



A deep learning approach to predicting vehicle trajectories in complex road networks

K. Sundari¹ · A. Senthil Thilak¹

Received: 31 March 2024 / Accepted: 23 May 2024
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2024

Abstract

Accurate prediction of vehicle trajectories is essential for safe and efficient navigation in urban environments, particularly with the increasing prevalence of autonomous vehicles and intelligent transportation systems. This paper introduces a deep learning-based approach for predicting vehicle trajectories on urban roads in real time. The method combines techniques from graph neural networks (GNNs) and long short-term memory (LSTM)-based models to capture intricate spatial and temporal dependencies among vehicles. Vehicles are represented as nodes in the proposed graph model, and graph attention mechanism is used to model the interactions between them. Additionally, LSTM modules encode motion patterns and temporal correlations, facilitating spatial and temporal information fusion to improve prediction accuracy. The effectiveness of the approach is demonstrated through extensive experimentation and evaluation in generating vehicle trajectories, surpassing baseline methods. The proposed method holds promise for real-time vehicle trajectory prediction, with the potential for applications in autonomous driving, traffic management, and intelligent transportation systems.

Keywords Autonomous vehicles · Trajectory prediction · Deep learning · Graph attention network · Intelligent transportation systems · Graph neural networks

1 Introduction

In today's rapidly evolving urban landscapes, the accurate prediction of vehicle trajectories plays a critical role in numerous applications ranging from traffic management to autonomous driving systems. The ability to anticipate the future movements of vehicles enables proactive decision-making, thereby enhancing safety, efficiency, and overall transportation effectiveness. Traditional approaches to vehicle trajectory prediction have primarily relied on mathematical models or simplistic statistical methods, often needing help to capture the intricate dynamics of real-world traffic scenarios [1–3].

However, recent advancements in deep learning have revolutionized vehicle trajectory prediction by offering more sophisticated and data-driven techniques capable of handling the complexities inherent in modern traffic environments [4]. Deep learning models have been widely applied in various domains of transportation research in recent years, providing promising solutions to complex problems. Some notable applications include traffic flow prediction [5], origin–destination trip matrix estimation [6], traffic flow estimation and accident analysis [7–9], and vehicle trajectory prediction. Hence traditional approaches often rely on simplistic assumptions, leading to suboptimal performance in dynamic and complex traffic environments. Moreover, existing deep learning-based methods are capable of capturing nonlinear dynamics of such complex traffic environments in real world.

One of the key challenges in vehicle trajectory prediction is to effectively capture both the temporal evolution of trajectories and the spatial interactions among vehicles. The temporal dependencies, such as acceleration, deceleration, and lane changes, play a significant role in shaping future vehicle movements. Additionally, spatial interactions, like lane merging, proximity to other vehicles, and traffic

K. Sundari and A. Senthil Thilak have contributed equally to this work.

✉ K. Sundari
sundarikrishnaperumal@gmail.com
A. Senthil Thilak
thilak@nitk.edu.in

¹ Department of Mathematical and Computational Sciences,
National Institute of Technology Karnataka, Surathkal,
Mangaluru, Karnataka 575025, India

flow patterns, offer crucial contextual information essential for accurate predictions. Among various techniques used for predicting vehicle trajectories, recurrent neural networks (RNNs) have demonstrated superior performance in recent studies [10–14]. However, RNNs, particularly long short-term memory (LSTM) networks commonly employed for trajectory prediction, primarily capture temporal dependencies in the data while overlooking the spatial relationships between traffic objects. This limitation restricts the usability of LSTMs for accurate trajectory predictions, especially in complex and dynamic environments where considering the relationships between objects is indispensable.

In contrast, graph neural networks (GNNs) rely on graph structures to model the underlying traffic scenarios. GNNs effectively capture spatial information by leveraging the structural connectivity among neighboring nodes within graph models [15]. Recent advancements in GNN-based vehicle trajectory prediction models utilize graph convolutional networks (GCN) to learn graph features before inputting trajectory data into Recurrent Neural Networks (RNN) for sequence prediction [16–18]. These models demonstrate lower prediction errors compared to other RNN-based approaches [19]. However, existing GCN-based vehicle trajectory prediction models typically assign equal importance to all neighbors of a given node during feature extraction. This contradicts real-time network behavior, where the influence of neighboring objects varies significantly depending on traffic settings. Notably, not all neighboring objects exert the same influence on a road object under consideration. To address this issue, Velickovic et al. proposed the graph attention network (GAT) [20]. GAT enables the assignment of varying importance to different nodes within a neighborhood without requiring costly matrix operations. Vehicle trajectory prediction models based on GAT have surpassed many state-of-the-art GCN-based models, showcasing superior performance.

However, although the diverse aspects have been well investigated, a factor was neglected in previous works. In the context of vehicle trajectory prediction, considering both spatial interactions at the same time-step and the temporal continuity of interactions is crucial for accurate and safe predictions. While existing approaches have often focused on the spatial interactions between vehicles at a given moment, neglecting the temporal aspect can lead to incomplete predictions. Temporal continuity refers to the idea that the current motion behavior of a vehicle is influenced not only by its immediate surroundings but also by the historical movements of other vehicles in the vicinity. Each vehicles in the given scene must account for the past trajectories of surrounding vehicles to plan their future paths effectively. By incorporating temporal correlations of interactions into vehicle trajectory prediction models, we enable vehicles to make more informed decisions based on a deeper understanding of

the dynamic interactions within the traffic environment. This approach enhances the predictive capabilities of autonomous vehicles and improves overall traffic safety and efficiency.

In this paper, we focus on the task of predicting vehicle trajectories in urban environments, which are characterized by complex road networks. While the terminology “complex networks” refers to a specific class of networks in the context of graph theory, in this paper, we use the term “complex road networks” to refer to the intricate and challenging nature of urban road structures, including factors such as dense traffic, diverse road types, and complex intersections. It is important to clarify this distinction, as our study primarily addresses the practical challenges posed by urban road networks rather than the abstract network properties. Here, we introduce a novel deep learning approach to predict vehicle trajectories in a complex urban road network and the key contributions are:

Integration of RNNs and GNNs: We propose a hybrid approach that integrates recurrent neural networks (RNNs) and graph neural networks (GNNs) for capturing both spatial and temporal dependencies in vehicle trajectories. This integration allows our model to effectively capture the complex interactions between vehicles and predict their future movements accurately.

Temporal correlations modeling with additional LSTM: Initially, we address the temporal correlations of interactions by integrating an additional LSTM network. Notably, prior approaches often overlooked the significance of explicitly considering the continuity of interactions.

Modeling spatial interactions: Representing each vehicle as a node in the underlying graph model and thereafter utilizing graph attention mechanisms, the proposed method captures the spatial relationships and interactions between vehicles in urban traffic scenarios. This enables our model to learn from the collective behavior of neighboring vehicles and incorporate their influence into trajectory predictions. Here, we tackle the spatial interactions among vehicles by employing Graph Attention Network (GAT) to aggregate the hidden states of LSTMs.

Evaluation and comparison: We conduct extensive experiments on real-world trajectory datasets and compare the performance of our method against baseline approaches. The results demonstrate the effectiveness of our strategy in generating multiple plausible trajectories and outperform the existing methods in terms of prediction accuracy and reliability.

2 Literature review

The vehicle trajectory prediction task involves forecasting the future positions of individual vehicles within the context of current traffic conditions. With the growing interest in

autonomous driving, there has been a surge in research attention towards improving vehicle trajectory prediction. Recent advancements in deep learning models have shown promising results in tackling this challenge. In this section, we provide a concise overview of the existing literature related to vehicle trajectory prediction.

Recurrent neural networks for vehicle trajectory prediction: Sequence prediction tasks revolve around utilizing past sequence data to forecast future values within the sequences. Recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) networks, are specifically tailored for sequence prediction challenges. They have demonstrated significant accomplishments across various domains, including speech and behavior recognition [21, 22], machine translation, resource allocation [23], as well as image or video captioning.

