MaTVT: A Transformer-Based Approach for Multi-Agent Prediction in Complex Traffic Scenarios

Anran Li, Hongsheng Yu, Yuyan Pan, Zhenlin Xu, Huibo Bi, *Member, IEEE*, Bolin Gao, Keqiang Li, and Yanyan Chen

***Abstract*—The future trajectories of surrounding agents are critical for the motion planning and control of autonomous vehicles. Thus, this study employs Transformer to develop a multi-agent trajectory prediction model named Multi-agent Trajectory Vector Transformer (MaTVT). MaTVT features a lightweight architecture, comprising a dual-level encoder formed by a low-level encoder and a high-level encoder, along with a multi-modal decoder. Once input enters MaTVT, the low-level encoder first constructs polar coordinate systems centered on target agents and then projects historical trajectories and map elements to each agent-centered coordinate system. Next, it utilizes attention mechanisms to encode motion features, agent-agent interactions, and agent-infrastructure constraints independently and fuses them into the agent encoding sequence. Considering the agent response delay, the low-level encoder extracts heterogeneous spatial-temporal features from agent encoding sequences as the local encodings for target agents. Afterward, the high-level encoder treats all agents as the nodes in a directed graph and utilizes a Graph Attention Network to convert inter-agent relationships into global encodings, which are fused with the local encodings of target agents. Finally, the multi-modal decoder translates these fusion encodings into multi-modal trajectory predictions for target agents. This study selects complex traffic scenarios from the Argoverse Motion Forecasting dataset to create a dedicated dataset for MaTVT training, validation, and testing. The test results demonstrate that MaTVT outperforms advanced benchmark methods in prediction performance, revealing its superb accuracy, efficiency, and robustness. In addition, ablation studies further explain the interpretability of the main functional components of MaTVT and their contributions to prediction performance.**

***Index Terms*—Attention mechanism, complex traffic scenario, multi-agent trajectory prediction, transformer.**

I. Introduction

SAFETY is the paramount concern in vehicle maneuvering. Therefore, autonomous vehicle (AV) motion planning and control must thoroughly account for the restrictions imposed by environmental obstacles and road characteristics [1], [2], particularly in complex traffic scenarios. This study defines complex traffic scenarios as those that meet one or more of the following conditions: lack of clear traffic regulations [3], dense coexistence of road users [4], or strict restrictions from the road environment [5]. In complex traffic scenarios, AVs require an accurate and rapid prediction of future motions of surrounding agents to support motion planning and control, ultimately avoiding potential conflicts. However, the prevalent agent-agent interactions and agent-infrastructure constraints in complex traffic scenarios significantly increase the uncertainty of vehicle motion, requiring AVs to comprehensively assess multiple possible future trajectories of surrounding agents. To this end, this study proposes a multi-agent trajectory prediction model that can simultaneously generate the trajectory predictions of multiple target agents in complex traffic scenarios.

Early research commonly employs physics models or machine learning algorithms to achieve trajectory predictions [6]. However, these approaches cannot adequately model complicated agent-agent interactions and agent-infrastructure constraints, limiting their prediction accuracy in complex traffic scenarios. With the advancement of deep neural networks, deep learning algorithms are increasingly adopted for trajectory prediction due to their flexible architectures and powerful nonlinear fitting capabilities [7]. Inspired by computer vision, some studies employ Convolutional Neural Networks (CNNs) to extract motion features from video frame-like maps and construct appropriate kernels to comprehend agent-agent interactions and agent-infrastructure constraints for multi-agent trajectory prediction [8], [9], but their extensive computation demands and stringent prerequisites for perception accuracy hinder widespread application. Recent studies focus on representing agent motion features, agent-agent interactions, and

Received 30 November 2024; revised 8 June 2025 and 21 August 2025; accepted 23 September 2025. Date of publication 26 September 2025; date of current version 6 March 2026. This work was supported by the National Key Research and Development Program of China under Grant 2021YFB2501000. The review of this article was coordinated by Prof. Luca D'Acierno. *(Corresponding author: Yanyan Chen.)*

Anran Li, Huibo Bi, and Yanyan Chen are with the College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China (e-mail: lianran@emails.bjut.edu.cn; huibobi@bjut.edu.cn; cdyan@bjut.edu.cn).

Hongsheng Yu is with the Institute of Electronic Computing Technology, China Academy of Railway Sciences Group Company, Ltd., Beijing 100081, China (e-mail: a1015940216@163.com).

Yuyan Pan is with the Department of Civil Environmental Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: yypan@psu.edu).

Zhenlin Xu is with the Department of Transportant Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology, 2826 CN Delft, The Netherlands (e-mail: g.xu-2@tudelft.nl).

Bolin Gao and Keqiang Li are with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: gaobolin@tsinghua.edu.cn; likq@mail.tsinghua.edu.cn).

Digital Object Identifier 10.1109/TVT.2025.3614859

agent-infrastructure constraints as vectors and employ Recurrent Neural Networks (RNNs), Graph Neural Networks (GNNs), or attention mechanisms to perform trajectory prediction [10], [11], [12], [13], [14], [15], [16], [17]. These methods not only unify the vectorization of multi-source traffic data to facilitate efficient feature fusion but also effectively capture spatial-temporal correlations and dependencies in dynamic systems to enhance prediction accuracy. This capability enables them to adapt to the requirements of various trajectory prediction tasks.

