SCIPEDIA Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

RIMNI Revista Internacional Métodos numéricos para cálculo y diseño en ingeniería

MIMTP: Mamba-Driven Interaction-Aware Multi-Modal Trajectory Prediction for Autonomous Driving

Jingen Li[1], **Lukun Wang**[1,*] **and Jiaming Pei**[2]

[1]College of Intelligent Equipment, Shandong University of Science and Technology, Tai'an, China

[2]School of Computer Science, The University of Sydney, Sydney, NSW, Australia

ABSTRACT

Accurate prediction of future vehicle trajectories is essential for ensuring safety and reliable decision-making in autonomous driving systems. However, existing deep learning-based approaches exhibit several limitations. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) struggle to effectively model long-term temporal dependencies and complex agent interactions, while Transformer-based architectures often suffer from high computational complexity and limited efficiency. To overcome these challenges, this paper proposes an efficient Mamba-based feature extraction framework for jointly encoding vehicle trajectories and map information. By leveraging state-space modeling and a selective scanning mechanism, the proposed approach effectively captures long-range dependencies and enhances the representation of complex traffic behaviors. Specifically, raw scene data are first normalized and embedded into a unified feature space. A Mamba Encoder is then employed to extract high-level features from historical vehicle trajectories and map elements. Subsequently, Vehicle-Vehicle and Vehicle-Map interaction modules are introduced to explicitly model dynamic interactions among traffic participants and between vehicles and the surrounding map. The resulting high-dimensional features are further fused using an additional Mamba Encoder, while a Global Interaction Module is designed to capture scene-level dependencies. Finally, a Gated Recurrent Unit (GRU) decoder generates multi-modal future trajectory predictions. Experimental results on the Argoverse 1 dataset demonstrate that the proposed method achieves superior performance in terms of minADE, minFDE, and minMR, while maintaining high computational efficiency.

OPEN ACCESS

Received: 28/01/2026

Accepted: 16/04/2026

DOI
10.23967/j.rimni.2026.10.79799

Keywords:
Autonomous driving
multi-modal trajectory prediction
mamba
attention mechanism

1 Introduction

With the rapid advancement of autonomous driving and intelligent transportation systems, vehicle trajectory prediction has emerged as a critical task for ensuring driving safety and improving traffic efficiency. The primary objective of trajectory prediction is to exploit historical motion information together with road environment data to forecast the future movements of surrounding vehicles, thereby providing reliable support for behavior decision-making and path planning in autonomous driving

*Correspondence: Lukun Wang (wanglukun@sdust.edu.cn).

systems [1]. In complex traffic environments, driver behaviors are inherently uncertain and diverse, which poses significant challenges to accurate vehicle trajectory prediction [2].

Early studies primarily relied on physics-based models; however, such methods struggle to handle complex interaction behaviors and multi-modal driving intentions. In recent years, the emergence of deep learning has brought new breakthroughs to vehicle trajectory prediction. CNNs and RNNs have been employed to model spatio-temporal dependencies [3]. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) can capture multi-modal trajectory distributions, better addressing the inherent uncertainty of driving behaviors. Moreover, architectures such as Graph Neural Networks (GNNs) have been widely used to explicitly model interactions among vehicles, demonstrating stronger generalization capabilities in complex traffic scenarios.

Despite the substantial progress achieved by existing methods in feature extraction and spatio-temporal modeling, several challenges remain. On the one hand, CNN-based and RNN-based architectures exhibit inherent limitations in modeling long-term dependencies. Specifically, CNNs are constrained in capturing global contextual information due to their localized receptive fields, while RNNs suffer from gradient vanishing when processing long sequences, which hampers their ability to model long-range temporal dependencies.

On the other hand, Transformer-based architectures enable explicit modeling of global dependencies and have significantly improved spatio-temporal feature representation. For instance, Liu et al. [4] proposed mmTransformer, a stacked Transformer-based multimodal framework that integrates historical trajectories, high-definition (HD) map information, and surrounding vehicle contexts. Ngiam et al. [5] introduced SceneTransformer, which employs a Transformer encoder to model spatial layouts and contextual relationships among agents and map elements, producing scene-level embeddings. Similarly, Han et al. [6] presented Laneformer, which encodes lane geometry and agent trajectories as sequential inputs and leverages Transformer encoders to extract lane and vehicle kinematic features, effectively integrating structured map information with dynamic motion data.

In addition, Wang et al. [7] introduced a federated learning method based on dynamic sparsity, offering more efficient solutions for multi-device collaborative training. This approach has the potential to significantly improve the training efficiency and data sharing process in trajectory prediction models, where large-scale distributed learning is often required. Similarly, Chen et al. [8] proposed an intelligent group prediction algorithm for GPS trajectory based on vehicle communication, which integrates vehicle communication technology and an improved IGSA algorithm to enhance the prediction of traffic flow and vehicle locations, providing important ideas for improving trajectory prediction accuracy and efficiency in smart transportation systems. In related intelligent transportation research, Pei et al. explored neuro-VAE-based symbolic dynamic traffic management, which provides additional insights into modeling complex traffic systems and enhancing traffic intelligence [9].

However, despite their strong modeling capacity, Transformer architectures incur high computational and memory costs when applied to long-sequence modeling, which limits their efficiency in large-scale and real-time trajectory prediction scenarios.

To address these challenges, this paper proposes MIMTP, a Mamba-based efficient trajectory prediction method, which effectively extracts latent features from vehicle trajectories and HD map information. The Mamba model is grounded in the theory of Continuous-Time State Space Models (SSM) and employs a linear-time Selective Scan mechanism, which significantly reduces computational and memory costs while maintaining strong temporal modeling capabilities. Unlike Transformers, which require explicit computation of global attention, Mamba implicitly captures long-range dependencies, demonstrating superior efficiency and stability in long-sequence modeling.

The main contributions of this work are summarized as follows:

(1) We introduce Mamba as the feature extraction backbone for vehicle trajectory prediction, enabling efficient representation learning from both vehicle trajectories and HD map information. This design enhances the model's ability to capture long-range dependencies and complex traffic behaviors while significantly reducing computational complexity.

(2) Extensive experiments conducted on the Argoverse 1 and Argoverse 2 datasets demonstrate that the proposed MIMTP model achieves strong performance across multiple evaluation metrics, including minADE, minFDE, and minMR.

The remainder of this paper is organized as follows. Section 2 reviews related work, focusing on the strengths and limitations of existing feature extraction methods for trajectory prediction. Section 3 introduces the proposed Mamba-driven interaction-aware multimodal trajectory prediction model (MIMTP) and provides a detailed description of its key modules. Section 4 presents the experimental setup and dataset selection, followed by comprehensive comparative evaluations of model performance. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Related Works

In vehicle trajectory prediction tasks, an efficient feature extraction module is crucial for capturing rich temporal and spatial information from heterogeneous data sources, such as historical vehicle trajectories and HD map representations. Existing methods primarily rely on CNNs, RNNs, and attention mechanisms to encode scene context and historical motion dynamics.