While many studies have highlighted the effectiveness of LSTM in modeling the trajectory of individual vehicles, such as those by Lin et al. [24] and Kavran et al. [25], Vanilla LSTM models often overlook the interactions among surrounding vehicles. This limitation arises because RNN models excel primarily in learning temporal sequences. To address this challenge, Deo and Trivedi introduced convolutional social pooling within an LSTM encoder–decoder framework to capture the interdependencies among moving vehicles. However, it is crucial to recognize that both spatial and temporal information of vehicles significantly influence trajectory prediction accuracy. Consequently, the applicability of LSTMs for precise trajectory predictions becomes constrained in complex, dynamic environments where accounting for vehicle relationships is essential.

Recently, the emergence of graph neural networks (GNNs) and their variations has addressed this limitation by integrating a GNN layer into the LSTM network architecture, thereby enhancing sequence prediction capabilities. GNNs leverage the structural connectivity of neighboring nodes inherent in graph models to capture spatial information. By extracting features from neighboring nodes, GNNs facilitate the learning of valuable relational information crucial for decision-making in interaction-based environments.

Graph neural networks for vehicle trajectory prediction: Graph neural networks represent a powerful neural network architecture tailored for machine learning tasks involving graphs. In recent years, systems built upon Graph Convolutional Networks (GCNs) and Gated Graph Convolutional Neural Networks (GGNNs) have showcased remarkable performance across various domains. These include tasks such as modeling physics systems, learning molecular fingerprints, and predicting protein interfaces. In the existing literature, numerous research contributions have leveraged the principles of graph neural networks (GNNs) for sequence prediction tasks [17, 26–28]. Among these, several notable studies are particularly relevant to the proposed investigation.

One such contribution is the trajectory prediction method developed by Chandra et al. in 2019 [29]. This method is tailored for dense and heterogeneous traffic scenarios, employing a weighted interaction mechanism to capture the influence of neighboring vehicles on the trajectory of the target vehicle. By assigning weights to interactions based on their relevance and importance, Traphic demonstrates improved trajectory prediction accuracy in complex traffic conditions. However, its performance in sparse environments remains unclear, highlighting the need for further evaluation across a broader range of traffic scenarios and real-world datasets to assess its generalizability.

Additionally, Li et al. proposed an interaction-aware graph convolutional vehicle trajectory prediction model in 2020 [17], aiming to accurately forecast longer sequences with lower error rates across diverse traffic settings. This model notably reduces error rates in longer prediction intervals by utilizing graph convolution operations to transform features into a graph, where uniform weights are assigned to all neighboring traffic participants. Nevertheless, in real-world scenarios, the influence of neighboring vehicles on a given participant may vary, constituting a critical factor in achieving accurate trajectory prediction.

In their work [30], Sunwoo and Lee introduced a multimodal maneuver-based trajectory prediction model that integrates LSTM and hierarchical GNN architectures. This model employs a two-stage prediction process. Initially, all potential trajectories of surrounding vehicles are forecasted based on multimodal maneuvers, alongside the probability for each maneuver determined through a multilayer perceptron (MLP). As LSTM alone struggles to capture spatial interactions among vehicles, the authors combined LSTM encoder–decoder with GCN to extract relevant node and edge features from the associated graph model. Subsequently, an interaction-aware graph model is proposed in the second stage, utilizing these maneuver-based predicted trajectories. Despite the advancements, accurately predicting longer sequences of trajectories remains a primary challenge in these models. To address the issue of increased error rates in long-term prediction, Cao et al. proposed a temporal attention-based vehicle trajectory prediction model [31]. By leveraging attention mechanisms in long-term prediction, this model demonstrated improved accuracy compared to existing baseline models.

In their study [32], Mo et al. presented a novel approach for vehicle trajectory prediction in highway driving scenarios, merging GNNs and RNNs. By leveraging the graph structure of the road network to model spatial relationships between vehicles and incorporating temporal dependencies using RNNs, their model achieved promising results in trajectory prediction accuracy. However, the model's focus primarily on highway driving scenarios may limit its applicability to other road environments, despite its strengths.

Additionally, real-time traffic networks introduce uncertainties due to inevitable circumstances.

Among these methodologies, Velickovic et al. [20] introduced the Graph Attention Network (GAT), offering the ability to assign varying levels of importance to different nodes within a neighborhood without requiring computationally expensive matrix operations. GAT-based models have consistently achieved or matched state-of-the-art performance across various benchmarks for graph-related tasks [33–37]. In our context, the intricate latent motions of vehicles can be effectively modeled using GAT. Here, vehicles within the road scene are conceptualized as nodes on the graph at each time-step, with the interactions between vehicles represented as graph edges.

Contributions: While previous works have largely focused on either modeling each vehicle’s motion using RNNs or integrating RNNs with GCN, CNN, or GAT, an essential factor has often been overlooked. In the context of vehicle trajectory prediction, considering both spatial interactions and temporal continuity is crucial for achieving accurate and safe predictions. To the best of our knowledge, most of the existing approaches have primarily addressed spatial interactions at a given moment. However, neglecting the temporal aspect can result in incomplete and nonrealistic predictions. Temporal continuity indicates that a vehicle’s current motion behavior is not only influenced by its immediate surroundings but also by the historical movements of neighboring vehicles. Each vehicle in a given scenario must consider the past trajectories of surrounding vehicles to effectively plan its future path. By incorporating temporal correlations of interactions into vehicle trajectory prediction models, we can facilitate vehicles to take more informed decisions based on a deeper understanding of the dynamic interactions within the traffic environment.

In line with this objective, our proposed study aims to develop a deep learning approach that combines GNN-based trajectory prediction models, leveraging both GAT and LSTM architectures. Additionally, an extra LSTM layer is utilized to capture temporal correlations effectively. As outlined in Sect. 1, the primary goal of our study is to develop a model that is adaptive to complex road scenarios commonly encountered in urban environments.

3 Methodology

In this section, we introduce a novel deep learning approach, utilizing a sequence-to-sequence architecture to predict future vehicle trajectories. In addition to capturing spatial interactions through a graph attention mechanism at each time-step, we incorporate an LSTM to encode the temporal correlations of interactions. In the following section, we outline the problem statement and provide comprehensive

insights into our model’s architecture, functionality, implementation, and training procedures.

3.1 Problem definition

In our study, we focus on dynamic scenarios involving multiple vehicles moving along urban roadways, each identified as v_1, v_2, \dots, v_M . At any given time-step t , we represent the position of vehicle v_i (where i ranges from 1 to M) as $P_{v_i}^t = (x_{v_i}^t, y_{v_i}^t)$, indicating coordinates along the horizontal and vertical axes, respectively. Our objective is to utilize historical trajectory data to forecast the future positions of these vehicles. The historical trajectory data, denoted as $P_{v_i}^t$, encapsulates the past movement of each vehicle from time-step $t = 1$ to T_P . Here, T_P represents the last observed time-step in the historical data. Our primary goal is to predict the future positions of the vehicles beyond the observed time-steps, specifically targeting time-steps ranging from $t = T_P + 1$ to T_F . Here, T_F signifies the final time-step for prediction.

3.2 The architecture and working of the proposed model

The basic architecture of our model is a encoder–decoder sequence to sequence model as depicted in Fig. 1. The model consists of three primary components within the encoder architecture as follows: (1) *LSTM-based vehicle trajectory encoding module*: This module is responsible for encoding the historical trajectory data of each vehicle. (2) *GAT-based module for modeling spatial interactions*: The GAT-based module is designed to capture spatial interactions among vehicles. By treating vehicles as nodes in a graph and lever-

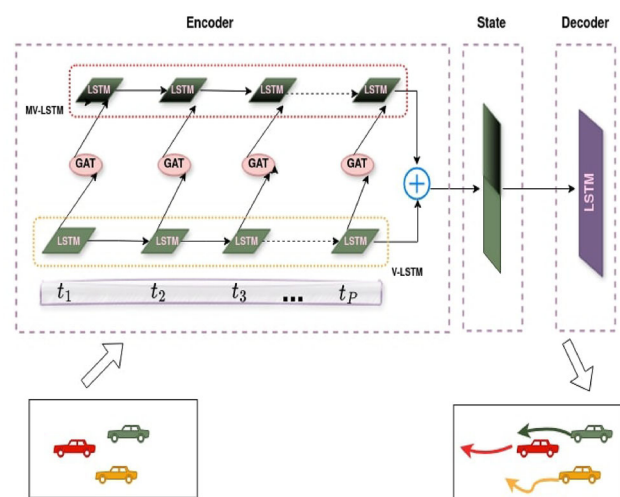


Fig. 1 Architecture of the proposed vehicle trajectory prediction model: A sequence to sequence-based framework comprising Encoder, Intermediate State, and Decoder modules for predicting vehicle trajectories

aging attention mechanisms, the model can learn to prioritize and weigh the influence of neighboring vehicles on each other's trajectories. This allows for a more nuanced understanding of the complex spatial relationships within the environment. (3) *LSTM-based module for capturing temporal correlations*: This module focuses on capturing the temporal correlations of the interactions modeled by the GAT-based module. By incorporating LSTM units, the model can effectively track how spatial interactions evolve over time, enabling it to predict future trajectories based on both historical movement patterns and dynamic spatial relationships. Next, after obtaining the encoder vector from the encoder, the decoder LSTM begins decoding the data to predict future trajectories.

