

Article

# Dynamic Perception-Based Vehicle Trajectory Prediction Using a Memory-Enhanced Spatio-Temporal Graph Network

Zhiming Gui <sup>\*</sup>, Xin Wang and Wenzheng Li

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; wang\_xin@emails.bjut.edu.cn (X.W.); liwww@bjut.edu.cn (W.L.)

\* Correspondence: zmgui@bjut.edu.cn

**Abstract:** In the realm of intelligent transportation systems, accurately predicting vehicle trajectories is paramount for enhancing road safety and optimizing traffic flow management. Addressing the impacts of complex traffic environments and efficiently modeling the diverse behaviors of vehicles are the key challenges at present. To achieve precise prediction of vehicle trajectories, it is essential to fully consider the dynamic changes in traffic conditions and the long-term dependencies of time-series data. In response to these challenges, we propose the Memory-Enhanced Spatio-Temporal Graph Network (MESTGN), an innovative model that integrates a Spatio-Temporal Graph Convolutional Network (STGCN) with an attention-enhanced Long Short-Term Memory (LSTM)-based sequence to sequence (Seq2Seq) encoder–decoder structure. MESTGN utilizes STGCN to capture the complex spatial dependencies between vehicles and reflects the interactions within the traffic network through road traffic data and network topology, which significantly influences trajectory prediction. Additionally, the model focuses on historical vehicle trajectory data points using an attention-weighted mechanism under a traditional LSTM prediction architecture, calculating the importance of critical trajectory points. Finally, our experiments conducted on the urban traffic dataset ApolloSpace validate the effectiveness of our proposed model. We demonstrate that MESTGN shows a significant performance improvement in vehicle trajectory prediction compared with existing mainstream models, thereby confirming its increased prediction accuracy.

**Keywords:** vehicle trajectory prediction; memory-enhanced spatio-temporal graph network; long short-term memory network; Seq2Seq encoder–decoder



**Citation:** Gui, Z.; Wang, X.; Li, W. Dynamic Perception-Based Vehicle Trajectory Prediction Using a Memory-Enhanced Spatio-Temporal Graph Network. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 172. <https://doi.org/10.3390/ijgi13060172>

Academic Editors: Wolfgang Kainz, Peng Peng, Shu Wang, Maryam Lotfian, Feng Lu and Yunqiang Zhu

Received: 15 March 2024  
Revised: 9 May 2024  
Accepted: 22 May 2024  
Published: 24 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

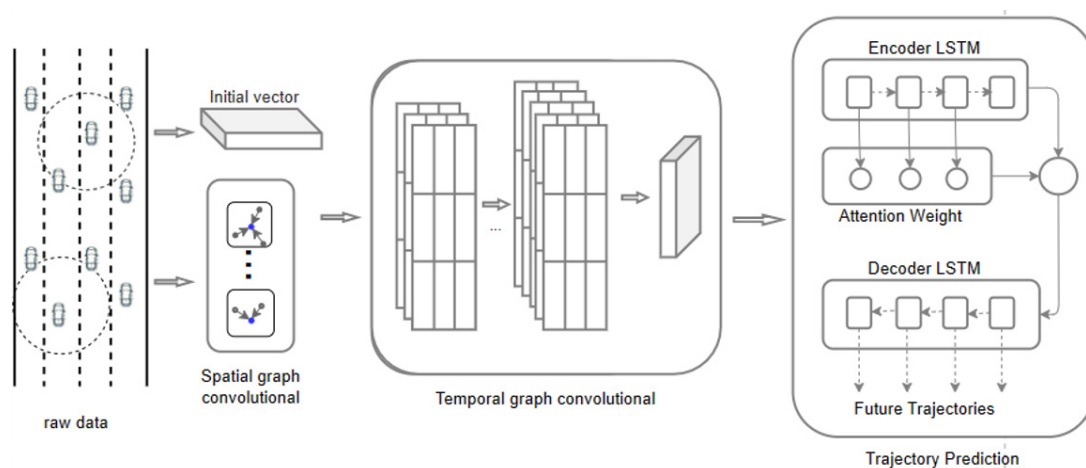
As urbanization accelerates and traffic volumes surge, traditional traffic management systems are increasingly overwhelmed by the demands of complex traffic patterns. Issues like congestion, frequent accidents, and environmental pollution are prompting researchers to explore more intelligent and efficient solutions. In response to these challenges, deep learning technology has become a key tool, offering innovative approaches for developing intelligent transportation systems. Vehicle trajectory prediction based on deep learning [1] has become an important research field for addressing contemporary transportation challenges. With the continuous evolution of autonomous driving technology, the accurate prediction of vehicle trajectories has become an indispensable task with wide-ranging application prospects. Motion trajectory data [2] often contain rich spatio-temporal information, requiring in-depth analysis to achieve trajectory prediction, ensuring the safety and efficiency of driving.

However, predicting vehicle trajectories accurately poses a highly challenging task due to the characteristics of trajectory data, such as their temporal nature, multidimensionality, large volume, trends, and complexity, which often involve long sequential data [3]. In the early stages of research on vehicle trajectory prediction algorithms, many scholars adopted approaches based on establishing kinematic or dynamic models to predict the

trajectories of surrounding vehicles, considering vehicles as objects governed by physical laws [4]. Vehicle trajectory prediction often relies on the assumption that specific vehicle states will either stay the same or change at a consistent rate moving forward [5]. However, this approach fails to utilize environmental information and spatio-temporal interaction information for modeling, resulting in insufficient prediction accuracy [6]. In recent years, with the continuous development of artificial intelligence and hardware computing power, an increasing number of researchers have adopted data-driven approaches for vehicle trajectory prediction. Data-driven trajectory prediction methods predict the possible future trajectories of the target vehicle based on a segment of its historical trajectory [7]. These data-driven models can be divided into classical machine learning models [8,9] and neural-network-based deep learning models [10,11].