CNN-based feature extraction methods typically take point clouds or RGB images as input and convert them into representations such as bird's-eye view (BEV) images to describe static and dynamic scenes, with convolutional layers used to extract spatio-temporal features [10,11]. The method proposed in [12] combines CNN layers with fully connected layers to capture temporally continuous features from historical trajectories for trajectory prediction. CoverNet encodes scene information into a rasterized bird's-eye view and utilizes CNNs to extract image features, enabling multi-modal behavior prediction [13]. Additionally, Strohbeck et al. [14] introduced a method for multiple trajectory prediction using deep temporal and spatial convolutional networks, which focuses on predicting complex vehicle interactions. DST-CNN uses CNNs to extract rasterized scene features and combines them with Temporal Convolutional Networks (TCNs) to capture historical trajectory features, which are then concatenated with current state features for multi-modal trajectory prediction. Chou et al. [15] integrate a bicycle-vehicle kinematic model into the CNN framework to extract spatial features on rasterized maps, ensuring kinematically feasible predicted trajectories. In MANTRA, CNNs are first used to interpret scene images and extract scene and trajectory features, which are subsequently fed into a Memory-Augmented Neural Network (MANN) to enhance the modeling of complex interaction behaviors [16]. Additionally, TPCN introduces point cloud feature extraction into the CNN framework to simultaneously capture spatial and temporal features for 3D point cloud trajectory prediction [17]. In summary, CNNs can efficiently extract spatial features from images or rasterized scenes and are suitable for capturing local patterns and multi-modal information. However, they are limited in their ability to capture global features.

RNN-based feature extraction methods primarily employ models such as Long Short-Term Memory (LSTM). Specifically, Xin et al. [18] use an LSTM module to extract the lateral driving intentions of the target vehicle (TV), which are then concatenated with longitudinal features and fed into another LSTM for trajectory prediction. Ding and Shen [19] encode the target vehicle's state using

an LSTM encoder to predict its driving intentions, while Zyner et al. [20] adopt a three-layer encoder-decoder LSTM to extract historical trajectory features and generate Gaussian Mixture Model (GMM) parameters. Additionally, Xing et al. [21] combine LSTMs with fully connected regression layers to simultaneously analyze temporal data and driving style features. In summary, RNNs can effectively capture temporal dependencies in trajectory sequences and handle variable-length inputs. However, they face challenges in modeling long-term dependencies and suffer from low parallelization efficiency.

Attention mechanisms have been widely applied in trajectory prediction tasks, as they explicitly model global temporal dependencies and spatial context information, effectively addressing the limitations of RNNs in long-term dependency modeling and CNNs in capturing global features. Kim et al. [22] utilize a multi-head attention module to extract environmental context information from lanes and vehicles, enhancing scene understanding. Wang et al. [23] employ multi-head attention to model interactions between vehicles and traffic states, and introduce Graph Attention Networks (GATs) to aggregate and model inter-node relationships. Meng et al. [24] incorporate attention mechanisms into graph-based modeling by using self-attention-based GATs to extract features of inter-vehicle interactions, and aggregate spatial information through attention weights assigned to nodes, thereby effectively modeling complex vehicle interactions and improving trajectory prediction performance. The HPNet model [25] introduces a Historical Prediction Attention (HPA) module and leverages self-attention for feature extraction and temporal modeling, dynamically optimizing trajectory representations to further enhance prediction consistency and stability. The MFAN model [26] incorporates a hybrid feature attention mechanism to adaptively measure the importance of different feature channels, enabling more effective extraction of individual and interaction information, thereby improving trajectory prediction accuracy and generalization. Although attention mechanisms and Transformer-based frameworks effectively overcome CNN limitations in modeling long-range dependencies and RNN limitations in parallel computation, significantly enhancing global feature interaction, they still face challenges of high computational complexity and memory consumption in high-dimensional dynamic scenarios.

In summary, existing trajectory prediction methods have developed multiple technical approaches for feature extraction. CNNs can effectively capture local spatial features but are limited in modeling global features. RNNs can model temporal dependencies but struggle with long sequences and exhibit low parallelization efficiency. Attention mechanisms, while significantly enhancing global dependency modeling, suffer from high computational complexity and memory consumption, making them less efficient in high-dimensional dynamic scenarios.

To overcome these challenges, this paper introduces MIMTP, a Mamba-based efficient trajectory prediction method, which effectively models dynamic temporal features of vehicle historical trajectories as well as HD map environment features. The Mamba model, grounded in state-space modeling, implicitly captures long-range dependencies through a linear-time Selective Scan mechanism. This design alleviates the high computational and memory costs associated with Transformer-based models, achieving a favorable balance between modeling capability and computational efficiency.

3 Methods

3.1 Problem Definition

In this study, the multi-modal trajectory prediction task in autonomous driving scenarios is formulated as a regression and probabilistic estimation problem based on historical scene information. Specifically, given scene information $\mathbf{X}$ within a historical time horizon T_{his}, including vehicle trajectory information $\mathbf{V}$ and HD map information $\mathbf{M}$, the objective of the model is to learn a mapping function

that projects $\mathbf{X}$ to the predicted trajectories $\hat{\mathbf{Y}}$ over a future horizon T_{pred} along with the corresponding multi-modal probability distribution π_i.

3.2 Framework

To achieve efficient and accurate multi-modal vehicle trajectory prediction, this paper proposes a Mamba-based trajectory prediction method, MIMTP, as illustrated in Fig. 1.

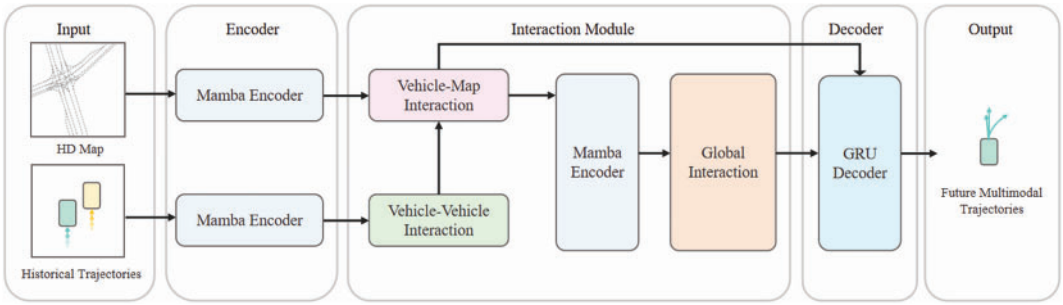

Figure 1: The overall architecture of MIMTP

First, the raw scene data are normalized and embedded. Next, the Mamba Encoder module extracts high-dimensional features from vehicle historical trajectories and map information, generating rich semantic representations. The Vehicle–Vehicle Interaction module further fuses temporal information among agents to produce local inter-vehicle interaction features, while the Vehicle–Map Interaction module models interactions between vehicles and lanes, extracting integrated vehicle–map interaction features. Subsequently, the Mamba Encoder module further enhances the features, and the Global Interaction module consolidates global interaction information among all entities. Finally, the Decoder module takes the outputs from the Vehicle–Map Interaction and Global Interaction modules as input, and generates K multimodal predicted trajectories along with their corresponding probability distributions.

3.3 Mamba Encoder

At the model input stage, the raw vehicle trajectory data and HD map information are first preprocessed and normalized. Through normalization and alignment, heterogeneous inputs from multiple sources are transformed into a unified representation, which facilitates subsequent feature embedding and interaction modeling. After preprocessing, the scene is represented as structured scene information $\mathbf{X}$ that jointly encodes dynamic vehicle states and static map elements, serving as the foundation for downstream spatiotemporal feature extraction and trajectory prediction.

$$\mathbf{X} = \{\mathbf{V}, \mathbf{M}\} \tag{1}$$

$$\mathbf{V} = \left\{\mathbf{v}_{tar}, \mathbf{v}_{sur_1}, \mathbf{v}_{sur_2}, \ldots, \mathbf{v}_{sur_i}\right\} \tag{2}$$

$$\mathbf{M} = \left\{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_j\right\} \tag{3}$$

where $\mathbf{V}$ denotes the set of vehicle-related information in the scene, and $\mathbf{M}$ represents the HD map information. Specifically, $\mathbf{v}_{\text{tar}}$ describes the dynamic state of the target vehicle, while $\mathbf{v}_{\text{sur}_i}$ denotes the dynamic state of the i-th surrounding vehicle. These vehicle states include historical trajectories, velocities, accelerations, and other dynamic attributes observed over the historical time horizon, which together characterize the motion patterns, behavioral tendencies, and interaction cues of each agent.