As a core trajectory prediction task, multi-agent trajectory prediction requires thoroughly leveraging input information to simultaneously generate possible position or state sequences for multiple target objects. To ensure accuracy and efficiency, multi-agent trajectory prediction typically employs specific paradigms to model both independent feature representations of each target agent and social relationships among agents and utilizes them to generate trajectory predictions for target objects in parallel. Compared to single-agent trajectory prediction, multi-agent trajectory prediction can explicitly model agent-agent interactions and agent-infrastructure constraints, enabling efficient and accurate inference of future trajectories for multiple target agents. However, it also faces challenges in computational scalability and efficient modeling due to the combinatorial complexity of multi-agent relationships and the multi-modal nature of agent motion. To this end, this study adopts the Transformer to develop a lightweight trajectory prediction model named Multi-agent Trajectory Vector Transformer (MaTVT) due to the Transformer's exceptional capability in contextual information fusion and parallel computing [18]. Compared to previous methods, MaTVT pioneers a heterogeneous spatial-temporal feature extractor to compensate agent response delays and a hierarchical fusion mechanism integrating local agent-centric encodings with global sociograms, significantly improving prediction accuracy and model interpretability in complex traffic scenarios. As depicted in Fig. 1, MaTVT comprises a dual-level encoder formed by a low-level encoder and a high-level encoder, along with a multi-modal decoder. Once input enters MaTVT, the low-level encoder first partitions traffic scenarios into multiple subregions centered on target agents and then projects historical trajectories and map waypoints into these agent-centered coordinate systems [19]. Next, the low-level encoder employs Fourier transforms and feedforward neural networks (FNNs) to enhance motion features, agent-agent interactions, and agent-map constraints, which are fused into agent encoding sequences through attention computation. It is noteworthy that agents exhibit response delays in decision making in dynamic traffic scenarios [20]. Therefore, the low-level encoder extracts heterogeneous spatial-temporal features from agent encodings at recent time steps as local encodings for target agents. Afterward, the high-level encoder treats all agents as nodes in a directed graph and employs a Graph Attention Network (GAT) to generate global encodings that capture inter-agent relationships, which are fused with local encodings of target agents. Finally, the multi-modal decoder independently translates fusion encodings to generate multi-modal trajectory predictions for multiple target agents. This study selects complex traffic scenarios from the Argoverse Motion Forecasting dataset to create a dedicated dataset for MaTVT training, validation, and testing [21]. The test results demonstrate that MaTVT achieves comparable prediction accuracy and computational efficiency compared to advanced benchmark methods. Furthermore, ablation, parameter, and sensitivity studies on MaTVT further demonstrate its excellent interpretability, adaptability, and robustness. The contributions of this study are summarized as follows:

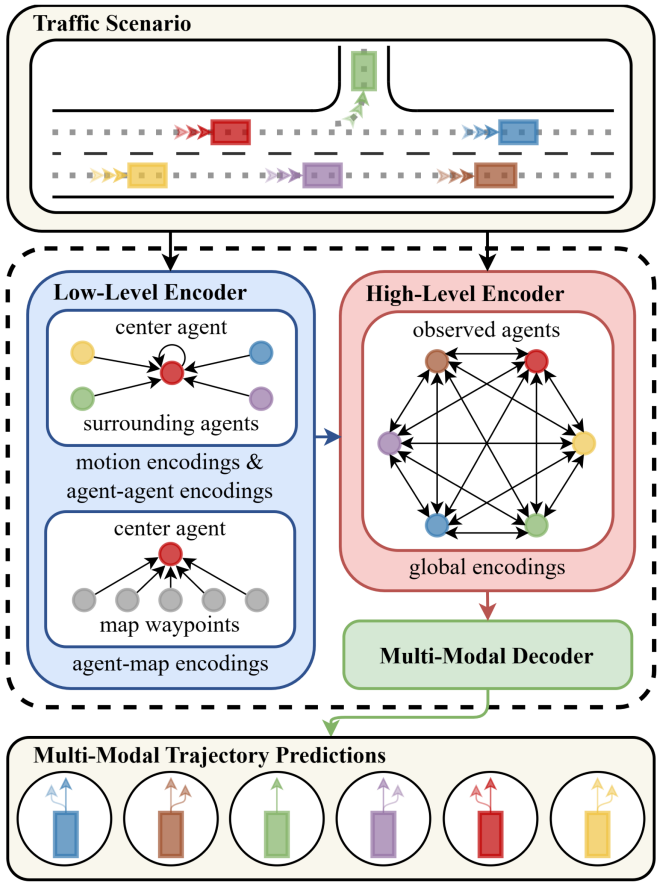

Fig. 1. MaTVT employs the Transformer architecture to extract local and global features from historical data and map information and fuses them to conduct multi-agent trajectory prediction.

- This study develops a Transformer-based model that utilizes historical trajectory data and waypoint-based maps to perform multi-agent trajectory prediction. Performance comparison reveals its superb accuracy and adaptability.
- This study employs attention mechanisms to efficiently encode agent-agent interactions and agent-infrastructure constraints. Ablation studies demonstrate their significant contribution to improving predictive accuracy.
- This study addresses agent response delay and extracts heterogeneous spatial-temporal features as local encodings for target agents. Ablation studies indicate they enhance both predictive accuracy and model interpretability.

The remainder of this study is organized as follows: Section II summarizes related studies. Section III identifies the problem formulation. Section IV introduces the model architecture. Section V presents experimental results and analysis. Section VI reviews this study and outlines future research.

II. Literature Review

Existing research on trajectory prediction can be categorized into physics-based, machine learning-based, deep learning-based, and reinforcement learning-based approaches [6]. This section first provides an overview of each category and then focuses on Transformer-based trajectory prediction methods.

A. Prediction Methods

Physics-based methods construct dynamic or kinematic models that represent real vehicles, integrating their kinematic properties, control inputs, and traffic factors to formulate physical equations to generate trajectory predictions [22], [23], [24]. On this basis, Ammoun and Nashashibi posited that uncertainties in trajectory sequences follow a Gaussian distribution and employed a Kalman linear filter to iteratively generate trajectory predictions [25]. Moreover, Polychronopoulos et al. fitted different Gaussian distributions to govern trajectory sequences in various traffic scenarios and employed a switching Kalman filter to perform trajectory predictions with enhanced accuracy [26]. Besides, Broadhurst et al. employed the Monte Carlo method to generate trajectory prediction sequences with equal weights and selected the sequence most compatible with the static and dynamic constraints of the traffic scenario as the final prediction result [27]. Wang et al. also employed the Monte Carlo method to predict multiple high-probability potential trajectories of target vehicles to enable predictive vehicle motion planning [28]. Although physics-based methods have high computational efficiency and strong interpretability, they struggle to utilize complex traffic factors to enhance prediction accuracy, resulting in significant performance degradation for long-term predictions.

