trajectories exhibit a final displacement error of less than 1 m within 3 s. Notably, the Transformer model achieves an even smaller error value. These visualized data not only provide solid evidence for our theoretical inference but also vividly demonstrate the extraordinary strength of the model in terms of prediction accuracy.

Our approach can model the internal relationships between different inputs, and the specific comparison results are listed in Tables 2 and 3. Since driving actions are largely influenced by real-time changes in traffic circumstances, map information provides fundamental route structures and semantic guidance conducive to long-term prediction. Vectorizing the semantic road map into context information as input into the Transformer (map + agents) yields superior outcomes. Our model's trajectory prediction accuracy is superior to that of dual-LSTM models, with more precise error control in endpoint displacement prediction; furthermore, ADE and FDE are reduced by 10 % and 11 %, respectively. From these re-

Table 2. Average displacement error. The bold values represent the results corresponding to the model proposed in this paper.

Method	Lateral position error (m)			Longitudinal position error (m)			Average displacement error (m)		
Prediction horizon	0–1 s	0–2 s	0–3 s	0–1 s	0–2 s	0–3 s	0–1 s	0–2 s	0–3 s
Transformer agents	0.58	0.73	1.03	0.53	1.03	1.49	0.79	1.27	1.82
Dual-LSTM map + agents	0.46	0.51	0.55	0.47	0.52	0.57	0.66	0.73	0.79
Our approach’s map + agents	**0.42**	**0.44**	**0.49**	**0.43**	**0.46**	**0.51**	**0.60**	**0.64**	**0.71**

Table 3. Final displacement error. The bold values represent the results corresponding to the model proposed in this paper.

Method	Lateral position error (m)			Longitudinal position error (m)			Final displacement error (m)		
Prediction horizon	1 s	2 s	3 s	1 s	2 s	3 s	1 s	2 s	3 s
Transformer agents	0.64	1.22	1.95	0.80	1.97	2.51	0.95	2.28	2.97
Dual-LSTM map + agents	0.51	0.58	0.69	0.53	0.59	0.72	0.74	0.83	0.99
Our approach’s map + agents	**0.43**	**0.50**	**0.66**	**0.44**	**0.52**	**0.69**	**0.62**	**0.73**	**0.95**

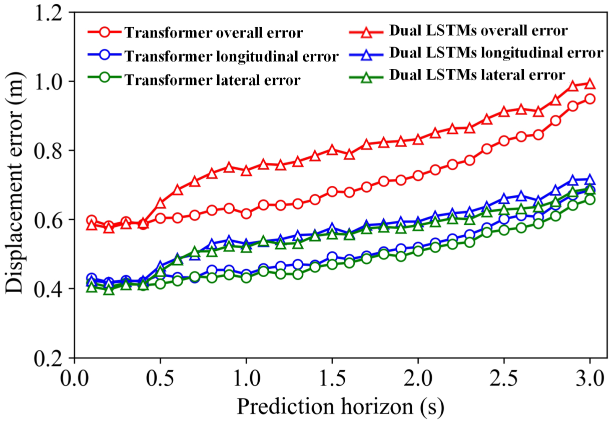

Figure 5. Final displacement error of prediction horizon.

sults, it is clear that the superiority of the model lies in the introduction of an improved agent attention layer and an improved scene attention layer, which enables the model to capture and understand complex traffic scene information and the intricate and subtle interactions between vehicles more deeply and comprehensively, ensuring the adaptability and prediction accuracy of the model in relation to various complex situations.

On the other hand, the average lateral and longitudinal displacement errors remain consistently low, with values of 0.6 m during the 3 s prediction period, indicating the excellent accuracy and reliability of the prediction model. As the prediction time increases and the estimated horizon extends, future vehicle behavior tends to be more uncertain, with an increase in both lateral and longitudinal final position errors. However, within a relatively short prediction period, this deviation still fluctuates within an acceptable error threshold. Further analysis reveals that the key to improving long-term prediction accuracy may lie in enriching input features and incorporating interactive effects between vehicles. These elements enable the model to capture the subtle changes in individual vehicle behaviors in complex road conditions more finely.