This study introduces an innovative trajectory prediction network model that integrates an interaction-aware spatio-temporal graph with an LSTM-based attention enhancement mechanism, applying the Seq2Seq framework for trajectory prediction output, as shown in Figure 1. By integrating these two core components, the model effectively captures the spatial complexities among vehicles while fully accounting for the temporal continuity and long-term dependencies in their trajectories, thereby achieving higher accuracy and reliability in vehicle trajectory prediction. Specifically, our main contributions include the following:

- (1) We propose a novel vehicle trajectory prediction framework that integrates a Spatio-Temporal Graph Convolutional Network (STGCN) with an attention-enhanced LSTM trajectory prediction network into a Seq2Seq architecture, accurately capturing the spatio-temporal dependencies in vehicle trajectory data.
- (2) The Memory-Enhanced Spatio-Temporal Graph Network (MESTGN) model focuses on analyzing the interactions between vehicles and deeply captures and understands these interactions' complex spatial relationships through the integration of a spatio-temporal graph. It accurately identifies and addresses the dynamic interplay between vehicles, thus achieving higher precision and reliability in vehicle trajectory prediction.
- (3) We conducted experiments on the urban traffic dataset ApolloSpace, validating our model's performance advantage in vehicle trajectory prediction. The experimental results validate our model's effectiveness in this field.



**Figure 1.** The architecture of the Spatial–Temporal Graph Convolutional Network.

The subsequent sections of this paper are organized as follows: Section 2 reviews related work. Section 3 provides a detailed exposition of our model architecture and methodology. Section 4 outlines the proposed design scheme. Section 5 showcases the experimental settings and result analysis. Section 5 summarizes the entire document. Finally, Section 6 discusses future research directions.

## 2. Related Works

The current methods for vehicle trajectory prediction are primarily categorized into two types: classification-based maneuver models that use traditional methods and perception models that employ deep learning to analyze vehicle behavior.

Classification-based maneuver models concentrate on recognizing the maneuvering actions of vehicles by treating these actions as a classification problem, relying on specific vehicle features. These models typically consist of two main components: a maneuver identification module and a trajectory prediction module tailored to specific maneuvers. The identification module functions as a classifier, utilizing historical trajectory data, motion status, and contextual clues from the surrounding environment to categorize different maneuvering behaviors. Various classifiers have been applied, including heuristic methods [12], Bayesian networks [13], hidden Markov models [14], random forest classifiers [15], and Gaussian process models [16]. These classifiers determine the kinematic parameters of the vehicle's maneuver based on its historical trajectory data, thus judging the vehicle's maneuvering behavior. A regression trajectory prediction model first forecasts the vehicle's future path. The trajectory prediction module then determines the future position based on the maneuver category.

However, traditional traffic prediction methods are typically based on modeling under simple scenario assumptions. When facing complex traffic environments, the performance of these methods may significantly degrade. Traditional approaches rely on handcrafted features that need to be adapted for different traffic conditions. Additionally, their precision tends to decrease more noticeably with long-term trajectory forecasts. This is because long-term prediction requires considering more uncertainty and dynamic changes, making it challenging to model accurately. Therefore, traditional methods may not provide satisfactory performance for complex traffic environments and long-term trajectory prediction tasks.

In vehicle perception, predicting vehicle trajectories can be viewed as a sequence classification or sequence generation task [17]. Sequence classification maps trajectory inputs to predefined categories or labels, while sequence generation creates new sequences related to these inputs, such as future vehicle trajectories. In the vehicle perception discussed in this paper, the key is to extract features from vehicle trajectory data in both temporal and spatial dimensions [18] to help the model understand the interaction relationships between vehicles and temporal dependencies. Before forwarding the vehicle trajectories to the predictive neural network, the raw trajectory data needs to be represented in an appropriate form. Common representations include sequential points [19], occupancy grids [20], and rasterized images [21]. Based on these representations, various network components can be used to extract temporal and spatial contexts.

For temporal correlation analysis, both recurrent and convolutional neural networks can be leveraged. Owing to their inherent memory mechanisms, Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Gated Recurrent Units (GRUs) are particularly well-suited to sequence data, enabling them to capture complex social interactions and temporal dependencies among vehicles effectively. These models analyze historical trajectory data to comprehend how vehicles interact and influence each other's actions at different time points.

Standard RNNs struggle with vanishing or exploding gradients, hindering their ability to learn from long sequences. To address this, LSTMs were introduced with three sophisticated gates: input, forget, and output, which help manage information flow and greatly improve the capture of long-term dependencies. In the study by Choi et al. [22], an LSTM-based trajectory prediction model initially processes the historical coordinate data of vehicles through separate LSTM units to extract the dynamic context of each grid. An integrated LSTM network subsequently computes the occupancy probability for each grid using this contextual information and predicts future vehicle trajectories accordingly. Ma et al. [23] propose a LSTM-based real-time traffic prediction algorithm, TrafficPredict. This uses an instance layer to learn instances' movements and interactions and has a category

layer to learn the similarities of instances belonging to the same type to refine the prediction. Contrarily, GRUs employ a simplified architecture with an update gate and a reset gate, which reduces the number of parameters and enhances training efficiency, making it especially suitable for short-term trajectory prediction. Han et al. [24] utilized a GRU-based approach in their research, which inherits the lightweight features of LSTM and effectively extracts information from historical trajectory data. Additionally, Zhu et al. [25] improved the traditional encoder–decoder LSTM structure by introducing trajectory prediction using deep neural networks within a star topology. This approach enhances prediction accuracy and computational efficiency by considering collective interactions and employing linear computations. Although the above models can extract features related to environmental interactions, they typically focus only on a single target and predict its future trajectory alone. The spatial relationships between vehicles include non-Euclidean characteristics such as curved paths, making methods based solely on LSTM both inefficient and non-intuitive in processing spatio-temporal data. The computational demand increases exponentially when predicting the future trajectories of all nearby vehicles, which will lead to inefficiencies in the model's performance. Therefore, the emergence of graph neural networks provides new avenues for addressing these issues.