By incorporating both kinematic and temporal information, the vehicle representation enables the model to capture short-term motion variations as well as longer-term behavioral trends.

Each map element $\mathbf{m}_j$ contains semantic and geometric information of lane segments, including lane vectors, intersection identifiers, turn types, traffic light states, and other relevant map attributes. These map features provide structured priors that describe the road topology, traffic regulations, and navigational constraints of the driving environment. By encoding both the geometric layout and semantic properties of the HD map, the model is able to leverage static contextual information to guide future trajectory prediction, ensuring that the predicted motions remain consistent with road structures and traffic rules.

On the one hand, the vehicle state information $\mathbf{V}$ is processed through an embedding module to project the raw vehicle features into a unified feature space

$$\tilde{\mathbf{V}} = embedding(\mathbf{V}) \tag{4}$$

$$\tilde{\mathbf{V}} = \left\{\tilde{\mathbf{v}}_{tar}, \tilde{\mathbf{v}}_{sur_1}, \tilde{\mathbf{v}}_{sur_2}, ..., \tilde{\mathbf{v}}_{sur_i}\right\} \tag{5}$$

where $\tilde{\mathbf{V}}$ denotes the embedded vehicle information, and $\tilde{\mathbf{v}}_{tar}$ represents the embedded feature of the target vehicle, and $\tilde{\mathbf{v}}_{sur_i}$ denotes the embedded feature of the i-th surrounding vehicle, which are used for subsequent feature extraction and interaction modeling.

Next, the Mamba encoder is applied for feature extraction. In our framework, the Mamba encoder is constructed by stacking multiple Mamba blocks to capture long-range temporal dependencies in vehicle motion sequences. Each Mamba block follows the selective state-space model formulation and consists of a state-space sequence modeling layer together with residual connections and feed-forward transformations. The encoder processes sequential vehicle features and progressively refines temporal representations across multiple layers. In our implementation, the Mamba encoder adopts an embedding dimension of 64, a state dimension of 16, and a convolution kernel size of 4.

As a state-space-model-based sequence modeling architecture with linear computational complexity, Mamba enables efficient modeling of long-range temporal dependencies. In our framework, it is used to extract high-level temporal representations from the embedded vehicle features, which can be formulated as:

$$\mathbf{h}_{tar} = Mamba\left(\tilde{\mathbf{v}}_{tar}\right) \tag{6}$$

$$\mathbf{h}_{sur_i} = Mamba\left(\tilde{\mathbf{v}}_{sur_i}\right) \tag{7}$$

where $\mathbf{h}_{tar}$ and $\mathbf{h}_{sur_i}$ denote the encoded features of the target vehicle and the i-th surrounding vehicle, respectively. On the other hand, the HD map information $\mathbf{M}$ is first transformed into the target coordinate frame and then processed through an embedding module:

$$\tilde{\mathbf{M}} = embedding(\mathbf{M}) \tag{8}$$

$$\tilde{\mathbf{M}} = \left\{\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, ..., \tilde{\mathbf{m}}_j\right\} \tag{9}$$

where $\tilde{\mathbf{M}}$ denotes the embedded map information, and $\tilde{\mathbf{m}}_j$ represents the embedded features of the lane segment indexed by j. Next, the embedded map features are further processed by the Mamba module for sequential feature extraction, enabling effective modeling of structural dependencies along lane geometries and yielding:

$$\mathbf{l}_j = Mamba\left(\tilde{\mathbf{m}}_j\right) \tag{10}$$

where $\mathbf{l}_j$ denotes the encoded map feature.

The Mamba module efficiently and accurately extracts the encoded features of both vehicles and map elements, providing rich spatiotemporal context that captures both temporal dynamics and structural information for the subsequent attention-based aggregation.

3.4 Vehicle-Vehicle Interaction

The Vehicle–Vehicle Interaction module models local interactions on a frame-by-frame basis to effectively capture the dynamic relationships among vehicles. At each time step $t = 1, 2, \ldots, T_{his}$, the vehicle encoding $\mathbf{h}i$ aggregates the encoded representations of both the target vehicle and its surrounding vehicles within the local neighborhood. Specifically, the target vehicle feature is denoted as $\boldsymbol{h}_{tar}^{t}$, while the features of the i-th surrounding vehicle are represented as $\boldsymbol{h}_{sur_i}^{t}$. These encoded features jointly characterize the instantaneous interaction context at time t, providing a compact yet expressive representation of the relative motion and spatial relationships among vehicles.

For each frame, the module employs a cross-attention mechanism to model the interactions between the target vehicle and its surrounding vehicles. Specifically, the encoded feature of the target vehicle is used as the query, while the encoded features of neighboring vehicles are adopted as the keys and values:

$$\boldsymbol{q}_{tar}^{t} = W_q^t \boldsymbol{h}_{\text{tar}}^{t},\ \boldsymbol{k}_i^t = W_k \boldsymbol{h}_{sur_i}^{t},\ \boldsymbol{v}_i^t = W_v \boldsymbol{h}_{sur_i}^{t} \tag{11}$$

Specifically, $\boldsymbol{q}_{tar}^{t}, \boldsymbol{k}_i^t, \boldsymbol{v}_i^t \in \mathbb{R}^{H \times d_k}$, where $\mathrm{W_q}, \mathrm{W_k}, \mathrm{W_v}$ are the linear projection matrices, H denotes the number of attention heads, and d_k is the dimensionality of each head.

For each attention head, the unnormalized attention score $\boldsymbol{a}_{\text{tar,sur}_i}^{t}$ is first computed, and then normalized using the softmax function to obtain the attention weight $\hat{\boldsymbol{a}}_{\text{tar,sur}_i}^{t}$.

$$\boldsymbol{a}_{\text{tar,sur}_i}^{t} = \frac{\boldsymbol{q}_{\text{tar}}^{t} \cdot \boldsymbol{k}_i^t}{\sqrt{d_k}} \in \mathbb{R} \tag{12}$$

$$\hat{\boldsymbol{a}}_{\text{tar,sur}_i}^{t} = \text{softmax}\left(\boldsymbol{a}_{\text{tar,sur}_i}^{t}\right) \tag{13}$$

The attention weight $\hat{\boldsymbol{a}}_{\text{tar,sur}i}^{t}$ is utilized to measure the relative importance of each surrounding vehicle with respect to the target vehicle at time step t. Based on these attention weights, a weighted summation of the value vectors corresponding to the surrounding vehicles is performed, enabling the model to selectively aggregate interaction-relevant information. This process yields an aggregated representation of the target vehicle at time step t, denoted as $\boldsymbol{z}_{\text{tar}}^{t}$, which effectively encodes the local interaction context and captures the influence of neighboring vehicles on the target vehicle's motion.

$$\boldsymbol{z}_{\text{tar}}^{t} = \sum_i \hat{\boldsymbol{a}}_{\text{tar,sur}_i}^{t} \boldsymbol{v}_i^t \tag{14}$$

Subsequently, a residual connection is applied to $\boldsymbol{h}_{tar}^{t}$, where the linearly transformed $\boldsymbol{z}_{tar}^{t}$ is added to $\boldsymbol{h}_{tar}^{t}$ to obtain the updated feature representation:

$$\tilde{\boldsymbol{h}}_{tar}^{t} = \boldsymbol{h}_{tar}^{t} + W_s \boldsymbol{z}_{tar}^{t} \tag{15}$$

Then, the temporally updated features are concatenated along the time dimension to form a sequence of length T_{his}, i.e., $\left[\tilde{\boldsymbol{h}}_{tar}^{1}, \tilde{\boldsymbol{h}}_{tar}^{2}, \ldots, \tilde{\boldsymbol{h}}_{tar}^{T_{\text{his}}}\right]$, where each element in the sequence represents the target vehicle feature at a specific historical time step after temporal updating. This ordered sequence explicitly preserves the temporal dependency structure of the historical observations.