Compared to physics-based methods, machine learning-based methods do not explicitly construct predictive models, but instead formulate trajectory prediction as a regression problem and employ artificial intelligence algorithms to mine evolution patterns from historical data [29]. For example, Joseph et al. posited that distinct vehicle driving behaviors are stochastic functions of Gaussian processes (GP), and used historical trajectory data to fit probability distributions of different driving behavior modes for trajectory prediction [30]. Similarly, Tran and Firl also constructed GP expressions representing different vehicle behavior patterns to achieve accurate trajectory predictions [31]. Moreover, Aoude et al. trained Rapidly-Exploring Random Trees capable of stochastic sampling in the vehicle state space and generated trajectory predictions by fitting the sampling results with vehicle behavior pattern deviations [32]. Kumar et al. further integrated Support Vector Machines with Bayesian filters, which first estimated the most probable future behavior modes of vehicles and then generated the final trajectory predictions under those behavior modes [33]. Besides, Berndt et al. constructed a Hidden Markov Model (HMM) using vehicle positions and front wheel steering angles to define state spaces to predict driving maneuvers [34]. Wang et al. also employed HMM to forecast driving intentions and trajectories of surrounding vehicles to provide reference information for subsequent driving decision-making [35]. Although machine learning-based methods surpass physics-based approaches in long-term trajectory prediction, they struggle to comprehend complex agent-agent and agent-infrastructure interactions, rendering them inadequate for multi-agent trajectory prediction requirements.

Compared to conventional machine learning-based methods, deep learning-based approaches display superior prediction performance due to their exceptional nonlinear fitting capabilities and flexible architectures. In early research, Zyner et al. utilized historical trajectories, azimuth angles, and point velocities as inputs and employed RNN to generate trajectory predictions [10]. Xing et al. used the Gaussian Mixture Model to differentiate driving styles and employed Long Short-Term Memory (LSTM) networks to generate diverse trajectory predictions [11]. Phan-Minh et al. categorized the trajectory data into typical behavior groups and employed CNN for classification-based trajectory prediction [12]. Moreover, Chandra et al. developed an LSTM-CNN hybrid network to leverage agent-agent interactions to improve trajectory prediction accuracy [13]. In recent research, Liang et al. utilized Graph Convolutional Networks (GCN) to capture agent-agent and agent-map interactions from historical trajectories and map elements for trajectory prediction [14]. Gu et al. also employed GCN to model agent-agent interactions for trajectory prediction [15]. Giuliari et al. employed the Transformer architecture to capture spatial-temporal correlations and dependencies from historical trajectories to make predictions [16]. Huang et al. modeled agent-agent relationships as social graphs and employed the Transformer architecture to achieve trajectory prediction in multi-agent scenarios [17]. Currently, deep learning-based methods have become the dominant methods for trajectory prediction due to their powerful feature extraction and nonlinear fitting capabilities, yet they often suffer from a lack of interpretability.

Furthermore, some approaches formulate trajectory generation as Markov Decision Processes and employ Reinforcement Learning algorithms to perform trajectory prediction. For instance, Sun et al. defined vehicle trajectories as a finite set of driving decisions and leveraged Inverse Reinforcement Learning (IRL) to continuously determine subsequent driving decisions and iteratively generate future trajectories [36]. Xu et al. also utilized IRL to simulate human driver behavior to generate trajectory predictions [37]. Moreover, Kuefler et al. utilized Generative Adversarial Imitation Learning (GAIL) to train human driver models using real trajectory data, thereby simulating future trajectories of real-world vehicles [38]. Choi et al. also developed a GAIL-based prediction model that was derived from real trajectory data to generate analogous trajectory predictions [39]. Although reinforcement learning-based methods are commonly leveraged in motion controllers, they require substantial computational resources and extended training periods.

Multi-agent trajectory prediction cannot be simply extended from single-agent prediction methods, as the prediction time delay increases linearly with the number of target objects. Compared to single-agent approaches, multi-agent trajectory prediction emphasizes holistic modeling and continual utilization of social relationships among agents, which not only enhances prediction accuracy but also reduces the computational cost

associated with repeated modeling efforts. Although this enables accurate and game-theoretically logical trajectory prediction in complex traffic scenarios, it also introduces serious challenges in interaction modeling, model interpretability, and multi-agent uncertainty quantification. Traditional single-agent methods (e.g., physics-based or classical machine learning-based methods) fail to capture these dynamic interactions, whereas deep learning algorithms excel due to their inherent capacity for powerful feature extraction and relational modeling. Among deep learning algorithms, the Transformer is optimal for this task, using the attention mechanism to efficiently capture complicated spatial-temporal dependencies within vectorized motions, agent interactions, and environmental constraints. Therefore, subsequent sections of this study focus on Transformer-based prediction methods, systematically analyzing their architectural innovations and scenario-specific adaptations.

B. Prediction and Transformer

The Transformer architecture utilizes attention mechanisms to accurately capture correlations and dependencies from inputs [18]. Compared to classical neural networks, Transformer resolves long-range dependency issues and demonstrates superb ability to handle contextual information and perform parallel computing, which enables its increasingly widespread adoption in trajectory prediction.

In recent studies, Transformer-based methods typically focus on effectively representing agent motions along with complex agent-agent interactions and environmental constraints. For example, Liu et al. utilized three stacked encoder blocks to independently encode agent trajectories, map elements, and agent-agent interactions, and then fused and decoded the encoded sequences to generate multi-agent trajectory predictions [40]. Similarly, Huang et al. employed two separate encoders to independently capture agent-agent interactions and environmental constraints, embedding them into motion representations to enhance predictive accuracy [17]. Moreover, Liu et al. developed a multi-stage Transformer architecture whose encoder employed cross-modal and aggregation-stage encoding layers to capture agent-agent and agent-map interactions to improve trajectory prediction [41]. In addition, Li and He inserted a parallel interaction extraction block into the encoder-decoder architecture to capture social relations between different agents [42]. Yuan et al. proposed an agent-aware attention network that simultaneously captured agent-agent interactions in time steps to facilitate the extraction of spatial-temporal features from inputs [43]. In addition, some Transformer-based methods capture agent-agent and agent-map interactions and fuse them with agent motion representations before the encoder-decoder structure, rather than within it. For example, Zhao et al. proposed a spatial-channel Transformer network that used a channel-wise attention block to extract agent-agent interactions and integrated them with the extracted motion features before feeding them into the encoder-decoder architecture [44]. Zhang et al. employed CNNs to capture motion representations and used GAT blocks to model agent-agent and agent-map interactions, subsequently integrating them into the encoder-decoder structure for trajectory prediction [45]. Furthermore, Nayakanti et al. constructed a scene encoder to independently encode motion features, map features, agent-agent interactions, and traffic light states, subsequently fusing and processing them within an encoder-decoder architecture to perform multi-modal trajectory predictions [46]. They also discussed the influence of different scene feature fusion strategies on trajectory prediction and indicated that models performing scene feature fusion before the encoder-decoder structure achieved the highest accuracy.