Graph Neural Networks (GNNs) are inherently well-suited for modeling interactions between vehicles, effectively capturing complex dependencies [26]. GNNs distill structural and semantic information into low-dimensional representations of nodes. Graph Convolutional Networks (GCNs) expand the scope of convolutional operations from conventional CNNs to graph data, facilitating nodes in propagating information and aggregating features through their neighbors [27]. This adaptation enables GCNs to generalize convolutional operations from traditional grid-based data to complex graph structures, enhancing the efficacy of feature extraction and representation learning [28]. Li et al. [29] utilized undirected graphs to represent vehicles and their interactions, employing GCNs alongside spatially related graph operations to effectively extract interaction contexts and discern dynamic inter-vehicle interactions. Additionally, Gao et al. [30] introduced a hierarchical architecture to refine interaction modeling, representing each vehicle through multiple subgraphs. These subgraphs are subsequently integrated using fully connected graphs to elucidate dependencies between vehicles.

### 3. Methodology

In a static traffic scenario at a certain moment, observing any behavior that vehicles are about to execute is based on two levels:

The first level focuses on the current time point, where features from the vehicle's historical trajectory, such as position, velocity, acceleration, heading angle, and relative distance, will affect its future behavior and the development of subsequent trajectories.

The second level involves interactions between various states of surrounding vehicles' historical trajectories and the observed vehicle's state. This includes how the historical spatial positions of nearby vehicles influence the observed vehicle and how their trajectory features interact.

This paper models vehicle trajectory prediction by estimating future trajectory distributions from past trajectories, while also forecasting positions of all adjacent vehicles. Mathematically, this approach is formulated as follows: first, the vehicle positions within the past time range  $T$  are denoted as

$$X = [P_1, P_2, \dots, P_t, \dots, P_T] \quad (1)$$

where

$$P_t = [(x_t^1, y_t^1), \dots, (x_t^n, y_t^n), \dots, (x_t^N, y_t^N)] \quad (2)$$

represents the coordinates at time  $t$ , and  $N$  is the total number of observed vehicles. When past trajectories of vehicles are collected at a certain frequency, the predicted trajectories for the future time range  $F$  can be represented as

$$Y = [P_{T+1}, P_{T+2}, \dots, P_{T+t}, \dots, P_{T+F}] \quad (3)$$

where

$$P_{T+1} = [(x_{T+t}^1, y_{T+t}^1), \dots, (x_{T+t}^n, y_{T+t}^n), \dots, (x_{T+t}^N, y_{T+t}^N)] \quad (4)$$

Therefore, the trajectory prediction problem can be summarized as follows: given the positions of all adjacent vehicles in the past time range  $T$ , the goal is to predict their trajectory distribution within the future time range  $F$ .

#### 4. Proposed Scheme

To overcome the shortcomings of current approaches, this section introduces a novel deep learning model designed to predict target trajectories. Our model consists of three parts: (1) an input preprocessing model, (2) a graph convolutional network (GCN), (3) a trajectory prediction model. This section initially details the computational steps involved in the input preprocessing model, followed by an explanation of how the graph convolutional network extracts spatio-temporal features from vehicle trajectories. Then, based on the features extracted by the GCN network, we construct a trajectory prediction model using a sequence-to-sequence (Seq2Seq) architecture with LSTM. Finally, the training process of the algorithm and the analysis of prediction results are presented.

##### 4.1. Data Preprocessing

To effectively model vehicle motion, it is crucial to accurately capture the spatial and temporal correlations inherent in vehicle dynamics. This requires the use of data input representations that encapsulate the movement of vehicles across both space and time. These representations should include variables such as the coordinates, velocity, and acceleration of vehicle trajectories, which are essential for the model to comprehend the dynamic evolution of vehicle motion over time. Additionally, historical trajectory data is typically formatted as a time series, allowing for the selection of suitable time windows to analyze and predict vehicle trajectories comprehensively.

In this paper, due to the requirements for subsequent data representation and computation, the original data must be transformed. Normalization of the coordinates is performed to eliminate the influence of specific scene scales, making the model more generalizable. Assuming all objects observed in the traffic scene within a time step of  $T_h$  are segmented into  $n$  time slices, we represent this observation information in a specific format as a three-dimensional array ( $n \times T_h \times c$ ). Since trajectories are given in coordinate form in the dataset, we set  $c$  to 2 to represent the  $x$  and  $y$  coordinates of objects. Both coordinate values are normalized to fall within the range  $(-1, 1)$ . The calculation formula can be expressed as

$$x_{norm} = 2 \left( \frac{x - x_{min}}{x_{max} - x_{min}} \right) - 1 \quad (5)$$

where  $x$  is the original coordinate value,  $x_{min}$  and  $x_{max}$  are the minimum and maximum values, respectively, among the coordinate values in the above-mentioned array, and  $x_{norm}$  is the normalized coordinate value.

##### 4.2. Spatio-Temporal Feature Extraction Based on GCNs

In our study, we explore the complex interactions between vehicles to enhance the accuracy of vehicle trajectory predictions. The prediction model considers not only the individual trajectories of vehicles but also the impact of the motion states of surrounding vehicles. This mutual influence is essential for creating a more realistic and reliable prediction model.