3.2.1 LSTM-based vehicle trajectory encoding module

The proposed model is designed to process unprocessed video data or drone images captured under realistic city traffic conditions, as detailed in Sect. 4. Prior to being inputted into the model, the raw data undergoes initial pre-processing procedures, as outlined in Sect. 4.3, to optimize its compatibility and quality for effective analysis. After generating the processed data, the proposed model uses LSTM encoder module to encode each vehicle data. This section describes how the proposed model utilizes an LSTM encoder module, termed V-LSTM, to encode the motion state of each vehicle based on the generated processed data and the corresponding traffic graph.

Each vehicle exhibits a unique motion pattern influenced by factors such as speed, acceleration, and steering behavior. LSTM models have shown effectiveness in capturing the historical motion state of single entities, including vehicles [38–40]. Therefore, we employ one LSTM for each vehicle to encode its motion state. We denote this LSTM as V-LSTM (LSTM for vehicle motion encoding). In our implementation, we first calculate the relative position of the vehicle at each time-step (t) compared to the previous time-step ($t - 1$) as in Eq. (1).

$$\Delta x_{v_i}^t = x_{v_i}^t - x_{v_i}^{t-1}, \quad \Delta y_{v_i}^t = y_{v_i}^t - y_{v_i}^{t-1} \quad (1)$$

This equation helps us to calculate how much a vehicle has moved horizontally ($\Delta x_{v_i}^t$) and vertically ($\Delta y_{v_i}^t$) between time t and $t - 1$. Subsequently, we embed the relative position of each vehicle v_i into a fixed-length vector $p_{v_i}^t$ for each time-step, and these vectors serve as inputs to the V-LSTM cell. We denote the embedding function as $\mathcal{E}(\cdot)$ as in Eq. (2).

$$p_{v_i}^t = \mathcal{E}(\Delta x_{v_i}^t, \Delta y_{v_i}^t; W_{pp}) \quad (2)$$

In Eq. (2), $\mathcal{E}(\cdot)$ represents the embedding function, and W_{pp} denotes the embedding weight. The V-LSTM cell takes

these embedded vectors along with the previous hidden state $z_{v_i}^{t-1}$ as input and produces the current hidden state $z_{v_i}^t$. This process is governed by the weight W_z associated with the V-LSTM cell by Eq. (3).

$$z_{v_i}^t = \text{V-LSTM}(z_{v_i}^{t-1}, p_{v_i}^t; W_z) \quad (3)$$

Here, $z_{v_i}^t$ represents the hidden state of the V-LSTM at time-step t . W_z is the weight of the V-LSTM cell. It is crucial to note that these parameters, including W_{pp} and W_z , are shared among all vehicles in the traffic scene, ensuring consistency in the encoding process across different entities. This encoding process allows the model to effectively capture the historical motion state of each vehicle, laying the foundation for further analysis and prediction tasks within the proposed framework.

3.2.2 GAT-based module for modeling spatial interactions

Simply using one LSTM per vehicle may not effectively capture the intricate interactions between vehicles on the road. To overcome this limitation and better capture the dynamics of traffic scenarios, we treat vehicles as nodes in a graph, leveraging the advancements in GNNs.

Graph attention networks (GATs) operate on graph-structured data and compute the features of each graph node by attending over its neighbors, following a self-attention strategy. GATs are constructed by stacking graph attention layers. Figure 3 illustrates the concept of a single graph attention layer, showcasing the process of attention computation and feature aggregation among nodes in a graph structure. Each node in the graph corresponds to a vehicle, and the edges between nodes represent the interactions between vehicles. This graph-based representation enables us to capture the spatial dependencies and interactions among vehicles in a traffic scenario effectively. The input of the graph attention layer is denoted as $\vec{h} = \{\vec{h}_{v_1}, \vec{h}_{v_2}, \dots, \vec{h}_{v_N}\}$, where $\vec{h}_{v_i} \in \mathbb{R}^F$ and v_N is the number of nodes, and F is the feature dimension of each node. The output is $\vec{h}' = \{\vec{h}'_{v_1}, \vec{h}'_{v_2}, \dots, \vec{h}'_{v_N}\}$, where $\vec{h}'_{v_i} \in \mathbb{R}^F$.

In our approach, vehicles are represented as nodes in a graph as illustrated in Fig. 2, and edges are formed between any two road agents that fulfill the interaction rule. The interaction rule dictates that an edge exists between two vehicles if they are within a proximity threshold distance, for example, 10m on an urban road. This graph-based representation enables us to model the complex interactions and dependencies between vehicles in a traffic scenario. During the observation period, $z_{v_i}^t$ ($t = 1, \dots, T_p$), representing the hidden states of the vehicles, is fed to the graph attention layer. The coefficients in the attention mechanism of the node pair (v_i, v_j) can be computed by:

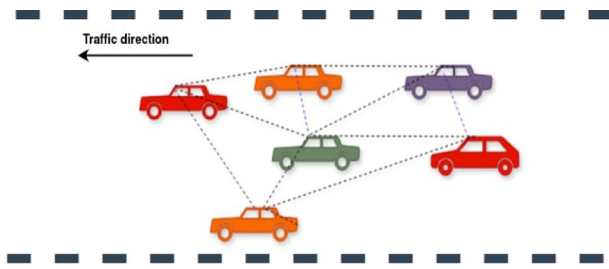


Fig. 2 Visualization of vehicles on a road as nodes, with edges representing vehicle-to-vehicle interactions at time step t . Each vehicle is depicted as a node, and edges between nodes illustrate interactions determined by proximity between vehicles at the given time step

$$\alpha_{v_i v_j}^t = \frac{\exp(\text{LeakyReLU}(a \cdot^T [\mathbf{W}z_{v_i}^t \parallel \mathbf{W}z_{v_j}^t]))}{\sum_{k \in \mathcal{N}_{v_i}} \exp(\text{LeakyReLU}(a \cdot^T [\mathbf{W}z_{v_i}^t \parallel \mathbf{W}z_{v_k}^t]))} \quad (4)$$

where \parallel denotes concatenation, \cdot^T represents transpose, $\alpha_{v_i v_j}^t$ is the attention coefficient of node v_j to v_i at time-step t , \mathcal{N}_{v_i} represents the neighbors of node v_i on the graph, $\mathbf{W} \in \mathbb{R}^{F \times F}$ is the weight matrix of a shared linear transformation applied to each node, and $a \in \mathbb{R}^{2F}$ is the weight vector of a single-layer feedforward neural network. The softmax function with LeakyReLU activation is applied for normalization. After obtaining the normalized attention coefficients, the output of one graph attention layer for node v_i at time t is given by:

$$\hat{z}_{v_i}^t = \sigma \left(\sum_{v_j \in \mathcal{N}_{v_i}} \alpha_{v_i v_j}^t \mathbf{W}z_{v_j}^t \right) \quad (5)$$

where σ is a nonlinear function, $\hat{z}_{v_i}^t$ is the result after graph attention layers, is the aggregated hidden state for vehicle v_i at time t , which contains the spatial influence from other vehicles. By integrating GAT-based modules into our model, we can effectively model the spatial interactions among vehicles,