These Transformer-based methods typically comprise three key modules: the feature encoding module, the feature fusion module, and the feature decoding module. Among these, the feature encoding module extracts distinct feature encoding sequences from multi-source data, which are then fused by the feature fusion module and translated by the feature decoding module to generate trajectory predictions. Based on the classical framework of Transformer-based methods, this study needs to develop a multi-agent trajectory prediction model that features a lightweight architecture and high computational efficiency to meet real-time requirements for trajectory prediction. In addition, the proposed trajectory prediction model should demonstrate strong interpretability to reveal the mechanisms underlying its trajectory prediction process.

III. Problem Formulation

Multi-modal trajectory prediction refers to leveraging multiple data modalities (e.g., historical trajectories, road structure data, semantic information) to generate possible future coordinate or state sequences for multiple target agents. This study defines the input sequence as $X = \{X_1, X_2, \ldots, X_{T_h}\}$, where $X_t \in X$ contains the position coordinates of target agents and map elements at time step t, and T_h represents historical time steps. Following existing studies [19], [47], x_t is defined as the position coordinate set $p_t = \{p_t^1, p_t^2, \ldots, p_t^{N_{obs}}\}$ and the map element set $m = \{m_1, m_2, \ldots, m_{N_{ls}}\}$, where N_{obs} represents the number of observed agents, N_{ls} represents the number of lane segments, $p_t^i \in \mathbb{R}^2$ represents the position coordinate of agent i at time step t, and $m_\xi \in M$ contains the starting node $p_\xi^0 \in \mathbb{R}^2$, the ending node $p_\xi^1 \in \mathbb{R}^2$, and the semantic information $s_\xi \in \mathbb{R}^2$ of lane segment ξ.

This study aims to generate trajectory predictions for target agents. Due to uncertainties in future trajectories, the trajectory prediction for target agent i contains multiple possible trajectories $p^i \in \mathbb{R}^{N_k \times T_f \times 2}$ with corresponding probability distributions $P^i \in \mathbb{R}^{N_k}$, where N_k represents diverse modes and T_f represents future time steps. To meet the computational complexity and real-time requirements of trajectory prediction, this study sets a maximum limit N_{tar} on the number of target agents that can be simultaneously predicted in a single task execution.

IV. Prediction Model

This section introduces the MaTVT architecture. It begins with an overview of MaTVT, followed by detailed explanations of the low-level encoder, high-level encoder, and multi-modal decoder. In addition, the MaTVT training objective is also discussed in this section.

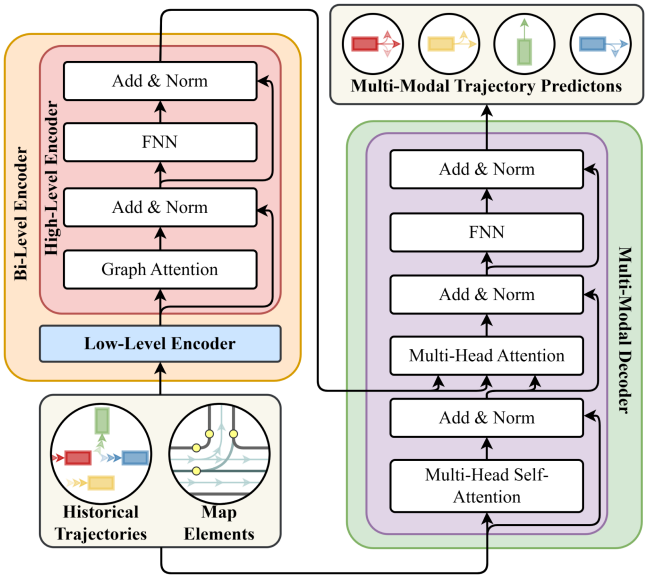

Fig. 2. The MaTVT architecture with its key modules: a dual-level encoder formed by low-level and high-level encoders, along with a multi-modal decoder.

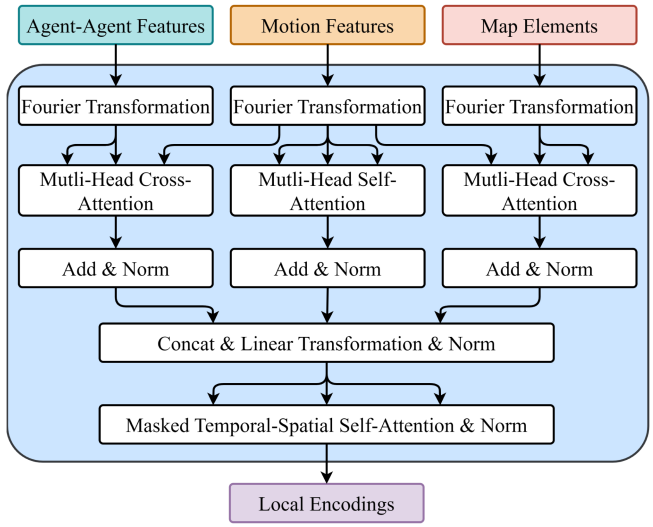

Fig. 3. The low-level encoder extracts motion, agent-agent, and agent-map features and integrates them into the encoding sequences for target agents.

A. Model Overview

Fig. 2 illustrates MaTVT with its core modules: the dual-level encoder consisting of low-level and high-level encoders, and a multi-modal decoder. Once input enters MaTVT, the low-level encoder first constructs independent polar coordinate systems centered on target agents. It then maps the historical trajectories and waypoints into each agent-centric coordinate system and employs FNNs combined with Fourier transform to encode motion features, agent-agent interactions, and agent-map constraints independently. Next, it uses attention mechanisms to fuse motion, agent-agent, and agent-map encodings into agent encoding sequences and extracts heterogeneous spatial-temporal features as local encodings for target agents. Afterward, the high-level encoder treats observed agents as nodes in a directed graph and utilizes GATs to convert inter-agent relationships into global encodings, which are fused with local encodings of target agents. Finally, the multi-modal decoder independently translates these fusion encodings to generate multi-modal trajectory predictions for target agents. The following text provides a detailed introduction to the key function modules in MaTVT.