In transportation research, vehicle movements often display characteristics of interdependence and constraint. For instance, the lane-changing, deceleration, or acceleration of one vehicle can trigger similar behaviors in nearby vehicles. These intricate interactions are shaped by both the physical dynamics between vehicles and driver behaviors, such as adherence to traffic regulations, subjective decision-making, and the anticipation of the actions of other drivers.

To better capture these interactional relationships, we choose to represent the interactions between vehicles using an undirected graph  $G = \{V, E\}$ . In this representation, nodes  $V$  in the graph represent vehicles on the road, and edges  $E$  represent the interactions between vehicles. Given that GCNs perform convolution operations on each vertex in the graph and have been proven to be effective in capturing spatial dependencies between a node and its neighboring nodes [31], we have incorporated spatial graph convolution operations in our paper. This approach enables the model to perceive the relationships between vehicle nodes, enhancing the accuracy of capturing spatial trajectory evolution.

Due to the significant dependence of vehicle interactions on distance, within a specified observation range, the farther a vehicle is from the vehicle being predicted, the smaller its relative impact. To accurately describe these interactions among vehicles, we assign different weights to each edge in the graph model. Considering that the influence between vehicles is inversely proportional to distance, we use a weighted adjacency matrix in the construction of the graph's spatial convolution module to express this relationship. Specifically, we use the reciprocal of the distance as the weight, ensuring that vehicles closer in proximity receive higher weights in the graph. Our weighted adjacency matrix  $A$  can be specifically represented as follows:

$$A[i][j] = \begin{cases} \frac{1}{D_{i,j}}, & v_i, v_j \in E \\ 0, & \text{else} \end{cases} \quad (6)$$

where  $D_{i,j}$  in  $A[i][j]$  denotes the distance between the vehicles  $i$  and  $j$ .

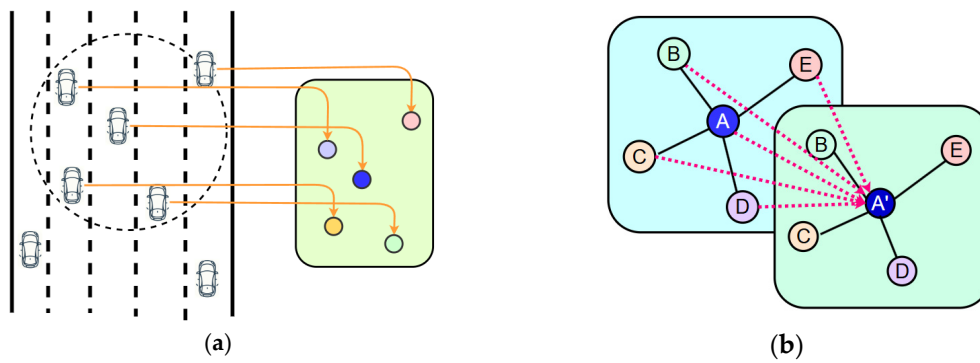
The two-dimensional convolution operation of graph convolution [27] can be expressed as

$$Z_{i+1} = F\left(\Lambda_t^{-\frac{1}{2}} A_t \Lambda_t^{\frac{1}{2}} Z_i W_i\right) \quad (7)$$

where  $Z$  represents the feature matrix of the graph vertices,  $A$  is the adjacency matrix,  $\Lambda$  is the diagonal node degree matrix of  $A$ , and  $W$  is the weight matrix. It aims to normalize the adjacency matrix, which speeds up the training process of the GCN.

As shown in Figure 2, by extracting spatial information around the vehicle and performing graph convolutional operations, the features (including position, speed, and influence weights) of the target vehicle and its surrounding vehicles are weighted and summed, and then the result is passed onto the next layer. The design specifically accounts for how the target vehicle's state impacts its future movement.

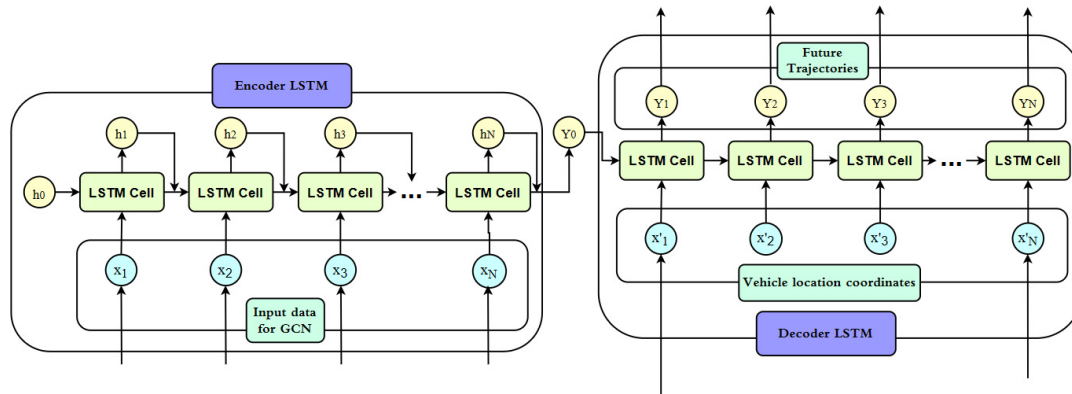
Therefore, by modeling the spatio-temporal relationships of vehicle trajectory data through graph convolutional networks overall, this study can comprehensively and deeply understand the complex interactions between vehicles, providing a more reliable foundation for vehicle trajectory prediction.



**Figure 2.** The graph structure of spatial extraction and convolutional summing. (a) the orange lines represents the relative position mapping of the vehicles.; (b) the red dashed lines represent the weights.