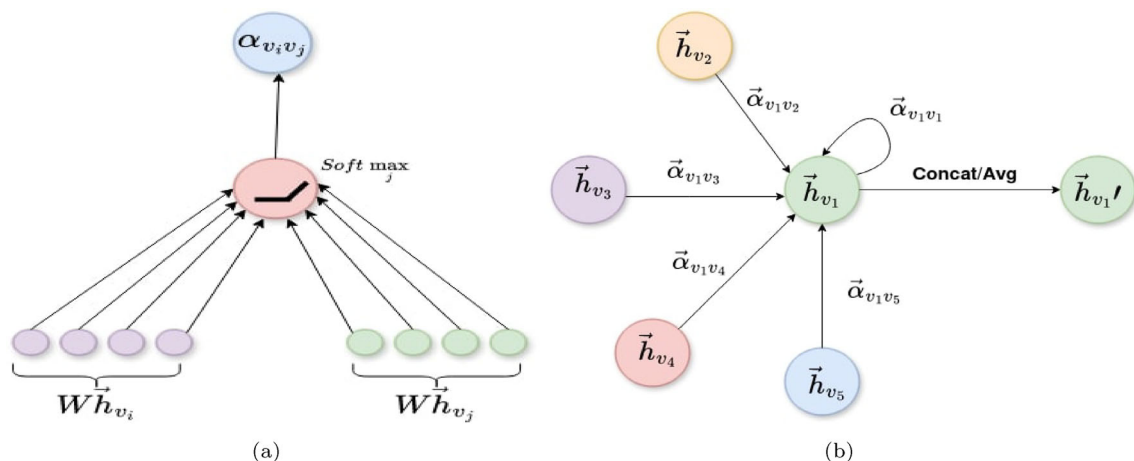


Fig. 3 Illustration of attention coefficient calculation (a) and hidden layer feature generation (b) in a graph attention network (GAT), demonstrating attention computation and feature aggregation among nodes in a graph structure

enabling more accurate predictions and analysis of traffic behavior.

3.2.3 LSTM-based module for capturing temporal correlations

In traditional LSTM-based methods for modeling interactions in traffic scenarios, the sharing of hidden states among vehicles is a common strategy. However, these methods often neglect to explicitly capture the temporal correlations between these interactions. In our approach, we introduce an additional LSTM to explicitly model these temporal correlations, termed as MV-LSTM:

$$m_{v_i}^t = \text{MV-LSTM}(m_{v_i}^{t-1}, \hat{z}_{v_i}^t; W_m) \quad (6)$$

Here, $\hat{z}_{v_i}^t$ is obtained from Eq. (5). The parameter W_m represents the weight matrix of the MV-LSTM, which is shared across all sequences. Within the encoder module as illustrated in Fig. 1, we utilize two LSTMs, namely V-LSTM and MV-LSTM, to capture the motion patterns of each vehicle and the temporal correlations of interactions, respectively. The fusion of spatial and temporal information is achieved by integrating these two modules. At time-step T_p , two hidden variables ($z_{v_i}^{T_p}, m_{v_i}^{T_p}$) are obtained from the two LSTMs for each vehicle. In our implementation, these variables are processed through separate multilayer perceptrons $v_1(\cdot)$ and $v_2(\cdot)$ before being concatenated as follows:

$$\bar{z}_{v_i} = v_1(z_{v_i}^{T_p}) \quad (7)$$

$$\bar{m}_{v_i} = v_2(m_{v_i}^{T_p}) \quad (8)$$

$$s_{v_i} = \bar{z}_{v_i} \parallel \bar{m}_{v_i} \quad (9)$$

Here, \parallel denotes concatenation.

3.2.4 LSTM-decoder module for future vehicle trajectory prediction

In vehicle trajectory prediction, our objective is to anticipate future movements of vehicles based on real-world trajectory datasets. Our model’s intermediate state vector comprises two parts: the hidden variables of V-LSTM, the hidden variables of MV-LSTM. The intermediate state vector is calculated as:

$$s_{v_i}^{T_P} = s_{v_i} \tag{10}$$

where s_{v_i} is obtained from Eq. (9). This intermediate state vector serves as the initial hidden state of the decoder LSTM, termed as D-LSTM.

The predicted relative position is given by

$$(\Delta x_{v_i}^{T_P+1}, \Delta y_{v_i}^{T_P+1}) = v_3(s_{v_i}^{T_P+1}) \tag{11}$$

where

$$\hat{s}_{v_i}^{T_P+1} = \text{D-LSTM}(s_{v_i}^{T_P}, p_{v_i}^{T_P}; W_s) \tag{12}$$

with W_s representing the D-LSTM weight, and $v_3(\cdot)$ denoting a linear layer. After obtaining the predicted relative position at time-step $T_P + 1$, the subsequent inputs of D-LSTM are calculated based on the last predicted relative position. Converting relative positions to absolute positions facilitates computing the loss.

3.3 Loss function

In our model training process, we employ the Mean Squared Error (MSE) loss function, as detailed in [41]. This loss function quantifies the disparity between predicted and observed trajectory data. The overall loss, denoted as \mathbb{L} , is computed using the following formula:

$$\mathcal{L} = \frac{1}{T_F} \sum_{t=1}^{T_F} \mathcal{L}^t = \frac{1}{T_F} \sum_{t=1}^{T_F} \|\mathcal{Y}_{\text{pred}}^t - \mathcal{Y}_{\text{real}}^t\| \tag{13}$$

Here, T_F represents the prediction window, \mathcal{L}^t signifies the loss at instance t , and $\mathcal{Y}_{\text{pred}}^t$ and $\mathcal{Y}_{\text{real}}^t$ denote the predicted and real trajectory data, respectively. The loss function computes the mean squared difference between the predicted and real trajectories over the prediction window, providing a measure of the model’s predictive accuracy.

4 Dataset

Our model has been trained and evaluated using two urban datasets: the Apolloscape trajectory prediction dataset [42]

Table 1 Urban dataset used to train the proposed model

Dataset	Train	Test	Validation
Apolloscape	18,905	2338	4964
Argoverse	232,566	25,626	66,365

and the Argoverse Motion Forecasting dataset. While the Apolloscape dataset provides information about object types, the Argoverse dataset lacks proper object definitions. Our aim was to create a universal model capable of accommodating all datasets. However, even in the Apolloscape dataset, there are vague classes such as ‘others,’ and uncertainties persist regarding the sizes of vehicles.

Despite these challenges, our model focuses more on the maneuver patterns and surrounding context of objects rather than relying solely on object size or type. This approach is supported by the fact that object positions are typically given as centroids, reducing the influence of object size or type on trajectory prediction. Instead, the model considers factors such as surrounding vehicles and changes in movement over time, leveraging Graph Attention Layers and LSTM for predicting future trajectories. Details regarding the training, testing, and validation sets for the three datasets are provided in Table 1.

4.1 Apolloscape dataset

The Apolloscape trajectory dataset [42] comprises camera-based images, LiDAR-generated point clouds, and meticulously annotated trajectories. This dataset captures diverse urban environments with varying traffic densities and lighting conditions. Notably, it encompasses a diverse range of dynamic entities, including vehicles, cyclists, and pedestrians, navigating through highly complex traffic scenarios. The Apolloscape dataset comprises 53 min of training sequences and 50 min of testing sequences, all captured at a consistent frame rate of 2 s. Each line in the text files of the dataset provides detailed information about objects at specific time frames, including frame ID, object ID, object type, position coordinates (x, y, z), object dimensions (length, width, height), and heading. The positions are given in meters within the world coordinate system.

4.2 The Argoverse dataset

The Argoverse Motion Forecasting dataset is sourced from 1006h of driving data collected in Pittsburgh and Miami, USA. This dataset facilitates 3D tracking and motion forecasting in urban settings. With approximately 200 million trajectories recorded at a sampling rate of 10 Hz, trajectories are segmented into 5-s intervals for analysis. For this study,

each trajectory sequence is resampled from 10 to 5 Hz to maintain consistency in the investigation. These datasets provide rich and diverse sources of data, enabling our model to learn and generalize across a wide spectrum of urban driving scenarios, thereby enhancing its effectiveness in real-world applications. The dataset is organized into CSV files, with each file corresponding to a scenario. Each row in the CSV files includes a timestamp, object ID, object type, x and y coordinates representing the object's position, and the data collection location.