B. Model Architecture

1) Low-Level Encoder: Fig. 3 illustrates the rotation-invariant encoding paradigm adopted by the low-level encoder [48], [49], [50]. Once X enters MaTVT, the low-level encoder constructs independent polar coordinate systems centered on target agents at time step T_h. For example, the polar coordinate system ϕ_i for agent i is defined with its final position $p^i_{T_h}$ as the origin and the final motion vector $p^i_{T_h} - p^i_{T_h-1}$ as the polar axis. Subsequently, the historical trajectory coordinates of the target agent i and its surrounding agents are assigned to ϕ_i and converted into Fourier features $\hat{p}^i_t \in \mathbb{R}^2$ and $\hat{p}^j_t \in \mathbb{R}^2$ to enhance high-frequency features [51], [52]. Next, the low-level encoder employs FNNs with Rectified Linear Units (ReLU) to extract the motion feature $z^i_t \in \mathbb{R}^{d_h}$ and agent-agent feature $z^j_t \in \mathbb{R}^{d_h}$ from $\hat{p}^i_t$ and $\hat{p}^j_t$:

$$z^i_t = \text{ReLU}\left(W_c(\hat{p}^i_t - \hat{p}^i_{t-1}) + b_c\right), \tag{1}$$

$$z^{i,j}_t = \text{ReLU}\left(W_s\left[(\hat{p}^j_t - \hat{p}^j_{t-1}), (\hat{p}^j_t - \hat{p}^i_t)\right] + b_s\right), \tag{2}$$

where d_h represents the hidden dimension, and $W_c \in \mathbb{R}^{d_h \times 2}$, $W_s \in \mathbb{R}^{d_h \times 2}$, $b_c \in \mathbb{R}^{d_h}$, and $b_s \in \mathbb{R}^{d_h}$ are learnable matrices. Afterward, it transforms z^i_t into $q^i_t \in \mathbb{R}^{N_h \times d_k}$, and $z^{i,j}_t$ into the key vector $k^{i,j}_t \in \mathbb{R}^{N_h \times d_k}$ and the value vector $v^{i,j}_t \in \mathbb{R}^{N_h \times d_k}$, subsequently performing multi-head attention computation:

$$u^{i,j}_t = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_{N_h})W^O, \tag{3}$$

$$\text{head}_h = \text{Softmax}\left(\frac{q^{i,j}_t (k^{i,j}_t)^\top}{\sqrt{d_k}}\right) v^{i,j}_t, \tag{4}$$

where $u^{i,j}_t \in \mathbb{R}^{d_h}$ represents the agent-agent interaction between the target agent i and the surrounding agent j at time step t, N_h represents the number of heads, d_k represents the head dimension, and $W^O \in \mathbb{R}^{d_h \times d_h}$ is a learnable matrix. Finally, the low-level encoder integrates the agent-agent interactions between the target agent i and all surrounding agents at time step t:

$$u^i_t = \sum_{j=1}^{N^i_t} W_u u^{i,j}_t, \tag{5}$$

where $u^i_t \in \mathbb{R}^{d_h}$ represents the agent-agent encoding of the target agent i at time step t, N^i_t represents the number of agents surrounding agent i, and $W_u \in \mathbb{R}^{d_h \times d_h}$ is a learnable matrix. To ensure efficient computation, N^i_t considers only surrounding agents whose distance to the target agent i is less than the

threshold δ_1. In this way, the agent-agent encoding sequence for agent i can be expressed as $u^i = \{u_1^i, u_2^i, \dots u_{T_h}^i\}$.

Similarly, the low-level encoder also maps the starting node p_ξ^0 and the ending node p_ξ^1 of the lane segment ξ to ϕ_i. It then applies Fourier transforms to p_ξ^0 and p_ξ^1, and subsequently fuses them with the semantic information s_ξ to generate $\hat{p}_\xi^0 \in \mathbb{R}^2$ and $\hat{p}_\xi^1 \in \mathbb{R}^2$. Next, it utilizes an FFN to capture agent-map features:

$$z_t^{i,\xi} = \text{ReLU}\left(W_l\left[(\hat{p}_\xi^1 - \hat{p}_\xi^0), (\hat{p}_\xi^0 - \hat{p}_t^i)\right] + b_l\right), \tag{6}$$

where $z_t^{i,\xi} \in \mathbb{R}^{d_h}$ represents the agent-map feature of lane segment ξ at time step t, and $W_l \in \mathbb{R}^{d_h \times 2}$ and $b_l \in \mathbb{R}^{d_h}$ are learnable matrices. Following (3) and (4), it uses attention mechanisms to extract agent-map constraints $w_t^{i,\xi} \in \mathbb{R}^{d_h}$ at time step t between agent i and lane segment ξ from z_t^i and $z_t^{i,\xi}$. Following (5), it weights and aggregates the agent-map constraints from lane segments whose distance to target agent i is less than δ_1 at each time step to generate the agent-map encoding sequence $w^i = \{w_1^i, w_2^i, \dots w_{T_h}^i\}$ for agent i.

The low-level encoder also uses attention mechanisms to transform z_t^i into the motion encoding $m_t^i \in \mathbb{R}^{d_h}$ of agent i at time step t and stacks m_t^i across all historical time steps to form its motion encoding sequence $m^i = \{m_1^i, m_2^i, \dots m_{T_h}^i\}$. Subsequently, it concatenates m^i, u^i, and w^i with positional encodings and applies a linear transformation to produce the agent encoding sequence $s^i \in \mathbb{R}^{T_h \times d_h}$. Next, it transforms s^i into the query vector $q^i \in \mathbb{R}^{T_h \times N_h \times d_k}$, key vector $k^i \in \mathbb{R}^{T_h \times N_h \times d_k}$, and value vector $v^i \in \mathbb{R}^{T_h \times N_h \times d_k}$ and performs multi-head self-attention computation:

$$h^i = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_{N_h})W^O, \tag{7}$$

$$\text{head}_h = \text{Softmax}\left(\frac{q^i (k^i)^\top}{\sqrt{d_k}} + M\right) v^i, \tag{8}$$

where $h^i \in \mathbb{R}^{T_h \times d_h}$ represents the heterogeneous spatial-temporal feature sequence of agent i and $M \in \mathbb{R}^{T_h \times T_h}$ is an upper triangular matrix to prevent future information leakage. In this way, $h_t^i \in h^i$ aggregates heterogeneous spatial-temporal features from time steps up to t in s^i while excluding information from time steps after t. Concerned with agent response delay, the low-level encoder extracts heterogeneous spatial-temporal features from h^i for the last T_r time steps to generate the local encoding $\hat{h}^i \in \mathbb{R}^{T_r \times d_h}$ of agent i, where T_r represents agent response delay.