4.3. Trajectory Analysis and Prediction Network

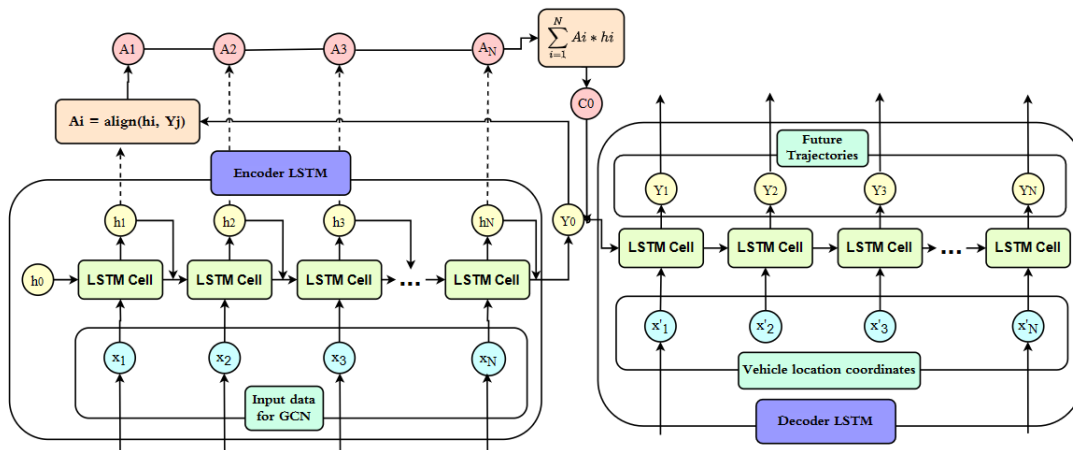
After capturing spatio-temporal features through graph convolution, it is possible to generate future trajectory distributions, which is a typical sequence generation task. In the basic prediction network, the trajectory prediction module is structured as an LSTM-based encoder–decoder network. As shown in Figure 3, it uses the output from the graph convolutional model as input, feeding it into the encoder LSTM at each time step. Subsequently, the encoder’s hidden features, combined with the previous time step’s object coordinates, are inputted into the decoder LSTM to predict the position coordinates for the current time step. This decoding process repeats several times until the model predicts the positions for all expected future time steps.



**Figure 3.** LSTM-based trajectory prediction.

In the traditional Seq2Seq model, the LSTM encoder typically generates a fixed-length context vector to represent the entire past trajectory of the vehicle. This approach may lead to information loss when dealing with long-term trajectory data, because a fixed-length vector struggles to capture all key details. Without using an attention mechanism, the model may perform poorly in capturing long-term dependencies within the sequence, especially in terms of how early behaviors influence future trajectories. This limitation could result in inadequate accuracy when predicting the future positions of vehicles.

With the introduction of the dot-product attention mechanism into the LSTM encoder [32], the traditional Seq2Seq model’s issue of information loss due to the use of fixed-length context vectors can be effectively overcome. This architecture is illustrated in Figure 4.



**Figure 4.** Attention-based LSTM trajectory prediction.

The dot-product attention mechanism dynamically determines the attention weights for each time step through the dot-product calculation between the hidden states produced by the encoder (referred to as “keys”) and the current state of the decoder (referred to as “query”). These weights reflect the contribution of each time step to the current decoding task, optimize the analysis of the relationship between early behaviors and future trajectories, and allow the model to precisely focus on specific input segments according to prediction needs. The computation formula is as follows:

$$A_i = \text{align}(h_i, Y_j), \text{ for } j = 0 \text{ to } M \quad (8)$$

In this process,  $h_i$  represents output states generated by the encoder, while  $Y_j$  denotes the computational outputs of the decoder. The steps for computing using the “align” function evaluates the relevance of each  $h_i$  to every  $Y_j$ .

(1) Linear maps:

$$\begin{cases} k_i = w_k \cdot h_i \\ q_i = w_Q \cdot Y_j \end{cases} \text{ for } j = 0 \text{ to } M \quad (9)$$

(2) Inner product:

$$\tilde{A}_i = k_i^T q_i \quad (10)$$

(3) Normalization:

$$[A_1, \dots, A_M] = \text{softmax}\left(\left[\tilde{A}_1, \dots, \tilde{A}_M\right]\right) \quad (11)$$

After the relevance is quantified as weights  $A_i$ , we generate a context vector  $C$  by taking a weighted average of these weight vectors. The computation formula is as follows:

$$C = \sum_{i=0}^N A_i * h_i \quad (12)$$

$$Y_{j+1} = \tanh\left(W \cdot [X'_i, Y_j, C_i]^{-1} + b\right) \quad (13)$$

Then, the context vector combined with the actual trajectory coordinates  $X'_i$  and the previous decoder output  $Y_j$ , forms a comprehensive vector that is continuously used to update the decoder’s output. This process generates trajectory prediction coordinates for the vehicle at each step.



## 5. Experiments

### 5.1. Dataset

This paper evaluates our approach using the ApolloScape trajectory dataset [33]. The ApolloScape trajectory dataset is collected by running Apollo autonomous vehicles in urban areas during peak hours. The collected traffic data includes camera-based images and LiDAR-based point clouds, and object detection and tracking algorithms are used to compute object trajectories. Overall, the trajectory dataset consists of 53 min of training sequences captured at a rate of 2 frames per second and 50 min of testing sequences. The dataset provides information such as object ID, object type, object location, object size, and heading angle. As the data is collected in urban areas, it involves five different types of objects: small vehicles, large vehicles, pedestrians, motorcycle riders, and bicycle riders. This specific dataset allows researchers to stress-test their designed trajectory prediction solutions to accommodate various types of traffic agents with different behaviors, posing additional challenges in design.

During training, 20% of the sequences from our training subset were used for validation, while we trained the model with the other 80%. After training, we generated predictions for the test sequences and assessed the outcomes.