4.3 Pre-processing

The datasets undergo pre-processing before being converted into text files, where each row comprises the columns: Frame ID, Object ID, X coordinate, Y Coordinate, and Dataset ID. Frame IDs are adjusted to range between 1 and n , and Object IDs between 1 and N , ensuring sequential and nonmissing IDs. Dataset IDs differentiate scenes within a dataset, as objects or agents may vary across scenes. The training sequence length and prediction sequence length for each dataset are determined based on the dataset's frame rate and the desired time intervals. If a dataset is too large, only selected scenes may be considered using Dataset IDs. Finally, pickle files are generated for training and evaluation. The size of training, testing, and validation samples is determined by the number of T-second-long observation sequences of road agent positions, with T set to 3 in this case.

5 Experiments and results analysis

In this section, we evaluate our method using two publicly available trajectory datasets: Argoverse and Apolloscape trajectory dataset. We use metrics such as average displacement error (ADE) and final displacement error (FDE) to quantitatively compare the performance of our method against existing state-of-the-art works.

5.1 Training of the model

In our implementation, we employed a neural network architecture consisting of long short-term memory (LSTM) layers, a graph attention layer, and activation functions to capture complex temporal and relational patterns in the data. Below, we provide a detailed overview of the key components and hyperparameters utilized.

LSTM layers: Each LSTM layer is configured as a single layer to balance model complexity and computational efficiency. The dimensions of the input and output tensors, denoted as $p_{v_i}^t$ and $z_{v_i}^t$, respectively, were set to 16 and 32. These dimensions were determined through experimentation to capture relevant features and ensure model expressiveness.

Graph attention layer: The graph attention layer plays a crucial role in modeling inter-node dependencies and capturing global graph structures. We utilized a weight matrix W with a shape of 16×32 to compute attention scores between nodes. The attention vector a was set to a dimension of 64, enabling the model to focus on relevant graph nodes during computation. Batch normalization was applied to the input of the graph attention layer to stabilize and accelerate the training process.

Activation functions: Activation functions $v_1(\cdot)$ and $v_2(\cdot)$ were employed to introduce nonlinearity into the model and facilitate feature transformation. Each activation function consists of 3 layers with Rectified Linear Unit (ReLU) activation functions. The number of hidden nodes in these layers was set to 32, 64, and 24 for $v_1(\cdot)$, and 32, 64, and 16 for $v_2(\cdot)$, respectively. These configurations were chosen to promote feature extraction and hierarchical representation learning.

Training configuration: We adopted the Adam optimizer with a learning rate of 0.01 to facilitate efficient optimization of the network parameters. A batch size of 64 was utilized during training to balance computational efficiency and model convergence. These hyperparameters were fine-tuned through experimentation on the training dataset to achieve optimal model performance and generalization capability.

5.2 Evaluation metrics

The evaluation of the proposed model performance in forecasting future trajectories relies on several key metrics, as referenced in citations [41] and [42]. These metrics provide quantitative insights into the model's effectiveness in predicting the trajectory of road agents within the dataset.

Average displacement error (ADE): ADE quantifies the average Euclidean distance between the predicted trajectory data and the real trajectory data over the prediction interval. It is computed as follows:

$$\text{ADE}_{v_i} = \frac{1}{T_F} \sum_{t=T_p+1}^{T_F} \sqrt{\sum_{i=1}^N (\hat{x}_{v_i}^t - x_{v_i}^t)^2 + (\hat{y}_{v_i}^t - y_{v_i}^t)^2} \quad (14)$$

Here, N represents the number of observed road agents in the traffic scene, T_F denotes the number of predicted time-steps or frames, and $\hat{x}_{v_i}^t$ and $\hat{y}_{v_i}^t$ are the predicted x and y -coordinates of road agent v_i at time instance t . Similarly, $x_{v_i}^t$ and $y_{v_i}^t$ represent the real x and y -coordinates of road agent v_i at instance t .

Final displacement error (FDE): FDE measures the mean Euclidean distance between the last predicted trajectory data and the real trajectory data of road agents. It is calculated as:

$$\text{FDE}_{v_i} = \sqrt{(\hat{x}_{v_i}^{T_F} - x_{v_i}^{T_F})^2 + (\hat{y}_{v_i}^{T_F} - y_{v_i}^{T_F})^2} \quad (15)$$

Here, T_F represents the last instance or frame in the predicted trajectory data, and $\hat{x}_{v_i}^{T_F}$ and $\hat{y}_{v_i}^{T_F}$ denote the predicted x and y -coordinates of road agent v_i at the last instance T_F . Likewise, $x_{v_i}^{T_F}$ and $y_{v_i}^{T_F}$ represent the real x and y -coordinates of road agent v_i at instance T_F . The overall FDE is computed by averaging the FDEs of all observed road agents. This metric provides insights into the model's long-term forecasting capabilities.

5.3 Baseline methods

To comprehensively assess the effectiveness of our proposed model, we conducted a comprehensive comparison with various state-of-the-art methodologies using urban datasets. Each of these baseline methods represents a distinct approach to trajectory prediction, incorporating various techniques and architectures to capture the complex dynamics of urban traffic.

Lane attention and trajectory prediction: Unifying lane features with an encoder–decoder network, this method adeptly forecasts future trajectories by incorporating lane-related context into the prediction process [43].

LaPred: Leveraging a CNN-LSTM network, LaPred synthesizes nearby agents' joint characteristics with lane and trajectory data to provide future trajectory predictions, emphasizing contextual understanding in trajectory forecasting [35].

DATF: Pioneering a holistic approach, DATF incorporates signals from the multimodal world and dynamic agent interactions, drawing insights from environmental context to inform trajectory predictions across varied pathways [44].

VectorNet: Distinguished by its hierarchical graph neural network architecture, VectorNet elegantly captures the geographical locality and high-order interactions between road components, enriching trajectory predictions with spatial context [45].

UST: A straightforward trajectory prediction model, UST, relies primarily on spatio-temporal pooling for sequence prediction, offering a simplified yet effective approach to trajectory forecasting [46].

GRIP and GRIP++: These methods harness LSTM-based trajectory prediction coupled with Graph Convolutional Networks (GCN), with GRIP++ further enhancing GCNs with a GRU-based encoder–decoder, propelling graph-based trajectory prediction techniques [18, 41].

SCOUT: By leveraging attention mechanisms and Graph Neural Networks (GNN), SCOUT meticulously models trajectory prediction, dynamically allocating attention to relevant spatial and temporal features [47].

TraPHic: Employing a hybrid LSTM-CNN network, TraPHic capitalizes on dynamic weighted graph representation to encapsulate the intricate interactions among dynamic agents in the traffic, thereby enriching trajectory predictions [29].

Graph LSTM: This model synthesizes LSTM for trajectory prediction with the power of graph neural networks, enabling the extraction of high-dimensional features from complex trajectory data [48].

CS-LSTM: By integrating convolutional social pooling within an LSTM encoder–decoder architecture, CS-LSTM adeptly captures interdependencies in vehicle movement, leveraging spatial context to inform trajectory predictions [49].

TrafficPredict: Pioneering an LSTM-based trajectory prediction model, TrafficPredict innovatively represents trajectories and their interactions within a 4D graph framework using an instance layer [42].

Social GAN: This method fuses generative adversarial networks and sequence-to-sequence models to amplify trajectory predictions, leveraging adversarial training to enhance the fidelity of generated trajectories [50].

Multiscale spatial–temporal graph: The authors introduce a novel framework for autonomous vehicle trajectory prediction, integrating spatial–temporal layers, dilated temporal convolutions, and LSTM-based trajectory generation module [51].

HVTD: The hierarchical vector transformer diffusion model (HVTD) is a novel trajectory prediction method for autonomous driving, combining local and global information acquisition, aleatoric uncertainty capture, and adaptive graph-based spatial–temporal feature extraction for superior speed and accuracy [52].

5.4 Analysis of results

In our evaluation, we compared the performance of the proposed model with these baseline methods on both the Apolloscape and Argoverse datasets. The comparison was based on key metrics such as ADE and FDE, which provide insights into the accuracy of trajectory predictions. Our analysis revealed several noteworthy observations: While some models showcased commendable performance across both datasets, others exhibited variability in their effectiveness, highlighting the nuanced challenges inherent in urban trajectory prediction. Models that demonstrated proficiency in capturing contextual cues, such as lane features and dynamic agent interactions, tended to exhibit superior performance. Certain models excelled in short-term predictions but faced challenges in accurately forecasting long-term trajectories, underscoring the complexity of capturing temporal dependencies in trajectory data.