4.3 Qualitative results and analyses

It is arduous to precisely evaluate the model’s effectiveness solely based on average displacement error analysis because of the concealment potential of lane change behavior predictions. As a consequence, qualitative results are composed of the trajectory visualization of the dual-LSTM models and the improved Transformer model, which better expresses the model performance in a more intuitive way. Figure 6 depicts the error distribution of trajectory prediction in various traffic scenarios, covering typical driving modes such as straight, left turn, and right turn. When the vehicle is driving straight, the improved Transformer model shows prediction accuracy that is almost identical to the ground truth trajectory. Faced with lane-changing actions such as left and right turns, although the final position error predicted by the model has increased, it still demonstrates its excellent predictive ability, clearly surpassing the dual-LSTM model.

The Transformer model significantly outperforms the dual-LSTM models in sharp bends, as shown by the visualized findings. It indirectly indicates that our approach has exhibited favorable performance in terms of the extraction of the temporal and spatial characteristics of long-term sequences and in providing more accurate predictions for a given observation trajectory aided by maps and road rules.

4.4 Ablation studies

This section endeavors to delve deeper into analyzing the influence of diverse parameters on the prediction accuracy of the Transformer model subsequent to reporting and contrast-

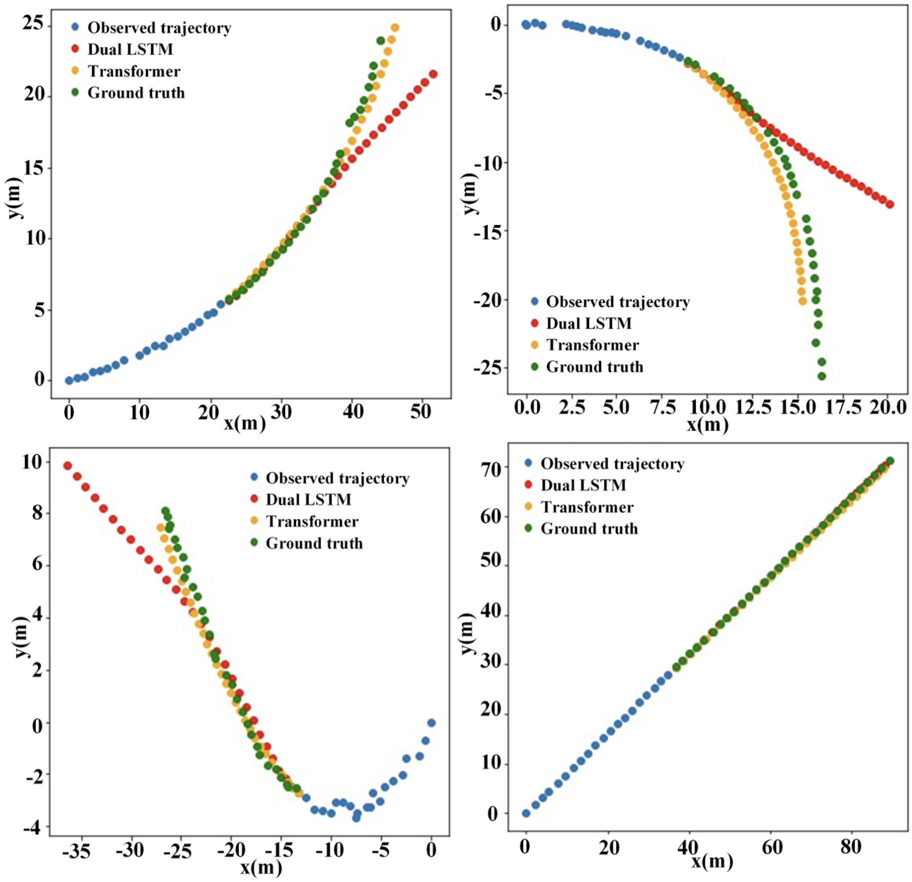

Figure 6. Trajectory visualization. List of the trajectory prediction results of the dual-LSTM models and Transformer model in three common traffic situations for comparison.

ing the existing methodologies. As is evident from Table 4, the utilization of varying attention heads subtly modulates the prediction performance, where the model equipped with eight attention heads demonstrates a marginal reduction in prediction error. Each attention head encapsulates a unique set of weights, primarily influencing the interactions between the ego vehicle and its surroundings.