To enable causal temporal modeling and prevent information leakage from future frames, we adopt an upper-triangular masking scheme during the sequence processing. Specifically, this masking strategy ensures that, at each time step, the model can attend only to the current and previous historical frames, while strictly blocking access to any subsequent information. Such a design enforces temporal causality and aligns the modeling process with real-world sequential decision-making scenarios.

The masked sequence is then fed into a multi-layer Transformer Encoder, which leverages self-attention mechanisms to capture long-range temporal dependencies and aggregate contextual information across different time steps. Through this process, interactions among vehicles over time are effectively modeled and integrated, resulting in the vehicle–vehicle interaction features $\boldsymbol{h}_{\mathrm{vv}}$:

$$\boldsymbol{h}_{\mathrm{vv}} = Trans former\ Encoder\left(\left[\tilde{\boldsymbol{h}}_{tar}^{1}, \tilde{\boldsymbol{h}}_{tar}^{2}, ..., \tilde{\boldsymbol{h}}_{tar}^{T_{\mathrm{his}}}\right]; mask\right) \tag{16}$$

Consequently, the resulting vehicle–vehicle interaction features provide informative temporal context for accurately modeling the target vehicle's motion influenced by surrounding traffic participants.

3.5 Vehicle-Map Interaction

In the Vehicle–Map Interaction module, the model takes as input the vehicle–vehicle interaction features $\boldsymbol{h}_{\mathrm{vv}}$ and the map features $\mathbf{l}_j$. This module aims to explicitly model the interactions between the dynamically evolving vehicle context and the static yet semantically rich map information, enabling the model to reason about vehicle behavior under road structure constraints. By incorporating map features, the model can effectively capture how lane geometry, road topology, and other environmental elements influence vehicle motion.

Concretely, a cross-attention mechanism is employed, where the vehicle–vehicle interaction features serve as the query, and the map features are used as the keys and values. Through this design, the model selectively attends to map elements that are most relevant to the current vehicle interaction state. The query, key, and value representations are obtained via linear projections:

$$\boldsymbol{q}_{vv} = W_q \boldsymbol{h}_{\mathrm{vv}},\ \boldsymbol{k}_{\mathrm{l}} = W_k \boldsymbol{l}_j,\ \boldsymbol{v}_{\mathrm{l}} = W_v \boldsymbol{l}_j \tag{17}$$

Specifically, $\boldsymbol{q}_{vv}, \boldsymbol{k}_{\mathrm{l}}, \boldsymbol{v}_{\mathrm{l}} \in \mathbb{R}^{H \times d_k}$, where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are the linear projection matrices. This multi-head formulation allows the model to attend to different aspects of map information in parallel, thereby capturing diverse spatial and semantic relationships between vehicles and map elements.

Subsequently, for the target vehicle and its neighboring map entities, the local attention scores are first computed and then normalized using the softmax function:

$$\mathbf{a}_{\mathrm{vl}} = \frac{\boldsymbol{q}_{vv} \times \boldsymbol{k}_{\mathrm{l}}}{\sqrt{d_k}} \tag{18}$$

$$\hat{\boldsymbol{a}}_{\mathrm{vl}} = \mathrm{softmax}(\mathbf{a}_{\mathrm{vl}}) \tag{19}$$

Here, $\mathbf{a}_{\mathrm{vl}}$ denotes the unnormalized attention scores, and $\hat{\boldsymbol{a}}_{\mathrm{vl}}$ represents the normalized attention weights. $\hat{\boldsymbol{a}}_{\mathrm{vl}}$ is then used to compute the weighted aggregation over the neighboring map features, yielding the aggregated representation $\boldsymbol{z}_{\mathrm{vl}}$:

$$\boldsymbol{z}_{\mathrm{vl}} = \sum_i \hat{\boldsymbol{a}}_{\mathrm{vl},i}\, \boldsymbol{v}_{\mathrm{l},i} \tag{20}$$

Subsequently, a residual connection is applied to $\boldsymbol{h}_{\mathrm{vv}}$, where the linearly transformed $\boldsymbol{z}_{\mathrm{vl}}$ is added to $\boldsymbol{h}_{\mathrm{vv}}$ to obtain the updated feature representation $\boldsymbol{h}_{\mathrm{vl}}$.

$$\boldsymbol{h}_{\mathrm{vl}} = \boldsymbol{h}_{\mathrm{vv}} + W_s \boldsymbol{z}_{\mathrm{vl}} \tag{21}$$

Finally, $\boldsymbol{h}_{\mathrm{vl}}$ is further processed by the Mamba Encoder to enhance the feature representation, yielding the vehicle–map interaction features $\boldsymbol{h}_{final}$:

$$\boldsymbol{h}_{final} = Mamba\,(\boldsymbol{h}_{\mathrm{vl}}) \tag{22}$$

Through the aforementioned process, the model can efficiently integrate the spatiotemporal information of both vehicles and the map, providing rich spatiotemporal context for the subsequent global prediction module.

3.6 Global Interaction

To further model global interactions among all traffic participants, global information aggregation is achieved through a graph attention mechanism. Unlike local interaction modeling, this module focuses on capturing long-range dependencies and holistic relational patterns across the entire scene. In the global interaction module, each node represents an individual agent. Edges are defined between all pairs of nodes, and edge features encode the relative spatial positions along with rotation-invariant angular information.

Specifically, the global interaction module takes $\boldsymbol{h}_{final}$ as the initial node features, where each node represents an individual agent, and performs interaction modeling over the global graph structure at the final time step T_{his} of the historical trajectories. By operating on the global graph, the model is able to explicitly reason about interactions that extend beyond local neighborhoods and incorporate broader contextual information.

At the final historical time step, the spatial configuration of all agents is fixed, enabling the construction of a graph that reflects their relative geometric relationships. First, the relative positions between agents a and b in the graph are computed, which serve as essential inputs for characterizing spatial relationships and guiding the subsequent attention-based message passing process.

$$\Delta \boldsymbol{p}_{ab} = \boldsymbol{p}_a^{T_{\mathrm{his}}} - \boldsymbol{p}_b^{T_{\mathrm{his}}} \tag{23}$$

The relative spatial relationships $\Delta \boldsymbol{pab}$ are then mapped into high-dimensional edge features $\boldsymbol{e}_{ab}$ through an embedding function, which enables the model to capture rich spatial correlations between agents. The cosine and sine values of the relative angular difference are appended to the embedded features, thereby obtaining a rotation-invariant spatial encoding that facilitates consistent interaction modeling under different coordinate frames.