Table 2 Performance evaluation of the proposed model using cutting-edge techniques on the Argoverse and Apolloscape datasets (Validation =3 s)

Models	Apolloscape		Argoverse	
	ADE	FDE	ADE	FDE
Proposed model	1.19	1.85	0.89	1.58
Lane attention and trajectory prediction	–	–	1.24	2.49
LaPred	–	–	1.48	3.29
Multiscale spatial–temporal graph	1.9	3.5	–	–
HVTD	–	–	1.59	1.27
DATF	–	–	2.04	3.69
VectorNet	–	–	1.66	2.21
UST	1.24	2.25	1.47	2.94
GRIP++	1.25	2.36	–	–
GRIP	1.25	2.34	–	–
SCOUT	1.26	2.25	–	–
TraPHic	1.283	11.699	1.039	3.079
CS-LSTM	2.144	11.699	1.050	3.085
Graph LSTM	3.59	3.71	2.40	3.09
Social GAN	3.980	6.750	3.610	5.390
TrafficPredict	8.588	24.2262	–	–

Bold values indicate that the proposed model's results, allowing for differentiation from other state-of-the-art methods

As illustrated in Table 2, the proposed model consistently outperforms the comparison models. Notably, both ADE and FDE values show significant decreases compared to other models, except for exceptions such as GRIP++, UST, VectorNet, and Graph LSTM, where FDE values are higher than those of the reported model. Moreover, the proposed model surpasses GCN-based approaches and achieves the highest accuracy for both short- and long-term predictions. This success is attributed to the GAT layer, which effectively captures interaction patterns with selected importance, aiding in learning the localization patterns of surrounding vehicles. On Apolloscape dataset, TrafficPredict exhibits limitations in accurately forecasting road agents' movements, as evidenced by high ADE and FDE values. Social GAN, Graph LSTM, Multiscale spatial–temporal graph, and CS-LSTM demonstrate competitive performance in capturing complex interactions and improving trajectory predictions. TraPHic excels in short-term predictions but shows limitations in accurately forecasting long-term trajectories. The proposed model emerges as the top-performing model on the Apolloscape dataset, showcasing remarkable accuracy with small ADE and FDE values.

On Argoverse dataset, similar trends are observed on the Argoverse dataset, with the proposed model maintaining superior performance. Social GAN, Graph LSTM, and CS-LSTM continue to demonstrate competitive performance in capturing diverse urban traffic scenarios. The proposed model sustains its superior performance, emphasizing its efficacy in both short- and long-term trajectory predictions on the Argoverse dataset. In Fig. 4, we present visualizations

of prediction results across varying traffic conditions such as mild, moderate, and congested utilizing datasets from the Apolloscape and the Argoverse. Upon examining 3 s of historical trajectories, our model forecasts trajectories for a 5 s future horizon. Furthermore, the visualizations provided in Fig. 4 highlight the striking comparability between estimated and actual trajectories, demonstrating the effectiveness of the proposed model when paired with graph attention and two types of LSTM models. The model's ability to perform well in complex urban traffic settings underscores its general suitability for trajectory prediction in autonomous vehicles.

On an average, the proposed model achieves a prediction time of 0.65 s over 30 frames, with a runtime of 30.62 frames per second (fps). We define the size of the adjacency matrix as $M \times M$, where M represents the maximum number of objects detected in a single frame. This prediction time is influenced by the value of M , which we analyze further. Examining the traffic densities of the datasets, we observe an average density of 3.19 for the Apolloscape dataset and 1.03 for the Argoverse dataset. Considering Apolloscape as the denser dataset, we set M to its maximum value of 120, resulting in a prediction time of 0.112 s for a one-second-ahead prediction. Table 3 illustrates the variations in prediction time as M increases. Real-time prediction becomes infeasible for $M \leq 650$, where $M = 650$ indicates the observation of 650 road agents in a single frame by the camera or LIDAR on the Ego vehicle's roof a scenario rare in real-world heavy traffic conditions.

The impact of threshold distance on constructing a graph for vehicle prediction models is crucial for understanding how surrounding objects affect prediction accuracy. By defin-

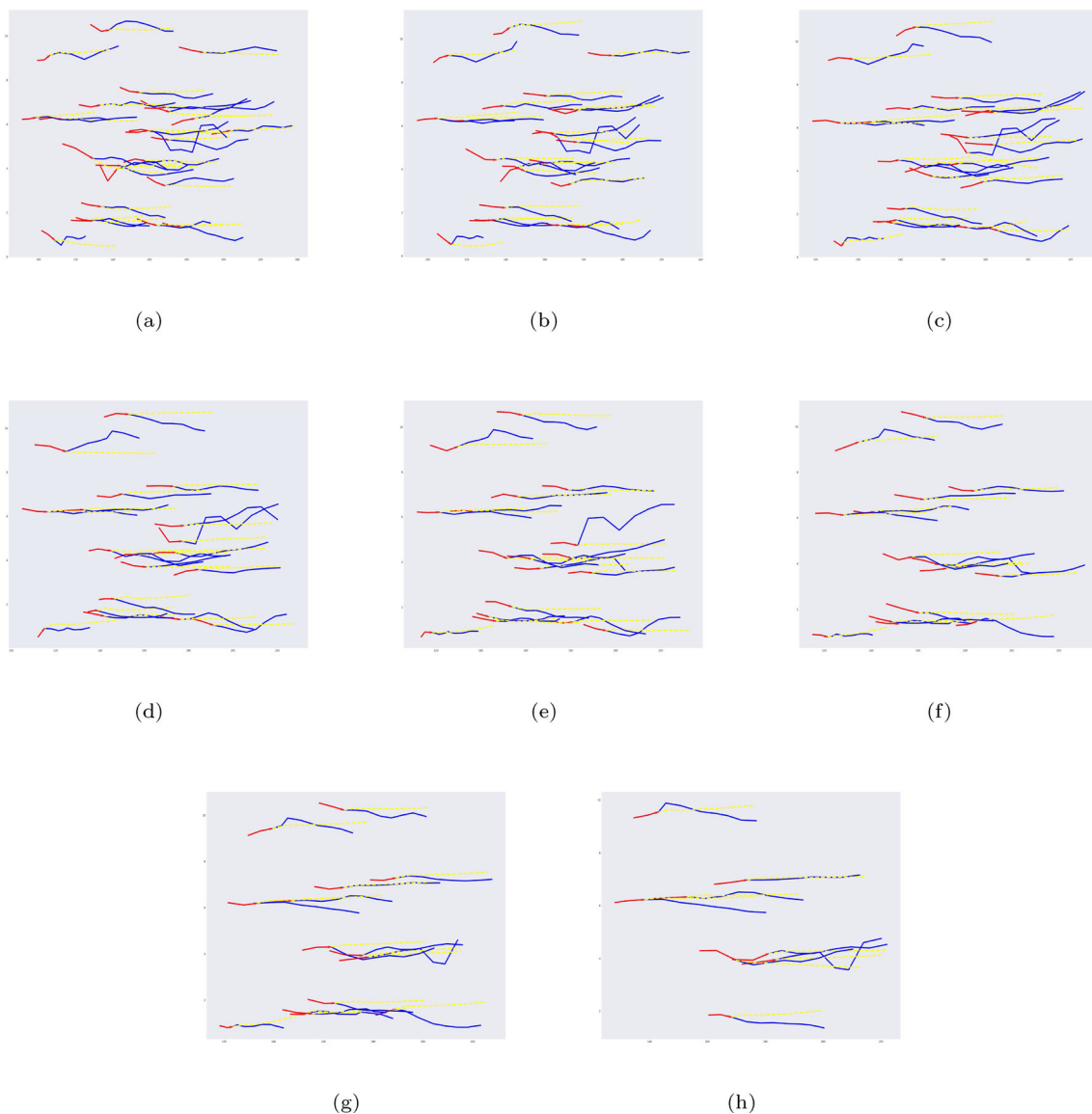


Fig. 4 Visualization of vehicle trajectories from the Apolloscape and Argoverse datasets. Red lines denote past observed trajectories, blue lines represent ground truth trajectories, and yellow lines depict pre-

dicted trajectories. Each trajectory reflects the movement history and future predictions of multiple vehicles within the scene