Furthermore, our investigation into the impact of encoder and decoder layers on model efficacy revealed that, from a qualitative and quantitative standpoint, there exists no noteworthy disparity between the 6-layer and 12-layer architectures. However, a notable divergence emerges when assessing the time efficiency for single-sample prediction. More precisely, the adoption of a 12-layer encoder–decoder configuration significantly escalates the computational costs, thereby posing a formidable challenge for real-time prediction applications where promptness is paramount.

After weighing the relationship between computational complexity and prediction error, we have decided to adopt an eight-head attention configuration with a six-layer encoder and decoder in our experiments. This parameter not only maximizes model prediction accuracy but also fully considers the need for computational efficiency, striving to achieve rapid and accurate prediction of vehicles in complex traffic environments while ensuring real-time operation of the model and avoiding adverse effects on driving decisions caused by calculation delays.

5 Conclusion

In this paper, an end-to-end trajectory prediction model based on Transformer has been proposed to address long-term prediction accuracy in complex traffic environments. Multi-head attention optimization based on scene context and vehicle position knowledge generates interactions between maps and agents, as well as among agents themselves. The effectiveness of the proposed algorithm has been evaluated on the basis of an outdoor dataset to achieve higher precision and a closer distance to the real destination. Accordingly, the proposed approach demonstrates the capacity to handle complex traffic scenarios for different road users. In future work, to validate the broad applicability of our algorithm, we plan to conduct cross-dataset evaluations and research in the field of pedestrian trajectory prediction. Specifically, we will apply

Table 4. Ablation studies on different parameters of the improved Transformer. The bold values represent the results corresponding to the model proposed in this paper.

Parameters	Average displacement error (m)			Final displacement error (m)			MADE	MFDE	Prediction time
Prediction horizon	0–1 s	0–2 s	0–3 s	1 s	2 s	3 s	(m)	(m)	(s)
$h=2$, layers = 6	0.62	0.69	0.79	0.65	0.84	1.18	0.58	0.65	0.159
$h=4$, layers = 6	0.65	0.77	0.93	0.71	1.03	1.45	0.60	0.71	0.165
$\mathbf{h=8}$, layers = 6	**0.60**	**0.64**	**0.71**	**0.62**	**0.73**	**0.95**	**0.58**	**0.62**	**0.152**
$h=8$, layers = 12	0.62	0.65	0.71	0.63	0.73	0.91	0.60	0.63	0.372

our algorithm to additional datasets such as nuScenes and KITTI, which are renowned for their rich scene diversity, high-quality annotations, and widespread industry recognition. By testing our algorithm on these diverse datasets, we aim to demonstrate its generalization capabilities and robustness under various environments and conditions, thereby proving its potential and value in practical applications.

Code and data availability. The data and code that support the findings of this study are available upon request from the first author.

Author contributions. RX built the model and wrote the paper. JL constructed the overall framework of the paper. SZ was responsible for the experiment. LL completed the corresponding data analysis. HL performed the data analysis. GR performed the validation. XT performed the visualization of data.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. The authors express their deepest gratitude to the editors and anonymous reviewers for their valuable time, insightful suggestions, and constructive comments, which have significantly enhanced the quality of this work.

Financial support. This research has been supported by the Chongqing Municipal Fiscal Science and Technology Innovation (grant no. KYLX20240500075).

Review statement. This paper was edited by Daniel Condurache and reviewed by four anonymous referees.

References

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., and Savarese, S.: Social LSTM: Human Trajectory Prediction in Crowded Spaces, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016, IEEE, 961–971, https://doi.org/10.1109/CVPR.2016.110, 2016.