### 5.2. Experimental Environment

Table 1 outlines the specific hardware configuration used for the experiments detailed in this paper, which were conducted using Python and the PyTorch deep learning framework.

**Table 1.** Hardware testing platform.

Experimental Equipment	Experimental Specifications
CPU	12th Gen Intel(R) Core(TM) i7-12700KF
CPU Frequency	3600 MHz
GPU	NVIDIA GeForce RTX 3090
System	Ubuntu 20.04
Memory	64 GB

### 5.3. Results Evaluation

RMSE: In this paper, we report our results based on the root mean square error (RMSE) of the predicted trajectories for the future (within a 5 s range). The RMSE at time  $t$  can be calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_{\text{Pred}}^t[i] - Y_{\text{true}}^t[i])^2} \quad (14)$$

Here,  $n$  is the number of samples, and  $Y_{\text{pred}}^t$  and  $Y_{\text{true}}^t$  are the corresponding predicted outcomes and the ground truth at time  $t$ .

### 5.4. Ablation Experiments

In this section, we conducted two ablation studies to validate the efficacy of different components within our MESTGN:

(1) Removal of different modules: This includes excluding the GCN from the MESTGN, disabling the attention mechanism within the encoder–decoder, or reverting it to a conventional sequence to sequence (Seq2Seq) model for predictions. The outcomes of these ablation studies, as presented in Table 2, unequivocally demonstrate that the removal of any component from the MESTGN results in an increase in the RMSE.

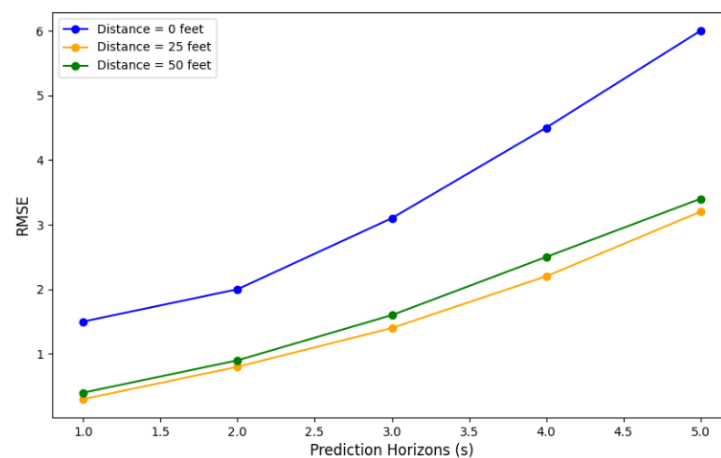
**Table 2.** Comparison off the RMSE for the different components within the MESTGN.

Prediction Horizons	MESTGN (No GCN)	MESTGN (No Attention)	MESTGN
1	0.57	0.37	<b>0.35</b>
2	1.73	1.12	<b>0.81</b>
3	2.81	1.67	<b>1.43</b>
4	4.12	3.12	<b>2.20</b>
5	5.34	4.36	<b>3.24</b>
Average	2.51	2.13	<b>1.40</b>

The bold values indicate the lowest RMSE results within each prediction horizons.

Table 2 can be understood as follows: the absence of the GCN component leads to a deficiency in the spatio-temporal dependencies among the features inputted into the LSTM. This deficiency stems from the lack of GCN's support, which is crucial for adequately reflecting the dynamic interactions among vehicles across different times and locales, thereby restricting the model's capability to adapt to intricate traffic scenarios. Conversely, when the dot-product attention mechanism is excluded from the encoder–decoder configuration, the model's ability to capitalize on the longitudinal characteristics of vehicle trajectory data is significantly compromised. The attention mechanism is instrumental in enabling the model to prioritize historical events that exert substantial influence on the predictive outcomes, such as specific driving behaviors or preferred routes.

(2) Due to our model's focus on the spatio-temporal positions of vehicles using graph convolution, surrounding objects around vehicles become essential considerations. Therefore, we investigated how the distance between vehicles affects the performance of our model. In this study, we use a distance threshold, referred to as "Distance" to determine whether there is an edge connection between two vehicle nodes, and we explore the optimal distance between two vehicles in the graph network for the best predictive performance of the model. In Figure 5, we can see that when "Distance" = 0 (indicating exclusion of surrounding objects, depicted by blue lines in Figure 5), the prediction error is significantly higher than when "Distance" > 0 (considering nearby objects). Thus, considering surrounding objects indeed contributes to improving the predictive capability of our model. Additionally, we observed a slight increase in prediction error as "Distance" increased from 25 feet (depicted by orange lines) to 50 feet (represented by green lines). This is due to vehicles in road traffic being more affected by nearby objects while often ignoring the influence of distant ones.

**Figure 5.** Prediction error at different distance values.

### 5.5. Quantitative Analysis

In Table 3, we compared the proposed MESTGN model with other models for validation, including TrafficPredict [23], StarNet [25], and GRIP [29]. These methods are all based on LSTM for trajectory prediction. The algorithm's performance metrics adopt error weight configurations provided by the ApolloScape website. They use the following metrics to assess the algorithm's performance:

- Average displacement error (ADE): the mean Euclidean distance over all the predicted positions and ground truth positions during the prediction time.
- Final displacement error (FDE): the mean Euclidean distance between the final predicted positions and the corresponding ground truth locations.

**Table 3.** Model prediction error.

Method	Model	WSADE	ADE (Vehicles)	ADE (Pedestrian)	ADE (Cyclists)	WSFDE	FDE (Vehicles)	FDE (Pedestrians)	FDE (Cyclists)
TrafficPredict [23]		8.588	7.946	7.810	12.880	24.226	12.77	11.121	22.791
StarNet [25]		1.342	2.386	0.785	1.862	2.498	4.285	1.515	3.464
Grip [29]		1.259	2.240	0.714	<b>1.802</b>	2.363	4.076	1.373	3.415
MESTGN (ours)		<b>1.238</b>	<b>2.217</b>	<b>0.681</b>	1.819	<b>2.277</b>	<b>3.894</b>	<b>1.297</b>	<b>3.391</b>

The bold values indicate the smallest prediction error for each method; lower values are better.