2) High-Level Encoder: The target agent-centric local encoding ignores large-scale spatial relationships among all observed agents, which hampers improving predictive accuracy. Thus, this study develops a high-level encoder that uses GATs to encode the global relationships among observed agents. Specifically, it considers all observed agents as nodes in a directed graph $G_{T_h} = (V, E_{T_h})$, where V represents observed agents, and E_{T_h} contains social relations among observed agents at time step T_h:

$$e_{T_h}^{i,j} = \text{ReLU}\left(W_g\left[(p_{T_h}^j - p_{T_h}^i), \sin(\Delta\theta_{T_h}^{i,j}), \cos(\Delta\theta_{T_h}^{i,j})\right] + b_g\right), \tag{9}$$

where $e_{T_h}^{i,j} \in E_{T_h}$ represents the social relation between agent i and agent j at time step T_h, $\Delta\theta_{T_h}^{i,j}$ represents the included angle between the azimuth of agent i and agent j at time step T_h, and $W_g \in \mathbb{R}^{d_h \times 2}$ and $b_g \in \mathbb{R}^{d_h}$ are learnable matrices. It then calculates the attention of agent i to agent j:

$$\alpha_{T_h}^{i,j} = \frac{\exp(\text{LeakyReLU}(a^\top(W_c\hat{h}_{T_h}^i || W_n\hat{h}_{T_h}^j || W_e e_{T_h}^{i,j})))}{\sum_{j=1}^{N_{obs}} \exp(\text{LeakyReLU}(a^\top(W_c\hat{h}_{T_h}^i || W_n\hat{h}_{T_h}^j || W_e e_{T_h}^{i,j})))}, \tag{10}$$

where $\alpha_{T_h}^{i,j} \in \mathbb{R}$ represents the attention weight of agent i to agent j, and $a \in \mathbb{R}^{d_h}$, $W_c \in \mathbb{R}^{d_h \times d_h}$, $W_n \in \mathbb{R}^{d_h \times d_h}$, and $W_e \in \mathbb{R}^{d_h \times d_h}$ are learnable matrices. On this basis, it performs weighted aggregation on local encodings:

$$g_{T_h}^i = \text{ReLU}\left(W_g\left[\sum_{j=1}^{N_{obs}} \alpha_{T_h}^{i,j}\hat{h}_{T_h}^j || \sum_{j=1}^{N_{obs}} \alpha_{T_h}^{i,j} e_{T_h}^{i,j}\right]\right), \tag{11}$$

where $g_{T_h}^i \in \mathbb{R}^{d_h}$ represents the global encoding of agent i at time step T_h, and $W_g \in \mathbb{R}^{d_h \times d_h}$ is a learnable matrix. Moreover, it linearly fuses $\hat{h}_t^i \in \hat{h}^i$ and $g_{T_h}^i$ to generate the fusion encoding $f^i \in \mathbb{R}^{T_r \times d_h}$ for agent i.

3) Multi-Modal Decoder: This study employs a Transformer decoder to translate f^i into multi-modal trajectory predictions for agent i. It utilizes attention mechanisms to decode f^i, followed by two separate FNNs that transform the decoding sequence into future potential trajectories $y^i \in \mathbb{R}^{N_k \times T_f \times 2}$ for agent i with corresponding probability distributions $P^i \in \mathbb{R}^{N_k}$.

C. Training Objective

This study adopts a fully supervised end-to-end prediction framework that generates potential future trajectories and their associated probabilities. Existing studies demonstrate that these potential future trajectories can be modeled as distinct components of a Laplacian distribution [19], which can be expressed as follows:

$$p_t^i = \sum_{k=1}^{N_k} P^{i,k} \prod_{t=1}^{T_f} \text{Laplace}\left(p_t^i | p_t^{i,k}, P^{i,k}\right), \tag{12}$$

where $y_t^i \in \mathbb{R}^2$ represents the final predicted position coordinates of agent i at time step t, and $\pi^{i,k} \in \mathbb{R}$ represents the mixture coefficient of the k-th trajectory prediction for agent i. To reduce the error between predicted trajectories and ground-truth trajectories, this study defines training loss as a weighted sum of classification loss L_{cls} and regression loss L_{reg}:

$$L = L_{cls} + \lambda L_{reg}, \tag{13}$$

where λ is a learnable parameter to balance L_{cls} and L_{reg} [53]. This study employs the cross-entropy loss of $\pi^{i,k}$ as L_{cls} and the mean negative log-likelihood loss between y_t^i and p_t^i as L_{reg}. In training, the MaTVT parameters are iteratively optimized by minimizing L until convergence is achieved.

V. Experiments and Results

A. Datasets

This study utilizes the Argoverse Motion Forecasting dataset for training and validating MaTVT, which comprises 324,557 real-world traffic scenarios and high-definition maps [21]. These traffic scenarios provide ground-truth trajectory data sampled at 10 Hz over 5 seconds, including initial 2-second observations and subsequent 3-second predictions. Additionally, each scenario includes map elements represented as lane centerlines composed of waypoints. Based on the definition of complex traffic scenarios, this study constructs a dedicated dataset comprising 265,452 traffic scenarios selected from the Argoverse dataset, characterized by either complex road structures (e.g., left/right turns and intersections) or a large number of agents (more than 8) [3], [4], [5]. The dedicated dataset is split into training, validation, and test sets, consisting of 185,816, 26,546, and 53,090 traffic scenarios, respectively. The Argoverse dataset is available at https://www.argoverse.org.

B. Implementations

This study deploys MaTVT on a blade server with an Intel Xeon Gold 5218 CPU, and dual NVIDIA Tesla V100S GPUs, and runs on Ubuntu 22.04 LTS. MaTVT is implemented using the PyTorch 1.5 framework in a Python 3.8 environment. This study employs Adam optimizer to train MaTVT [54], with a learning rate of 10^{-3}, batch size of 64, β_1 set at 0.9, β_2 set at 0.99, and epsilon set at 10^{-8}.