Subsequently, the module performs stacked processing on the nodes through multiple layers of graph attention, allowing information to be progressively propagated and refined across the global interaction graph. Within each graph attention layer, interactions between agents a and b are modeled via a multi-head attention mechanism, where the attention weights are computed based on query–key matching that jointly considers node features and the corresponding edge features. This design enables the model to adaptively weight interactions according to both agent states and their spatial relationships, effectively capturing complex and heterogeneous global interaction patterns.

$$\boldsymbol{Q}_a = \boldsymbol{h}_{final_a} \tag{24}$$

$$\boldsymbol{K}_b = \boldsymbol{h}_{final_b} \tag{25}$$

$$\boldsymbol{K}_{e_{ab}} = \boldsymbol{e}_{ab} \tag{26}$$

$$\boldsymbol{\beta}_{ab} = \text{softmax}\left(\frac{\boldsymbol{Q}_a \cdot \left(\boldsymbol{K}_b + \boldsymbol{K}_{e_{ab}}\right)}{\sqrt{d_k}}\right) \tag{27}$$

Here, $\boldsymbol{h}_{\text{final}_a}$ and $\boldsymbol{h}_{\text{final}_b}$ represent the vehicle and map interaction information for agents a and b, respectively. $\boldsymbol{Q}_a$, $\boldsymbol{K}_b$, $\boldsymbol{K}_{e_{ab}}$ denote the linear projections for the nodes and edges, while $\boldsymbol{\beta}_{ab}$ represents the attention weight of agent a on the information from agent b.

In the node update phase, the node representation is fused with the node value vector $\boldsymbol{V}_b$ and the edge value vector $\boldsymbol{V}_{e_{ab}}$, enabling context modeling based on the global graph structure. The aggregated result $\mathbf{H}_{\text{out}}$ represents the feature representation of the node after integrating the neighborhood context within the global graph structure.

$$\boldsymbol{V}_b = \boldsymbol{h}_{final_b} \tag{28}$$

$$\boldsymbol{V}_{e_{ab}} = \boldsymbol{e}_{ab} \tag{29}$$

$$\boldsymbol{H}_{\text{out}} = \boldsymbol{\beta}_{ab}\left(\boldsymbol{V}_b + \boldsymbol{V}_{e_{ab}}\right) \tag{30}$$

To enhance the model's stability and nonlinear expressive capacity, residual connections, layer normalization, and feed-forward networks are applied between consecutive graph attention layers. The residual connections facilitate effective gradient propagation and preserve essential node-level information across layers, while layer normalization mitigates internal covariate shift and improves training stability. In addition, the feed-forward networks introduce nonlinearity and increase the representational power of the model, enabling it to capture more complex interaction patterns.

After multiple layers of global interaction, each node representation has aggregated information from a broad set of agents through iterative attention-based message passing, resulting in a context-aware encoding that reflects both local and global interaction effects. The resulting node representations are then normalized to ensure consistent feature scaling and are subsequently passed through a linear projection to align them with the multimodal output space, thereby producing a unified representation that is well-suited for downstream trajectory generation and prediction tasks.

$$\boldsymbol{H}_{\text{global}} = \text{Linear}\left(\boldsymbol{H}_{\text{out}}\right) \in \mathbb{R}^{num \times F \times D} \tag{31}$$

Here, *num* denotes the number of nodes, F is the number of prediction modalities, and D represents the embedding dimension. This structure provides globally consistent and interaction-aware high-level semantic information for the subsequent multimodal trajectory decoding.

3.7 GRU Decoder

The Decoder module is designed to fuse local interaction features with global interaction features in order to generate multi-modal future trajectories along with their corresponding probabilities. By jointly leveraging fine-grained local interaction cues and holistic global context, the decoder is able to model diverse future behaviors under complex traffic scenarios.

The inputs to the decoder include the local interaction features $\boldsymbol{h}_{final}$, which encode detailed motion and interaction information of the target vehicle, and the global interaction features $\boldsymbol{H}_{global}$, which provide complementary context derived from global graph-based reasoning. Together, these

representations form a comprehensive feature basis that supports accurate and diverse trajectory prediction in downstream decoding stages.

First, the local interaction features are expanded along the modality dimension and concatenated with the global interaction features. The combined representation is then passed through a multilayer perceptron (MLP) to produce the probability weight for each modality, $\pi_i \in \mathbb{R}^{Num \times F}$.

$$\pi_i = MLP\left([\boldsymbol{h}_{final}, \boldsymbol{H}_{global}]\right) \tag{32}$$

Here, *Num* denotes the number of vehicles in the scene, and F is the number of trajectory modalities.

Subsequently, the Decoder performs temporal modeling to generate future trajectories over the prediction horizon. To this end, the global interaction features $\boldsymbol{H}_{global}$, which encode holistic context from all agents and their spatial relationships, are first rearranged and unfolded along the temporal dimension to form a sequential input suitable for the GRU. This allows the recurrent network to capture temporal dependencies and evolving interaction patterns across time. Simultaneously, the local interaction features $\boldsymbol{h}_{final}$, which encode fine-grained motion cues and immediate neighborhood influences, are used as the initial hidden state of the GRU, providing a strong prior that grounds the prediction in the current vehicle context.

At each time step, the GRU updates its hidden state to integrate both past information and the dynamic influence of global interactions. The updated hidden state is then mapped to the two-dimensional coordinates (x, y) through a two-layer multi-layer perceptron (MLP), producing the predicted position of the target vehicle. In addition, the positional uncertainty, reflecting the inherent stochasticity of future motion, is estimated using the ELU activation function. This uncertainty modeling enables the decoder to capture multi-modal behaviors and provide probabilistic predictions, allowing the framework to reason about diverse and plausible future trajectories under complex traffic scenarios.

Finally, the Decoder outputs the future trajectories over the prediction horizon T_{pred}, denoted as $\hat{\mathrm{Y}} = \left(x, y, \sigma_x^2, \sigma_y^2\right) \in \mathbb{R}^{[F, \mathrm{Num}, T_{\text{pred}}, 4]}$.

3.8 Loss

The final loss of the model consists of two complementary components: a classification loss $\mathcal{L}$cls and a regression loss $\mathcal{L}$reg, which together guide the model to produce accurate and probabilistically meaningful future trajectory predictions:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} \tag{33}$$

The classification loss $\mathcal{L}_{\text{cls}}$ is defined as:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{F} \tau_i^{(m)} \log \pi_i^{(m)} \tag{34}$$

This term penalizes the discrepancy between the predicted probability distribution $\pi_i^{(m)}$ over multiple trajectory modes and the ground-truth distribution $\tau_i^{(m)}$. Here, $\pi_i^{(m)}$ denotes the probability weight of the predicted trajectory indexed by m, while $\tau_i^{(m)}$ represents the true likelihood of each trajectory mode. By minimizing $\mathcal{L}_{\text{cls}}$, the model is encouraged to assign higher confidence to the trajectories that are more likely to match the observed ground-truth, effectively enabling multi-modal prediction.

The regression loss $\mathcal{L}_{\text{reg}}$ is computed using the negative log-likelihood of a Laplace distribution:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N}\left[\log(2\theta_i) + \frac{|v_i - \mu_i|}{\theta_i}\right] \tag{35}$$

where $\mu_i = (x, y)$ denotes the predicted trajectory positions, θ_i is the corresponding uncertainty parameter, and $v_i = (x, y)$ is the ground-truth trajectory point. This loss encourages the predicted trajectory points to be close to the ground-truth while simultaneously modeling the positional uncertainty through θ_i. In other words, the model learns not only the expected position of the vehicle but also how confident it is in that prediction.

By jointly optimizing both the classification and regression losses, the model is able to capture both the multi-modal nature of future trajectories and the associated uncertainties of each prediction. This combination ensures that the model produces predictions that are both accurate and robust, effectively accounting for the inherent stochasticity in vehicle motion and complex traffic interactions.