Bhat, M., Francis, J., and Oh, J.: Trajformer: Trajectory Prediction with Local Self-Attentive Contexts for Autonomous Driving, arXiv [preprint], https://doi.org/10.48550/arXiv.2011.14910, 30 November 2020.

Chang, M. F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., and Ramanan, D.: Argoverse: 3D Tracking and Forecasting With Rich Maps, arXiv [preprint], https://doi.org/10.48550/arXiv.1911.02620, 6 November 2019.

Cui, Z., Henrickson, K., Ke, R., and Wang, Y.: Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, IEEE T. Intell. Transp., 21, 4883–4894, https://doi.org/10.1109/TITS.2019.2950416, 2020.

Dai, S., Li, L., and Li, Z.: Modeling vehicle interactions via modified LSTM models for trajectory prediction, IEEE Access, 7, 38287–38296, https://doi.org/10.1109/ACCESS.2019.2907000, 2019.

Deo, N. and Trivedi, M. M.: Convolutional social pooling for vehicle trajectory prediction, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 1–22 June 2018, IEEE, 1549–1548, https://doi.org/10.1109/CVPRW.2018.00196, 2018.

Deo, N., Rangesh, A., and Trivedi, M. M.: How would surround vehicles move? a unified framework for maneuver classification and motion prediction, IEEE Transactions on Intelligent Vehicles, 3, 129–140, https://doi.org/10.1109/TIV.2018.2804159, 2018.

Djuric, N., Radosavljevic, V., Cui, H., Nguyen, T., Chou, F.-C., Lin, T.-H., Singh, N., and Schneider, J.: Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving, in: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020, IEEE, 2084–2093, https://doi.org/10.1109/WACV45572.2020.9093332, 2020.

Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., and Schmid, C.: Vectornet: Encoding hd maps and agent dynamics from vectorized representation, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–16 june 2020, IEEE, 11525–11530, https://doi.org/10.1109/CVPR42600.2020.01154, 2020.

Hou, L., Xin, L., Li, S. E., Cheng, B., and Wang, W.: Interactive trajectory prediction of surrounding road users for autonomous driving using structural-LSTM network, IEEE T. Intell. Transp., 21, 4615–4625, https://doi.org/10.1109/TITS.2019.2942089, 2019.

Hubmann, C., Schulz, J., Becker, M., Althoff, D., and Stiller, C.: Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction, IEEE Transactions on Intelligent Vehicles, 3, 5–17, https://doi.org/10.1109/TIV.2017.2788208, 2018.

Jeong, Y., Kim, S., and Yi, K.: Surround vehicle motion prediction using LSTM-RNN for motion planning of autonomous vehicles at multi-lane turn intersections, IEEE Open Journal of Intelligent Transportation Systems, 1, 2–14, https://doi.org/10.1109/OJITS.2020.2965969, 2020.

Keller, C. G. and Gavrila, D. M.: Will the pedestrian cross? A study on pedestrian path prediction, IEEE T. Intell. Transp., 15, 494–506, https://doi.org/10.1109/TITS.2013.2280766, 2013.

Lee, N., Choi, W., Vernaza, P., Choy, C. B., Torr, P. H., and Chandraker, M.: Desire: DESIRE:Distant future prediction in dynamic scenes with interacting agents, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017, IEEE, 2165–2174, https://doi.org/10.1109/CVPR.2017.233, 2017.

Li, L. L., Yang, B., Liang, M., Zeng, W., Ren, M., Segal, S., and Urtasun, R.: End-to-end contextual perception and prediction with interaction transformer, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021, IEEE, 5784–5791, https://doi.org/10.1109/IROS45743.2020.9341392, 2020.

Min, K., Yeon, K., Jo, Y., Sim, G., Sunwoo, M., and Han, M.: Vehicle deceleration prediction based on deep neural network at braking conditions, Int. J. Automot. Techn., 21, 91–102, https://doi.org/10.1007/s12239-020-0010-2, 2020.