As trajectories of vehicles, cyclists, and pedestrians have different scales, they use the following weighted sum of the ADE (WSADE) and the weighted sum of the FDE (WSFDE) as metrics.

$$\text{WSADE} = D_v \cdot \text{ADE}(\text{vehicles}) + D_p \cdot \text{ADE}(\text{pedestrian}) + D_c \cdot \text{ADE}(\text{cyclist}) \quad (15)$$

$$\text{WSFDE} = D_v \cdot \text{FDE}(\text{vehicles}) + D_p \cdot \text{FDE}(\text{pedestrian}) + D_c \cdot \text{FDE}(\text{cyclist}) \quad (16)$$

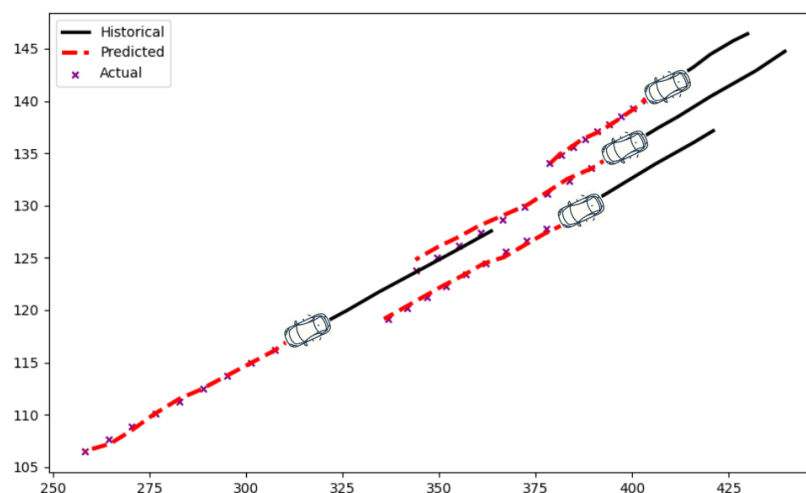
The weights for vehicles, pedestrians, and cyclists are respectively set to 0.20, 0.58, and 0.22. These weights are inversely related to the average speeds of vehicles, pedestrians, and cyclists in the dataset.

In Table 3, GRIP has a lower ADE for cyclists, but MESTGN has overall lower WSADE and WSFDE metrics compared with GRIP. MESTGN uses an attention-enhanced LSTM network to analyze historical data more precisely, resulting in superior predictive performance. Additionally, MESTGN outperforms both TrafficPredict and StarNet across all predictive metrics. This highlights the superiority of GCN compared to LSTM in capturing spatial dependencies, underlining the critical role of incorporating surrounding vehicle information in trajectory prediction.

### 5.6. Vehicle Trajectory Visualization Analysis

To investigate the accuracy of the MESTGN model in vehicle trajectory prediction, we conducted a series of predictions using the ApolloScape trajectory dataset under consistent road scenarios, visualizing the results of the MESTGN model in Figure 6. The original trajectories were based on 3 s of historical data, with the model predicting trajectories over a 6 s span.

In Figure 6, it is observed that vehicles closely follow their predecessors within the three straight lanes, indicating the MESTGN model's effective recognition of vehicle-following behavior. Further examination reveals that the predicted coordinates closely match the actual coordinates, demonstrating the MESTGN model's ability to accurately capture and understand vehicle trajectories. This enhances accuracy in vehicle trajectory prediction tasks.



**Figure 6.** Vehicle trajectory visualization.

## 6. Conclusions

In this paper, we propose an innovative model framework based on Graph Convolutional Networks (GCNs) and a combination of Long Short-Term Memory (LSTM) networks with attention mechanisms. This framework predicts the trajectories of all vehicles in a scene and calculates their future trajectory distributions. In our approach, we introduce a weighted adjacency matrix to differentiate the varying impacts of nearby vehicles on the target vehicle. Building upon this, we utilize the graph convolutional network module to learn the temporal and spatial dependencies among vehicles. Additionally, within the LSTM-based Seq2Seq encoder–decoder structure, we incorporate an attention mechanism to focus more on the historical trajectory information of vehicles. Experimental results demonstrate that our model framework outperforms mainstream methods, exhibiting smaller prediction errors and better prediction outcomes, thus indicating its potential for deployment in autonomous vehicles.

## 7. Discussion

For future work, we can further explore and research more complex attention mechanisms to enhance the model’s ability to model the intricate relationships between vehicles. Additionally, considering the integration of more data sources such as traffic signal states, road conditions, weather information, etc., can improve the model’s robustness and generalization capability. Furthermore, applying the proposed model framework to real intelligent transportation systems and conducting field testing and evaluation to validate its effectiveness and reliability in real-world environments will be crucial. Considering deploying this model for actual vehicle trajectory prediction in different environments is also important. Specific deployment environments may include, but are not limited to, vehicles themselves, cloud platforms, or roadside units. Finally, our future work will focus on further optimizing the model’s structure and parameter settings to enhance prediction accuracy and efficiency while reducing computational costs and resource usage. By pursuing these goals, we aim to refine and advance the proposed model framework, thereby expanding the potential and applications of intelligent transportation systems.