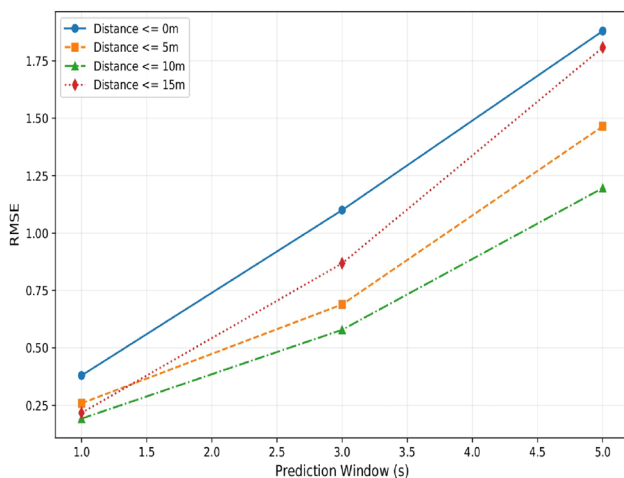
ing a threshold distance, we determine the proximity of vehicles and investigate how varying this threshold affects model performance. Figure 5 presents a comparison of results for different threshold distances. Notably, when the threshold distance is less than or equal to 0 m (as depicted by the blue bars in Fig. 5), indicating no consideration of surrounding objects, the prediction error is higher compared to scenarios where nearby vehicles are taken into account ($\text{Distance} \leq 0$ m). This underscores the importance of considering surrounding objects in enhancing prediction accuracy.

Moreover, as the threshold distance increases, the prediction error initially decreases, suggesting better performance as more surrounding objects are considered. However,

beyond a certain threshold (for example, $\text{Distance} > 10$ m), the prediction error starts to increase. This observation implies that an excessive number of surrounding objects can degrade prediction accuracy, indicating a balance between considering enough surrounding objects for accurate predictions and avoiding information overload. Additionally, the graph shows that prediction errors tend to increase as we move from left to right, reflecting the decreasing influence of front objects on predictions. This suggests that objects closer to the vehicle have a more significant impact on prediction accuracy compared to those farther away. Furthermore, it is noteworthy that predicting farther into the future (e.g., prediction window 5 s) results in higher prediction errors compared

Table 3 Variation in the time required for one-second-ahead prediction as the value of M increases

M	Time for 1-second-ahead prediction (s)
120	0.112
240	0.178
480	0.331
500	0.621
550	0.682
600	0.733
625	0.778
630	0.985
650	1.015

**Fig. 5** Comparison of various threshold distance on Apolloscape dataset

to shorter-term predictions (prediction window 1 s). This is expected due to the inherent uncertainty in forecasting distant future states, highlighting the challenges associated with long-term prediction. It is also important to mention that a threshold distance of 10 m provides good accuracy for both short-term and long-term predictions. On the other hand, a threshold distance of 5 m yields less accurate longer-term predictions due to the reduced influence of surrounding vehicles. This indicates the necessity of balancing the threshold distance based on the desired prediction accuracy and the influence of surrounding vehicles.

5.5 Future research directions

By incorporating temporal correlations of interactions into vehicle trajectory prediction models, we empower vehicles to make more informed decisions, leveraging a deeper understanding of the dynamic interactions within the traffic environment. This approach enhances the predictive capabilities of autonomous vehicles and contributes to overall

traffic safety and efficiency. However, one limitation of our proposed model is its exclusive focus on the motion of other vehicles, overlooking contextual or intentional factors influencing their movements. Future research endeavors could explore methods to integrate contextual information, such as road layout and traffic regulations, as well as discern the intentions of other drivers, thereby enhancing the accuracy of trajectory predictions.

6 Conclusion

In this work, we proposed a novel deep learning approach for real-time vehicle trajectory prediction on urban roads. Leveraging techniques from graph neural networks and LSTM-based models, we addressed the challenges associated with capturing complex interactions between vehicles and modeling the uncertainty in future trajectories. By representing vehicles as nodes on a graph and employing graph attention mechanisms, we effectively captured spatial dependencies and interactions between vehicles in crowded urban environments. Our model, incorporating multiple LSTM components for encoding motion patterns and temporal correlations, achieved fusion of spatial and temporal information, leading to improved prediction accuracy. Through extensive experimentation and evaluation, we demonstrated the effectiveness of our approach in generating multiple plausible trajectories and outperforming baseline methods. By evaluating using standard error metrics such as ADE and FDE, we quantified the predictive performance of our model. In conclusion, our proposed method offers a promising solution for real-time vehicle trajectory prediction, which is crucial for applications such as autonomous driving, traffic management, and intelligent transportation systems. Future work could explore further enhancements, such as incorporating additional contextual information to further improve prediction accuracy and robustness in diverse urban environments.

Acknowledgements The authors thank the reviewers for their valuable feedback in enhancing the quality of the manuscript.

Author Contributions Sundari K helped in conceptualization, computational work, methodology, data analysis, manuscript writing. Dr. A Senthil Thilak contributed to supervision, contribution to manuscript writing and improvement of methods, project guidance.

Funding This research received no specific funding.