Mo, X., Xing, Y., and Lv, C.: Interaction-aware trajectory prediction of connected vehicles using cnn-lstm networks, in: IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Singapore, 18–21 October 2020, IEEE, 5057–5062, https://doi.org/10.1109/IECON43393.2020.9255162, 2020.

Mozaffari, S., Al-Jarrah, O. Y., Dianati, M., Jennings, P., and Mouzakitis, A.: Deep learning-based vehicle behavior prediction for autonomous driving applications: A review, IEEE T. Intell. Transp., 23, 33–47, https://doi.org/10.1109/TITS.2020.3012034, 2020.

Neto, F. D. N., de Souza Baptista, C., and Campelo, C. E.: Combining Markov model and prediction by partial matching compression technique for route and destination prediction, Knowl.-Based Syst., 154, 81–92, https://doi.org/10.1016/j.knosys.2018.05.007, 2018.

Phan-Minh, T., Grigore, E. C., Boulton, F. A., Beijbom, O., and Wolff, E. M.: CoverNet: Multimodal behavior prediction using trajectory sets, arXiv [preprint], https://doi.org/10.48550/arXiv.1911.10298, 1 April 2020.

Quintanar, A., Fernández-Llorca, D., Parra, I., Izquierdo, R., and Sotelo, M.: Predicting vehicles trajectories in urban scenarios with transformer networks and augmented information, in: 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021, IEEE, 1051–1056, https://doi.org/10.1109/IV48863.2021.9575242, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I.: Attention is all you need, Advances in neural information processing systems, arXiv [preprint], https://doi.org/10.48550/arXiv.1706.03762, 2 August 2023.

Wang, L.-l., Chen, Z.-g., and Wu, J.: Vehicle trajectory prediction algorithm in vehicular network, Wirel. Netw., 25, 2143–2156, https://doi.org/10.1007/s11276-018-1803-3, 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y.: A comprehensive survey on graph neural networks, IEEE T. Neur. Net. Lear., 32, 4–24, https://doi.org/10.1109/TNNLS.2020.2978386, 2020.

Xie, G., Gao, H., Qian, L., Huang, B., Li, K., and Wang, J.: Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models, IEEE T. Ind. Electron., 65, 5999–6008, https://doi.org/10.1109/TIE.2017.2782236, 2017.

Xin, L., Wang, P., Chan, C.-Y., Chen, J., Li, S. E., and Cheng, B.: Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018, IEEE, 1441–1446, https://doi.org/10.1109/ITSC.2018.8569595, 2018.

Xu, J., Xiao, L., Zhao, D., Nie, Y., and Dai, B.: Trajectory Prediction for Autonomous Driving with Topometric Map, arXiv [preprint], https://doi.org/10.48550/arXiv.2105.03869, 9 May 2021.

Yao, Y., Atkins, E., Johnson-Roberson, M., Vasudevan, R., and Du, X.: BiTraP: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation, IEEE Robotics and Automation Letters, 6, 1463–1470, https://doi.org/10.1109/LRA.2021.3056339, 2021.

Ye, N., Wang, Z.-Q., Malekian, R., Lin, Q., and Wang, R.-C.: A method for driving route predictions based on hidden Markov model, Math. Probl. Eng., 2015, 1–12, https://doi.org/10.1155/2015/824532, 2015.

Yeon, K., Min, K., Shin, J., Sunwoo, M., and Han, M.: Ego-vehicle speed prediction using a long short-term memory based recurrent neural network, Int. J. Automot. Techn., 20, 713–722, https://doi.org/10.1007/s12239-019-0067-y, 2019.

Zeng, W., Liang, M., Liao, R., and Urtasun, R.: LaneRCNN: Distributed representations for graph-centric motion forecasting, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021, IEEE, 532–539, https://doi.org/10.1109/IROS51168.2021.9636035, 2021.

Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., Shen, Y., Shen, Y., Chai, Y., and Schmid, C.: TNT: Target-driveN Trajectory Prediction, arXiv [preprint], https://doi.org/10.48550/arXiv.2008.08294, 21 August 2020.