**Author Contributions:** Conceptualization, Xin Wang and Wenzheng Li; methodology, Zhiming Gui; writing—original draft preparation, Xin Wang; writing—review and editing, Zhiming Gui and Wenzheng Li; supervision, Zhiming Gui and Wenzheng Li. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The ApolloScope trajectory dataset used in this study can be accessed at <https://apolloscope.auto/trajectory.html>, accessed on 14 June 2023.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Liu, J.; Luo, Y.; Xiong, H.; Wang, T.; Huang, H.; Zhong, Z. An integrated approach to probabilistic vehicle trajectory prediction via driver characteristic and intention estimation. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3526–3532.
- Lin, W.; Zhou, Y.; Xu, H.; Yan, J.; Xu, M.; Wu, J.; Liu, Z. A tube-and-droplet-based approach for representing and analyzing motion trajectories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1489–1503. [[CrossRef](#)] [[PubMed](#)]
- Li, X.; Xia, J.; Chen, X.; Tan, Y.; Chen, J. SIT: A Spatial Interaction-Aware Transformer-Based Model for Freeway Trajectory Prediction. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 79. [[CrossRef](#)]
- Rasouli, A.; Tsotsos, J.K. Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 900–918. [[CrossRef](#)]
- Ma, Y.; Wang, Z.; Yang, H.; Yang, L. Artificial intelligence applications in the development of autonomous vehicles: A survey. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 315–329. [[CrossRef](#)]
- Abbas, M.T.; Jibrán, M.A.; Afaq, M.; Song, W. An adaptive approach to vehicle trajectory prediction using multimodel Kalman filter. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3734. [[CrossRef](#)]
- Jianping, L. Research on Traffic Vehicle Motion Prediction Method for Intelligent Driving. Master's Thesis, Jilin University, Changchun, China, 2018.
- Zhang, J.; Liao, Y.; Wang, S.; Han, J. Study on driving decision-making mechanism of autonomous vehicle based on an optimized support vector machine regression. *Appl. Sci.* **2018**, *8*, 13. [[CrossRef](#)]
- Li, J.; Dai, B.; Li, X.; Xu, X.; Liu, D. A dynamic bayesian network for vehicle maneuver prediction in highway driving scenarios: Framework and verification. *Electronics* **2019**, *8*, 40. [[CrossRef](#)]
- Jiang, H.; Chang, L.; Li, Q.; Chen, D. Trajectory prediction of vehicles based on deep learning. In Proceedings of the 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 5–7 September 2019; pp. 190–195.
- Zhao, T.; Xu, Y.; Monfort, M.; Choi, W.; Baker, C.; Zhao, Y.; Wang, Y.; Wu, Y.N. Multi-agent tensor fusion for contextual trajectory prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12126–12134.
- Shuai, L.; Jing, L. Random Local Search Heuristic Method Based on Deep Reinforcement Learning. *J. Jilin Univ.* **2021**, *51*, 1420–1426.
- Marcot, B.G.; Penman, T.D. Advances in Bayesian network modelling: Integration of modelling technologies. *Environ. Model. Softw.* **2019**, *111*, 386–393. [[CrossRef](#)]
- Eddy, S.R. Hidden markov models. *Curr. Opin. Struct. Biol.* **1996**, *6*, 361–365. [[CrossRef](#)] [[PubMed](#)]
- Belgiu, M.; Drăguț, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [[CrossRef](#)]
- Xing, Z.; Pei, J.; Keogh, E. A brief survey on sequence classification. *ACM Sigkdd Explor. Newsl.* **2010**, *12*, 40–48. [[CrossRef](#)]
- Rajan, K.; Harvey, C.D.; Tank, D.W. Recurrent network models of sequence generation and memory. *Neuron* **2016**, *90*, 128–142. [[CrossRef](#)] [[PubMed](#)]
- Dollár, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005; pp. 65–72.
- Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Non-local Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium, Paris, France, 9–12 June 2019.
- Deo, N.; Trivedi, M.M. Convolutional social pooling for vehicle trajectory prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1468–1476.
- Cui, H.; Radosavljevic, V.; Chou, F.-C.; Lin, T.-H.; Nguyen, T.; Huang, T.-K.; Schneider, J.; Djuric, N. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In Proceedings of the 2019 International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019; pp. 2090–2096.
- Choi, S.; Kim, J.; Yeo, H. Attention-based recurrent neural network for urban vehicle trajectory prediction. *Procedia Comput. Sci.* **2019**, *151*, 327–334. [[CrossRef](#)]
- Ma, Y.; Zhu, X.; Zhang, S.; Yang, R.; Wang, W.; Manocha, D. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6120–6127.
- Han, P.; Wang, W.; Shi, Q.; Yang, J. Real-time short-term trajectory prediction based on GRU neural network. In Proceedings of the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8 September–12 September 2019.
- Zhu, Y.; Qian, D.; Ren, D.; Xia, H. Starnet: Pedestrian trajectory prediction using deep neural network in star topology. *arXiv* **2019**, arXiv:1906.01797.
- Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [[CrossRef](#)] [[PubMed](#)]
- Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.

28. Henaff, M.; Bruna, J.; LeCun, Y. Deep convolutional networks on graph-structured data. *arXiv* **2015**, arXiv:1506.05163.
29. Li, X.; Yang, X.; Chuah, M. Grip: Graph-based interaction aware trajectory prediction. In Proceedings of the IEEE ITSC, Auckland, New Zealand, 27–30 October 2019.
30. Gao, J.; Sun, C.; Zhao, H.; Shen, Y.; Anguelov, D.; Li, C.; Schmid, C. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
31. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
32. Yi, S.; Wang, X.; Yamasaki, T. Impression prediction of oral presentation using lstm and dot-product attention mechanism. In Proceedings of the 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, 11–13 September 2019; pp. 242–246.
33. Huang, X.; Cheng, X.; Geng, Q.; Cao, B.; Zhou, D.; Wang, P.; Lin, Y.; Yang, R. The apollo-scape dataset for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 954–960.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.