4 Experiments

4.1 Experiments Setup

Dataset. We evaluate the MIMTP model on the Argoverse 1 Motion Forecasting dataset [27], a large-scale benchmark for vehicle trajectory prediction in urban environments. The dataset includes real-world vehicle trajectories and high-definition (HD) map data, such as lane centerlines, road boundaries, and traffic topology. It contains approximately 320,000 scenarios, with 205,942 for training, 39,472 for validation, and 78,143 for testing. Each sample consists of 20 frames of historical and 30 frames of future trajectories, with the model predicting the future trajectories based on the observed inputs.

Metrics. To comprehensively assess trajectory forecasting performance, we adopt four standard metrics with the number of predicted trajectories set to K = 6: minADE, minFDE, and MR. The minADE metric computes the average Euclidean distance between the predicted trajectory and the ground-truth trajectory over the entire prediction horizon, while minFDE measures the Euclidean distance between the predicted final position and the ground-truth final position. The MR (Miss Rate) metric represents the proportion of samples whose predicted final position deviates from the ground truth by more than 2 m. The DAC (Drivable Area Compliance) metric measures the fraction of predicted trajectories that remain entirely within the drivable area over the prediction horizon. It is computed as the number of trajectories that do not leave the drivable area divided by the total number of predicted trajectories. Together, these metrics provide a reliable and comprehensive evaluation of the model's effectiveness in multimodal trajectory prediction.

Implementation Details. All experiments are conducted on an Ubuntu 24.04 LTS system using an NVIDIA RTX 5090 GPU. The model is trained using the Adam optimizer with an initial learning rate of 5×10^{-4} and a weight decay of 1×10^{-4}. The batch size is set to 32, and the maximum number of training epochs is 60. The embedding dimension is 64, the number of attention heads is 8, and the dropout rate is 0.1. During training, 20 frames of historical trajectories are used as input to predict the next 30 frames, and 6 candidate trajectories are generated for multimodal prediction evaluation.

4.2 Main-Result

The performance of MIMTP on the Argoverse 1 test set is presented in Table 1. For minFDE (1.3215), the standard deviation is approximately 0.026. For minADE (0.8394), it is approximately 0.017, while for MR (0.1568), it is around 0.003.

Table 1: Performance comparison of different models on trajectory prediction metrics

Model	minFDE	minADE	MR
mmTransformer	1.3383	0.8435	0.1542
TNT	1.5384	0.9358	0.1328
LaneGCN	1.3640	0.8679	0.1634
WIMF	1.4220	0.8995	0.1669
Autobot	1.3715	0.8758	0.1635
DenseTNT	1.3814	0.9106	0.1032
LaneRCNN	1.4526	0.9038	0.1232
MIMTP	1.3215	0.8394	0.1568

Note: MR: Miss Rate.

To validate the effectiveness of the proposed MIMTP model in vehicle trajectory prediction, we compare it with several representative baseline models, including mmTransformer [4], TNT [28], LaneGCN [29], WIMF [30], Autobot [31], DenseTNT [32], and LaneRCNN [33].

Comprehensive performance across minFDE, minADE, and MR. Specifically, MIMTP achieves 1.3215 in minFDE and 0.8394 in minADE, both the lowest among all compared methods. Compared with TNT and LaneRCNN, the minFDE is reduced by approximately 14.10% and 9.03%, respectively, while the minADE is reduced by about 10.30% and 7.13%. Furthermore, compared with LaneGCN, MIMTP reduces minFDE and minADE by 3.12% and 3.28%, respectively, indicating consistent improvements in trajectory prediction accuracy. In terms of MR, MIMTP obtains a value of 0.1568, maintaining competitive performance among mainstream methods. Compared with LaneGCN, WIMF, and Autobot, the MR is reduced by 4.04%, 6.05%, and 4.10%, respectively, suggesting that the multimodal trajectory samples predicted by MIMTP exhibit a more reliable distribution.

To further assess the generalization ability of MIMTP, we evaluate it on the Argoverse2 (AV2) dataset. The model achieves a minADE of 0.862, a minFDE of 1.371, and an MR of 0.163 on AV2, indicating that the proposed method generalizes effectively across different datasets.

To further evaluate the prediction capability of the proposed MIMTP model under diverse driving scenarios, four representative cases are selected for visualization analysis, as shown in Fig. 2, including straight driving at T-junction, straight driving at dual-crossroad, left-turn maneuvers at standard intersections and right-turn maneuvers at standard intersections. As shown in the figure, MIMTP demonstrates consistently accurate prediction performance across a wide range of traffic conditions.

To further evaluate the prediction capability of the proposed MIMTP model under diverse driving scenarios, four representative cases are selected for visualization analysis, as shown in Fig. 2, including straight driving at a T-junction, straight driving at a dual-crossroad, left-turn maneuvers at standard intersections, and right-turn maneuvers at standard intersections. As shown in the figure, MIMTP demonstrates consistently accurate prediction performance across a wide range of traffic conditions.

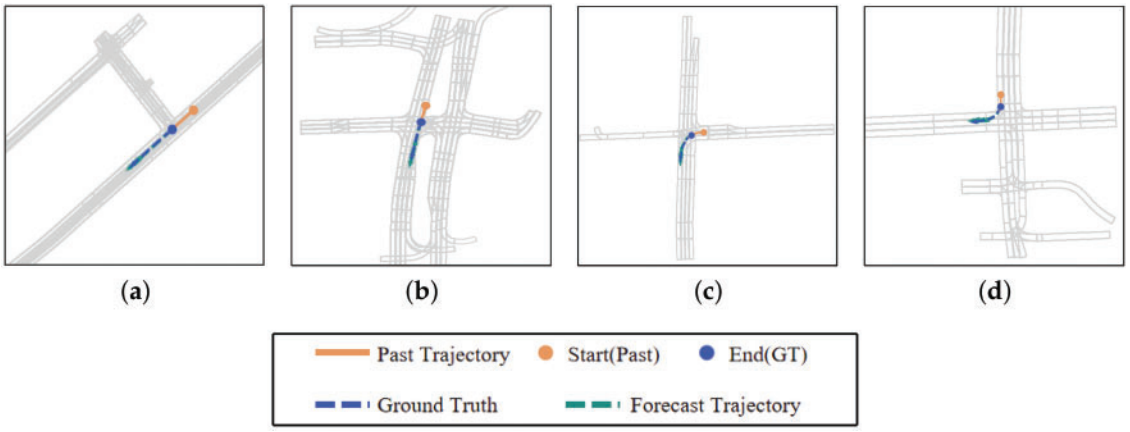

Figure 2: Visualization of predicted trajectories in different driving scenarios. (a) straight driving at a T-junction, (b) straight driving at a dual-crossroad, (c) left-turn maneuvers at standard intersections, (d) right-turn maneuvers at standard intersections

In the T-junction straight-driving scenario, the predicted trajectory almost completely overlaps with the ground-truth trajectory, and the predicted final position closely matches the true endpoint, indicating that the model can precisely capture vehicle motion patterns. The quantitative evaluation for this case shows a minADE of 0.549 m, minFDE of 1.086 m, demonstrating accurate and reliable trajectory prediction.

In the dual-crossroad straight-driving scenario, the predicted trajectory remains highly consistent with the ground truth, with only a slight deviation at the final position, demonstrating the model's robustness in complex multi-intersection environments. For this scenario, MIMTP achieves a minADE of 0.645 m and minFDE of 1.104 m, indicating effective coverage of multimodal possibilities even in more challenging intersections.