This study constructs the low-level encoder comprising one agent-agent encoding layer, one agent-map encoding layer, and three heterogeneous spatial-temporal feature extractors, while the high-level encoder stacks three global relation embedding layers. Moreover, this study sets the maximum target agent number at 32, the predicted modes N_k at 6, the hidden dimension d_h at 64, the attention heads N_h at 8, and the head dimension d_k equals d_h/N_h. In addition, this study determines the agent response delay T_r and the distance threshold δ_1 as 6 and 60 meters, respectively.

C. Metrics

This study uses three standard evaluation metrics to assess prediction accuracy: minimum mean Average Displacement Error (minADE$_k$), minimum mean Final Displacement Error (minFDE$_k$), and Miss Rate (MR$_k$). MinFDE$_k$ evaluates the average Euclidean distance between the generated trajectory predictions and corresponding ground-truth trajectories, and minFDE$_k$ assesses the average Euclidean distance between the endpoints of the trajectory predictions and the final ground-truth coordinates:

$$\text{minADE}_k = \frac{1}{N_{tar}T_f}\sum_{i=1}^{N_{tar}}\sum_{t=1}^{T_f}\left\|p_{t,k}^i - p_t^i\right\|, \tag{14}$$

$$\text{minFDE}_k = \frac{1}{N_{tar}}\sum_{i=1}^{N_{tar}}\left\|p_{T_f,k}^i - p_{T_f}^i\right\|. \tag{15}$$

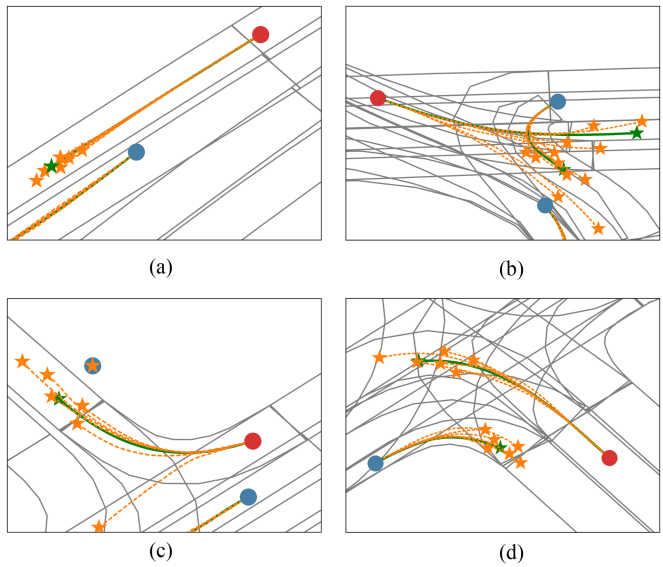

Fig. 4. Multi-modal trajectory prediction of MaTVT in multi-agent scenarios, with dots of different colors denoting different types of agents, solid lines denoting ground-truth trajectories, and dashed lines denoting predicted trajectory components. (a) Straight scene. (b) Left turn. (c) Right turn. (d) Intersection.

MR$_k$ represents the proportion of trajectory predictions with minFDE$_k$ less than 2.0 meters.

D. Results and Analysis

1) Quantitative Results: Table I quantitatively compares the prediction performance of MaTVT with benchmark methods, demonstrating consistent superiority in both single-modal and multi-modal trajectory prediction. For single-modal trajectory prediction ($k = 1$), MaTVT achieves state-of-the-art performance in all metrics, outperforming the best benchmark ProphNet (minADE$_1$: -1.34%, minFDE$_1$: -0.15%, MR$_1$: -0.76%). For multi-modal trajectory prediction ($k = 6$), MaTVT demonstrates comprehensive superior prediction performance compared to benchmarks (minADE$_6$: -0.94%, minFDE$_6$: -2.36%, MR$_6$: +2.94%). It should be noted that MaTVT performs slightly inferior to GOHOME in MR$_6$ because GOHOME is specifically optimized to decrease MR$_6$ during training. Moreover, MaTVT maintains a parameter count of 2851 K, exceeding the lightest benchmark mmTransformer (2607 K) by only 9.36%. The results collectively validate the exceptional prediction accuracy and parameter-efficient architecture of MaTVT.

2) Qualitative Results: Fig. 4 illustrates the representative prediction results of MaTVT, demonstrating its ability to perform multi-modal trajectory predictions for multiple target agents simultaneously in complex traffic scenarios, with both predicted paths and final positions exhibiting a high similarity to ground-truth trajectories. During prediction, MaTVT first generates N_k possible trajectory predictions and then produces either all possible predictions with corresponding probabilities or the highest-probability alternative as the final output. This validates the flexibility of MaTVT in multi-agent trajectory prediction, enabling accurate trajectory forecasting while fully

TABLE I
PREDICTION COMPARISON BETWEEN MATVT AND BENCHMARKS. THE BEST RESULT IS BOLDFACED, WHILE THE SECOND-BEST RESULT IS UNDERLINED

Model	minADE_1	minFDE_1	MR_1	minADE_6	minFDE_6	MR_6	#Param
LaneGCN [14]	1.702	3.763	0.589	0.870	1.364	0.163	3701K
mmTransformer [40]	1.774	4.003	0.599	0.844	1.338	0.154	**2607K**
TPCN [55]	1.575	3.488	0.564	0.815	1.244	0.133	-
GOHOME [56]	1.703	3.682	0.582	0.943	1.450	**0.102**	5100K
SceneTransformer [57]	1.811	4.055	0.591	0.803	1.232	0.125	15296K
HiVT [19]	1.562	3.441	0.563	0.774	1.169	0.127	**2529K**
GANet [58]	1.592	3.455	0.560	0.806	1.161	0.118	-
Wayformer [46]	1.636	3.656	0.564	0.768	1.162	0.119	-
DCMS [59]	1.576	3.451	0.550	0.766	1.135	0.109	-
ProphNet [60]	1.491	3.263	0.525	0.762	1.134	0.110	-
HPNet [61]	1.504	3.302	0.526	0.743	1.099	0.107	-
MaTVT	**1.471**	**3.258**	**0.521**	**0.736**	**1.073**	0.105	2815K