Data availability The dataset supporting the findings of this study is publicly available and can be accessed from the following link. Apolloscape dataset: <https://apolloscape.auto/trajectory.html> Argov-se dataset: <https://www.argovse.org/av2.html>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Bharilya, V., Kumar, N.: Machine learning for autonomous vehicle's trajectory prediction: a comprehensive survey, challenges, and future research directions. *Veh. Commun.* (2024). <https://doi.org/10.1016/j.vehcom.2024.100733>
- Nayak, A., Eskandarian, A.: Cooperative probabilistic trajectory prediction under occlusion. *IEEE Trans. Intell. Veh.* (2024). <https://doi.org/10.1109/TIV.2024.3365651>
- Li, H., Wang, X., Su, X., Wang, Y.: Improved gaussian mixture probabilistic model for pedestrian trajectory prediction of autonomous vehicle. *Recent Patents Mech. Eng.* **17**(1), 65–75 (2024). <https://doi.org/10.2174/0122127976268211231110055647>
- Yuan, H., Li, G.: A survey of traffic prediction: from spatio-temporal data to intelligent transportation. *Data Sci. Eng.* **6**(1), 63–85 (2021). <https://doi.org/10.1007/s41019-020-00151-z>
- Owais, M.: Deep learning for integrated origin-destination estimation and traffic sensor location problems. *IEEE Trans. Intell. Transp. Syst.* (2024). <https://doi.org/10.1109/TITS.2023.3344533>
- Alshehri, A., Owais, M., Gyani, J., Aljarbou, M.H., Alsulamy, S.: Residual neural networks for origin-destination trip matrix estimation from traffic sensor information. *Sustainability* **15**(13), 9881 (2023). <https://doi.org/10.3390/su15139881>
- Owais, M., Moussa, G.S., Hussain, K.F.: Robust deep learning architecture for traffic flow estimation from a subset of link sensors. *J. Transport. Eng. Part A Syst.* **146**(1), 04019055 (2020). <https://doi.org/10.1061/JTEPBS.0000290>
- Moussa, G.S., Owais, M., Dabbour, E.: Variance-based global sensitivity analysis for rear-end crash investigation using deep learning. *Accid. Anal. Prev.* **165**, 106514 (2022). <https://doi.org/10.1016/j.aap.2021.106514>
- Owais, M., Alshehri, A., Gyani, J., Aljarbou, M.H., Alsulamy, S.: Prioritizing rear-end crash explanatory factors for injury severity level using deep learning and global sensitivity analysis. *Expert Syst. Appl.* **245**, 123114 (2024). <https://doi.org/10.1016/j.eswa.2023.123114>
- Park, S.H., Kim, B., Kang, C.M., Chung, C.C., Choi, J.W.: Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder–decoder architecture. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1672–1678. IEEE (2018). <https://doi.org/10.1109/IVS.2018.8500658>
- Xie, G., Shanguan, A., Fei, R., Ji, W., Ma, W., Hei, X.: Motion trajectory prediction based on a CNN-LSTM sequential model. *Sci. China Inf. Sci.* **63**, 1–21 (2020)
- Dai, S., Li, L., Li, Z.: Modeling vehicle interactions via modified LSTM models for trajectory prediction. *IEEE Access* **7**, 38287–38296 (2019). <https://doi.org/10.1109/ACCESS.2019.2907000>
- Kim, B., Kang, C.M., Kim, J., Lee, S.H., Chung, C.C., Choi, J.W.: Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 399–404. IEEE (2017). <https://doi.org/10.1109/ITSC.2017.8317943>
- Ip, A., Irio, L., Oliveira, R.: Vehicle trajectory prediction based on LSTM recurrent neural networks. In: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), pp. 1–5. IEEE (2021). <https://doi.org/10.1109/VTC2021-Spring51267.2021.9449038>
- 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.* **32**(1), 4–24 (2020). <https://doi.org/10.1109/TNNLS.2020.2978386>
- Lu, Y., Wang, W., Hu, X., Xu, P., Zhou, S., Cai, M.: Vehicle trajectory prediction in connected environments via heterogeneous context-aware graph convolutional networks. *IEEE Trans. Intell. Transport. Syst.* (2022)
- Li, X., Ying, X., Chuah, M.C.: Grip++: enhanced graph-based interaction-aware trajectory prediction for autonomous driving (2019). arXiv preprint [arXiv:1907.07792](https://arxiv.org/abs/1907.07792). <https://doi.org/10.48550/arXiv.1907.07792>
- Li, X., Ying, X., Chuah, M.C.: Grip: graph-based interaction-aware trajectory prediction. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 3960–3966. IEEE (2019)
- Choi, S., Kim, J., Yeo, H.: Attention-based recurrent neural network for urban vehicle trajectory prediction. *Proc. Comput. Sci.* **151**, 327–334 (2019)
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks (2017). arXiv preprint [arXiv:1710.10903](https://arxiv.org/abs/1710.10903). <https://doi.org/10.48550/arXiv.1710.10903>
- Mohades Deilami, F., Sadr, H., Tarkhan, M.: Contextualized multidimensional personality recognition using combination of deep neural network and ensemble learning. *Neural Process. Lett.* **54**(5), 3811–3828 (2022). <https://doi.org/10.1007/s11063-022-10787-9>
- Kalashami, M.P., Pedram, M.M., Sadr, H., et al.: EEG feature extraction and data augmentation in emotion recognition. *Comput. Intell. Neurosci.* (2022). <https://doi.org/10.1155/2022/7028517>
- Khodaverdian, Z., Sadr, H., Edalatpanah, S.A., Nazari, M.: An energy aware resource allocation based on combination of CNN and GRU for virtual machine selection. *Multimed. Tools Appl.* **83**(9), 25769–25796 (2024)
- Lin, L., Gong, S., Li, T., Peeta, S.: Deep learning-based human-driven vehicle trajectory prediction and its application for platoon control of connected and autonomous vehicles. In: *The Autonomous Vehicles Symposium*, vol. 2018 (2018)
- Kavran, D., Mongus, D., Žalik, B., Lukač, N.: Graph neural network-based method of spatiotemporal land cover mapping using satellite imagery. *Sensors* **23**(14), 6648 (2023). <https://doi.org/10.3390/s23146648>
- Li, X., Ying, X., Chuah, M.C.: Grip: graph-based interaction-aware trajectory prediction. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 3960–3966. IEEE (2019). <https://doi.org/10.1109/ITSC.2019.8917228>
- Sheng, Z., Xu, Y., Xue, S., Li, D.: Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **23**(10), 17654–17665 (2022). <https://doi.org/10.1109/TITS.2022.3155749>
- Xu, D., Shang, X., Liu, Y., Peng, H., Li, H.: Group vehicle trajectory prediction with global spatio-temporal graph. *IEEE Trans. Intell. Veh.* **8**(2), 1219–1229 (2022). <https://doi.org/10.1109/TIV.2022.3200338>
- Chandra, R., Bhattacharya, U., Bera, A., Manocha, D.: Traffic: trajectory prediction in dense and heterogeneous traffic using weighted interactions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8483–8492 (2019)
- Jo, E., Sunwoo, M., Lee, M.: Vehicle trajectory prediction using hierarchical graph neural network for considering interaction among multimodal maneuvers. *Sensors* **21**(16), 5354 (2021)
- Cao, D., Li, J., Ma, H., Tomizuka, M.: spectral temporal graph neural network for trajectory prediction. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1839–1845. IEEE (2021)

32. Mo, X., Xing, Y., Lv, C.: Graph and recurrent neural network-based vehicle trajectory prediction for highway driving. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 1934–1939. IEEE (2021)
33. Liu, Y., Qi, X., Sisbot, E.A., Oguchi, K.: Multi-agent trajectory prediction with graph attention isomorphism neural network. In: 2022 IEEE Intelligent Vehicles Symposium (IV), pp. 273–279. IEEE (2022). <https://doi.org/10.1109/IV51971.2022.9827155>
34. Liu, S., Chen, X., Wu, Z., Deng, L., Su, H., Zheng, K.: Hega: heterogeneous graph aggregation network for trajectory prediction in high-density traffic. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 1319–1328 (2022)
35. Kim, B., Park, S.H., Lee, S., Khoshimjonov, E., Kum, D., Kim, J., Kim, J.S., Choi, J.W.: Lapred: Lane-aware prediction of multimodal future trajectories of dynamic agents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14636–14645 (2021)
36. Sadr, H., Nazari Soleimandarabi, M.: ACNN-TL: attention-based convolutional neural network coupling with transfer learning and contextualized word representation for enhancing the performance of sentiment classification. *J. Supercomput.* **78**(7), 10149–10175 (2022)
37. Jadidinejad, A.H., Sadr, H.: Improving weak queries using local cluster analysis as a preliminary framework. *Indian J. Sci. Technol.* **8**(5), 495–510 (2015). <https://doi.org/10.17485/ijst/2015/v8i15/46754>
38. Dai, S., Li, L., Li, Z.: Modeling vehicle interactions via modified LSTM models for trajectory prediction. *IEEE Access* **7**, 38287–38296 (2019)
39. Althé, F., La Fortelle, A.: An LSTM network for highway trajectory prediction. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 353–359. IEEE (2017)
40. Xing, Y., Lv, C., Cao, D.: Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles. *IEEE Trans. Veh. Technol.* **69**(2), 1341–1352 (2019)
41. Li, X., Ying, X., Chuah, M.: Grip++: enhanced graph-based interaction-aware trajectory prediction for autonomous driving, 2019. arXiv preprint arXiv (1907)
42. 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, vol. 33, pp. 6120–6127 (2019)
43. Luo, C., Sun, L., Dabiri, D., Yuille, A.: Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2370–2376 (2020)
44. Park, S.H., Lee, G., Seo, J., Bhat, M., Kang, M., Francis, J., Jadhav, A., Liang, P.P., Morency, L.-P.: Diverse and admissible trajectory forecasting through multimodal context understanding. In: Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, pp. 282–298. Springer (2020)
45. 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, pp. 11525–11533 (2020)
46. He, H., Dai, H., Wang, N.: Ust: unifying spatio-temporal context for trajectory prediction in autonomous driving. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5962–5969. IEEE (2020)
47. Carrasco, S., Llorca, D.F., Sotelo, M.: Scout: socially-consistent and understandable graph attention network for trajectory prediction of vehicles and VRUS. In: 2021 IEEE Intelligent Vehicles Symposium (IV), pp. 1501–1508. IEEE (2021)
48. Chandra, R., Guan, T., Panuganti, S., Mittal, T., Bhattacharya, U., Bera, A., Manocha, D.: Forecasting trajectory and behavior of road-agents using spectral clustering in graph-LSTMS. *IEEE Robot. Autom. Lett.* **5**(3), 4882–4890 (2020)
49. 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, pp. 1468–1476 (2018)
50. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social GAN: socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2255–2264 (2018)
51. Tang, L., Yan, F., Zou, B., Li, W., Lv, C., Wang, K.: Trajectory prediction for autonomous driving based on multiscale spatial-temporal graph. *IET Intel. Transport. Syst.* **17**(2), 386–399 (2023). <https://doi.org/10.1049/itr2.12265>
52. Tang, Y., He, H., Wang, Y.: Hierarchical vector transformer vehicle trajectories prediction with diffusion convolutional neural networks. *Neurocomputing* **580**, 127526 (2024). <https://doi.org/10.1016/j.neucom.2024.127526>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.