For left-turn maneuvers at standard intersections, the predicted path closely follows the ground truth, with minor deviations occurring near the end of the maneuver. Quantitatively, the model attains a minADE of 0.630 m and minFDE of 1.209 m, reflecting its capacity to capture turning dynamics accurately.

Similarly, in the right-turn scenario, MIMTP effectively captures the overall turning behavior, and the predicted curve remains well aligned with the true trajectory. The corresponding metrics are a minADE of 0.645 m and minFDE of 1.300 m confirming the model's reliable performance across both left- and right-turn maneuvers.

Overall, these visualization results indicate that the proposed MIMTP model achieves stable and accurate trajectory prediction across diverse driving scenarios. The model not only captures vehicle motion trends with high precision but also demonstrates strong adaptability to complex intersection environments.

4.3 Ablation Study

To validate the effectiveness of the Mamba Encoder in temporal feature modeling, we conduct an ablation experiment in which the Mamba Encoder in the original model is replaced with a standard Transformer Encoder, while keeping all other network components and training configurations unchanged. The evaluation is performed under the multimodal prediction setting ($K = 6$ and 3),

and we report the minFDE, minADE, and MR metrics, as well as the inference time. Tables 2 and 3 present the results of the ablation study.

Table 2: Performance comparison with $K = 6$ multimodal predictions

Model	minADE ↓	minFDE ↓	MR ↓	DAC ↑	Inference Time (ms) ↓
Transformer Encoder	0.8550	1.3547	0.1609	0.9855	0.5221
Mamba Encoder	0.8394	1.3215	0.1568	0.9903	0.4683

Note: MR: Miss Rate; DAC: Drivable Area Compliance.

Table 3: Performance comparison with $K = 3$ multimodal predictions

Model	minADE ↓	minFDE ↓	MR ↓	DAC ↑	Inference Time (ms) ↓
Transformer Encoder	1.2004	2.4005	0.3407	0.9861	0.5403
Mamba Encoder	1.1794	2.3165	0.3260	0.9983	0.4426

Note: MR: Miss Rate; DAC: Drivable Area Compliance.

The experimental results demonstrate that the Mamba Encoder substantially outperforms the baseline Transformer Encoder in terms of trajectory prediction accuracy and computational efficiency across different multimodal prediction settings.

Specifically, for $K = 6$ multimodal predictions, the Mamba Encoder achieves a reduction in minFDE from 1.3547 to 1.3215, corresponding to a relative improvement of 2.45%. Similarly, minADE decreases from 0.8550 to 0.8394, yielding a gain of 1.82%, while MR is reduced from 0.1609 to 0.1568, representing a 2.55% improvement. In addition, the inference time is shortened from 0.5221 to 0.4683 ms, achieving a 10.3% increase in computational efficiency. These results indicate that the Mamba Encoder effectively enhances the modeling of dynamic vehicle behaviors while maintaining high prediction accuracy for $K = 6$.

For $K = 3$ multimodal predictions, the performance benefits are even more pronounced. The minFDE is reduced from 2.4005 (Transformer Encoder) to 2.3165 (Mamba Encoder), corresponding to a 3.50% improvement. The minADE decreases from 1.2004 to 1.1794, yielding a gain of 1.76%, while MR drops from 0.3407 to 0.3260, reflecting a 4.30% enhancement. Moreover, the inference time decreases from 0.5403 to 0.4426 ms, resulting in an 18.1% improvement in computational efficiency. These findings substantiate that the Mamba Encoder consistently advances both predictive accuracy and efficiency across varying numbers of multimodal predictions, highlighting its capability to capture long-term dependencies and dynamic features effectively.

Furthermore, to evaluate the computational efficiency of the proposed Mamba encoder, we compare its training time and parameter size with those of a Transformer-based encoder under identical experimental settings. As shown in Table 4, the model with mamba encoder requires 21.28 min per epoch, whereas the model with transformer encoder takes 29.65 min. In addition, the Mamba-based model contains fewer parameters (1.973M) compared with the Transformer counterpart (2.529M). These results indicate that the proposed Mamba encoder achieves competitive performance while reducing both training time and model size, demonstrating improved computational efficiency and a smaller memory footprint.

Table 4: Computational efficiency of models with different encoder architectures

Encoder	Training Time (min/epoch)	Parameters (M)
Model with Mamba Encoder	21.28	1.973
Model with Transformer Encoder	29.65	2.529

In our framework, the Vehicle-Vehicle Interaction (VV) module forms the backbone of the model, capturing interactions between vehicles and temporal dynamics from historical trajectories. As such, VV is retained in all ablation configurations. To assess the contributions of additional components, we perform controlled ablation experiments by selectively incorporating the Vehicle-Map Interaction (VM) module and the Global Interaction module on top of the VV baseline. We evaluate four configurations: (1) VV only, (2) VV + VM, (3) VV + Global, and (4) the full model with all modules. Table 5 summarizes the results in terms of minADE, minFDE, and MR, demonstrating how each module contributes to overall prediction performance.

Table 5: Ablation study results with different module combinations

VV	VM	Global	minADE ↓	minFDE ↓	MR ↓
✓			0.9652	1.6283	0.2887
✓	✓		0.8737	1.4361	0.1824
✓		✓	0.9361	1.5883	0.2172
✓	✓	✓	0.8394	1.3215	0.1568

Note: VV: the Vehicle-Vehicle Interaction; VM: the Vehicle-Map Interaction; MR: Miss Rate.

The ablation results in Table 5 highlight the contributions of the VM and Global modules. Compared with the VV-only baseline, incorporating the VM module significantly improves performance, reducing minADE, minFDE, and MR to 0.8737, 1.4361, and 0.1824, respectively. This improvement indicates that vehicle–map interactions provide important structural cues. Adding the Global module also yields consistent gains, suggesting that modeling broader scene-level interactions helps capture additional contextual dependencies. When both modules are combined, the full model achieves the best performance, demonstrating that VM and Global modules provide complementary information for more accurate trajectory prediction.

5 Conclusion

This paper presents an efficient Mamba-based trajectory prediction model, termed MIMTP, which consists of five core modules: a Mamba Encoder, Vehicle–Vehicle Interaction module, Vehicle–Map Interaction module, Global Interaction module, and a Decoder. First, the Mamba Encoder incorporates the Mamba state-space architecture into the feature extraction process, enabling effective modeling of long-range temporal dependencies with reduced computational complexity, while enhancing the integration of global information. Second, the Vehicle–Vehicle Interaction and Vehicle–Map Interaction modules are designed to model dynamic relationships among vehicles and between vehicles and the road environment, respectively. The Global Interaction module further captures scene-level global interaction information. Finally, the Decoder generates multi-modal future trajectory predictions. Experimental results on the Argoverse 1 dataset demonstrate that the proposed model

achieves strong performance, exhibiting clear advantages in key evaluation metrics such as minADE, minFDE, and MR. Moreover, ablation studies indicate that the introduction of the Mamba Encoder not only improves minADE, minFDE, and MR, but also reduces inference time, thereby enhancing overall prediction efficiency. These findings validate the effectiveness of MIMTP in terms of feature extraction efficiency and long-range dependency modeling. Future work will explore multi-agent trajectory prediction to further extend the applicability of the proposed approach in more complex traffic scenarios.

Acknowledgement: The authors would like to thank Shandong University of Science and Technology for their support and valuable resources that contributed to this work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jingen Li and Lukun Wang; Methodology, Jingen Li; Software, Jingen Li; Validation, Jingen Li; Formal analysis, Jingen Li; Investigation, Jingen Li; Resources, Lukun Wang; Data curation, Jingen Li; Writing—original draft preparation, Jingen Li; Writing—review and editing, Jingen Li and Lukun Wang; Visualization, Jingen Li; Supervision, Lukun Wang; Project administration, Lukun Wang; Funding acquisition, Lukun Wang, Jiaming Pei. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in the Argoverse dataset at https://www.argoverse.org/.

Ethics Approval: Not applicable.

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

References

1. Wang T, Xiao D, Xu X, Yuan Q. A survey on vehicle trajectory prediction procedures for intelligent driving. Sensors. 2025;25(16):5129. doi:10.3390/s25165129.
2. Liu J, Mao X, Fang Y, Zhu D, Meng MQ-H. A survey on deep-learning approaches for vehicle trajectory prediction in autonomous driving. In: Proceedings of the 2021 IEEE International Conference on Robotics and Biomimetics; 2021 Dec 27–31; Sanya, China. p. 978–85.
3. Jo E, Sunwoo M, Lee M. Vehicle trajectory prediction using hierarchical graph neural network for considering interaction among multimodal maneuvers. Sensors. 2021;21(16):5354. doi:10.3390/s21165354.
4. Liu Y, Zhang J, Fang L, Jiang Q, Zhou B. Multimodal motion prediction with stacked transformers. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 7577–86.
5. Ngiam J, Caine B, Vasudevan V, Zhang Z, Chiang H-TL, Ling J, et al. Scene transformer: a unified architecture for predicting future trajectories of multiple agents. In: Proceedings of the 2022 International Conference on Learning Representations (ICLR); 2022 Apr 25; Virtual.
6. Han J, Deng X, Cai X, Yang Z, Xu H, Xu C, et al. Laneformer: object-aware row-column transformers for lane detection. Proc AAAI Conf Artif Intell (AAAI). 2022;36(1):799–807.
7. Wang L, Xu X, Pei J. Communication-efficient federated learning via dynamic sparsity: an adaptive pruning ratio based on weight importance. IEEE Trans Cogn Commun Netw. 2025;12:1068–77.

8. Chen G, Wang L, Alam M, Elhoseny M. Intelligent group prediction algorithm of GPS trajectory based on vehicle communication. IEEE Trans Intell Transp Syst. 2020;22(7):3987–96. doi:10.1109/tits.2020.3001188.
9. Pei J, Li J, Song Z, Al Dabel MM, Alenazi MJ, Zhang S, et al. Neuro-vae-symbolic dynamic traffic management. IEEE Trans Intell Transp Syst. 2025. doi:10.1109/tits.2025.3571210.
10. Luo W, Yang B, Urtasun R. Fast and furious: real-time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2018 Jun 18–23; Salt Lake City, UT, USA. p. 3569–77.
11. Casas S, Luo W, Urtasun R. IntentNet: learning to predict intention from raw sensor data. In: Proceedings of the Second Conference on Robot Learning (CoRL); 2018 Oct 29–31; Zurich, Switzerland. p. 947–56.
12. Nikhil N, Tran MB. Convolutional neural network for trajectory prediction. In: Leal-Taixé L, Roth S, editors. Computer Vision–ECCV 2018 Workshops. Cham, Switzerland: Springer; 2018.
13. Phan-Minh T, Grigore EC, Boulton FA, Beijbom O, Wolff EM. CoverNet: multimodal behavior prediction using trajectory sets. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 14074–83.
14. Strohbeck J, Belagiannis V, Müller J, Schreiber M, Herrmann M, Wolf D, et al. Multiple trajectory prediction with deep temporal and spatial convolutional neural networks. In: Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2020 Oct 24–2021 Jan 24; Las Vegas, NV, USA. p. 1992–8.
15. Chou F-C, Lin T-H, Cui H, Radosavljevic V, Nguyen T, Huang T-K, et al. Predicting motion of vulnerable road users using high-definition maps and efficient convnets. In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV); 2020 Oct 19–13; Las Vegas, NV, USA. p. 1655–62.
16. Marchetti F, Becattini F, Seidenari L, Bimbo AD. Mantra: memory augmented networks for multiple trajectory prediction. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 7143–52.
17. Ye M, Cao T, Chen Q. TPCN: temporal point cloud networks for motion forecasting. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25. Nashville, TN, USA. p. 11318–27.
18. Xin L, Wang P, Chan C-Y, Chen J, Li SE, Cheng B. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks. In: Proceedings of the 2018 IEEE International Conference on Intelligent Transportation Systems (ITSC); 2018 Nov 4–7; Maui, HI, USA. p. 1441–6.
19. Ding W, Shen S. Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning. In: Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA); 2019 May 20–24; Montreal, QC, Canada. p. 9610–6.
20. Zyner A, Worall S, Nebot E. Naturalistic driver intention and path prediction using recurrent neural networks. IEEE Trans Intell Transp Syst. 2019;21(4):1584–94. doi:10.1109/tits.2019.2913166.
21. Xing Y, Lv C, Cao D. Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles. IEEE Trans Veh Technol. 2019;69(2):1341–52. doi:10.1109/tvt.2019.2960110.
22. Kim H, Kim D, Kim G, Cho J, Huh K. Multi-head attention based probabilistic vehicle trajectory prediction. In: Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV); 2020 Oct 19–Nov 13; Las Vegas, NV, USA. p. 1720–5.
23. Wang J, Liu K, Li H. LSTM-based graph attention network for vehicle trajectory prediction. Comput Netw. 2024;248(3):110477. doi:10.1016/j.comnet.2024.110477.
24. Meng Q, Guo H, Liu Y, Chen H, Cao D. Trajectory prediction for automated vehicles on roads with lanes partially covered by ice or snow. IEEE Trans Veh Technol. 2023;72(6):6972–86. doi:10.1109/tvt.2023.3236947.
25. Tang X, Kan M, Shan S, Ji Z, Bai J, Chen X. HPNet: dynamic trajectory forecasting with historical prediction attention. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 15261–70.

26. Li J, Yang L, Chen Y, Jin Y. MFAN: mixing feature attention network for trajectory prediction. Pattern Recognit. 2024;25(8):8780–92.
27. Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, Hartnett A, et al. Argoverse: 3D tracking and forecasting with rich maps. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA. p. 8740–9.
28. Zhao H, Gao J, Lan T, Sun C, Sapp B, Varadarajan B, et al. TNT: target-driven trajectory prediction. In: Proceedings of the 4th Conference on Robot Learning (CoRL); 2020 Nov 16–18; Virtual. p. 895–904.
29. Liang M, Yang B, Hu R, Chen Y, Liao R, Feng S, et al. Learning lane graph representations for motion forecasting. In: Computer Vision–ECCV 2020. Cham, Switzerland: Springer International Publishing; 2020. p. 541–56.
30. Khandelwal S, Qi W, Singh J, Hartnett A, Ramanan D. What-if motion prediction for autonomous driving. arXiv:2008.10587. 2020.
31. Girgis R, Golemo F, Codevilla F, Weiss M, D'Souza JA, Ebrahimi Kahou S, et al. Latent variable sequential set transformers for joint multi-agent motion prediction (AutoBots). arXiv:2104.00563. 2021.
32. Gu J, Sun C, Zhao H. DenseTNT: end-to-end trajectory prediction from dense goal sets. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 15303–12.
33. Zeng W, Liang M, Liao R, Urtasun R. LaneRCNN: distributed representations for graph-centric motion forecasting. In: Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2021 Sep 27–Oct 1; Prague, Czech Republic. p. 532–9.