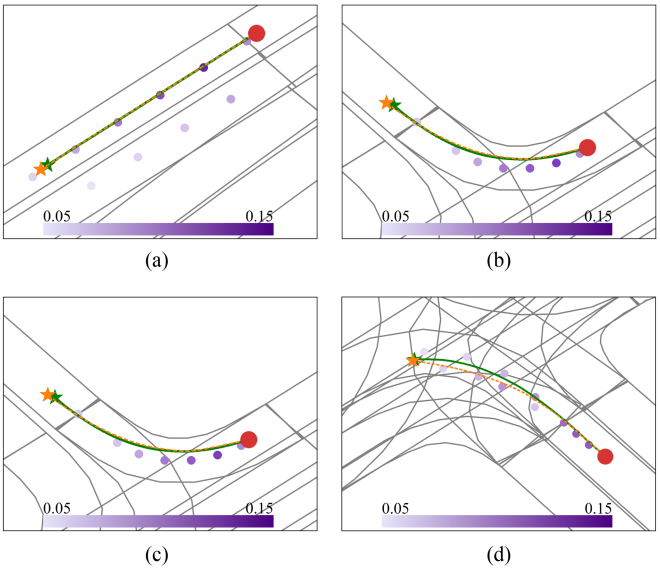

Fig. 5. Attention bias of the component with the high probability in multi-modal trajectory prediction toward waypoints. Representative examples only display waypoints with attention scores exceeding 0.05, with darker colors indicating higher attention bias. (a) Straight scene. (b) Left turn. (c) Right turn. (d) Intersection.

accounting for uncertainties in future motions of surrounding agents.

Fig. 5 illustrates the attention bias towards the waypoints in multi-modal trajectory predictions. To improve visualization, these representative examples display only the component with the highest probability in multi-modal trajectory predictions and its waypoint preferences. The attention bias of this trajectory component for all input waypoints undergoes softmax normalization to obtain attention scores, with those exceeding 0.05 being displayed. These representative examples demonstrate that MaTVT not only correctly identifies the lane where the ground-truth trajectory is located, but also utilizes waypoints to enhance its prediction accuracy, improving both predictive performance and interpretability.

3) Ablation Studies: To confirm the contributions of functional components to prediction performance, this study develops MaTVT variants and conducts three independent ablation

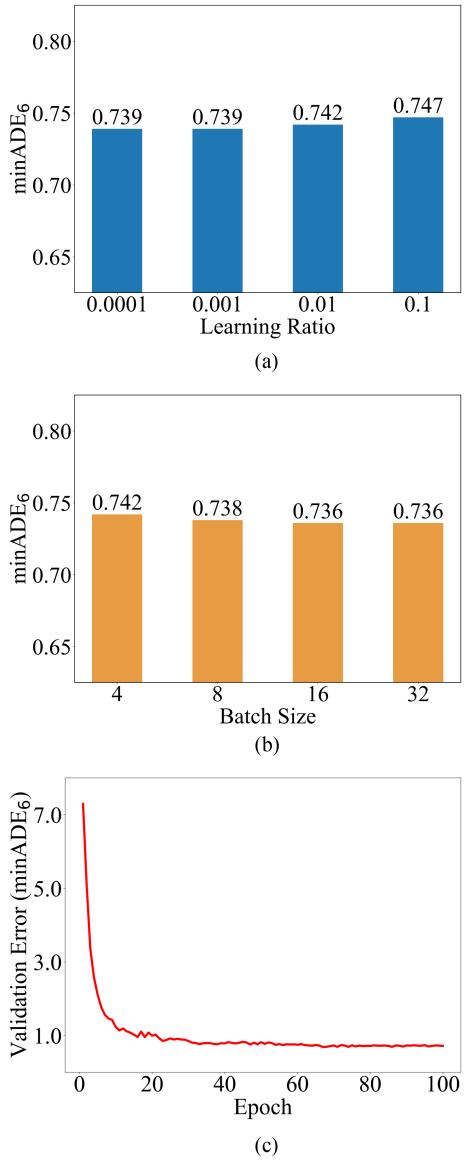

Fig. 6. Sensitivity analysis of MaTVT on learning rate and batch size, and the model convergence at the optimal learning rate and batch size. (a) The minADE_6 of MaTVT at different learning rates and batch sizes. (b) The minADE_6 of MaTVT at different batch sizes and the optimal learning rate. (c) The convergence of MaTVT at the optimal learning rate and batch size.

TABLE II
ABLATION STUDIES OF DIFFERENT FUNCTIONAL COMPONENTS IN MATVT TO ITS PREDICTION PERFORMANCE

Model	minADE_1	minFDE_1	MR_1	minADE_6	minFDE_6	MR_6	Delay (ms)
MaTVT_{noAAE}	1.563	3.492	0.554	0.791	1.142	0.115	93
MaTVT_{noAME}	1.618	3.679	0.576	0.825	1.217	0.122	90
MaTVT_{noGE}	1.500	3.293	0.529	0.747	1.104	0.108	85
$\text{MaTVT}_{complete}$	1.471	3.258	0.521	0.736	1.073	0.105	97
$\text{MaTVT}_{T_r=1}$	1.656	3.564	0.562	0.825	1.204	0.128	87
$\text{MaTVT}_{T_r=2}$	1.571	3.475	0.553	0.781	1.139	0.119	88
$\text{MaTVT}_{T_r=3}$	1.521	3.367	0.537	0.758	1.108	0.113	90
$\text{MaTVT}_{T_r=4}$	1.493	3.301	0.527	0.747	1.092	0.109	92
$\text{MaTVT}_{T_r=5}$	1.479	3.273	0.523	0.740	1.080	0.107	94
$\text{MaTVT}_{T_r=6}$	1.471	3.258	0.521	0.736	1.073	0.105	97
$\text{MaTVT}_{T_r=7}$	1.468	3.251	0.520	0.734	1.069	0.105	99
$\text{MaTVT}_{T_r=8}$	1.466	3.249	0.520	0.734	1.067	0.105	102
$\text{MaTVT}_{\delta_1=20m}$	1.582	3.509	0.558	0.795	1.166	0.121	67
$\text{MaTVT}_{\delta_1=40m}$	1.515	3.357	0.533	0.760	1.113	0.114	79
$\text{MaTVT}_{\delta_1=60m}$	1.471	3.258	0.521	0.736	1.073	0.105	97
$\text{MaTVT}_{\delta_1=80m}$	1.475	3.264	0.523	0.733	1.065	0.104	130

TABLE III
PARAMETER ANALYSIS ON THE IMPACT OF HIDDEN DIMENSION AND ATTENTION HEADS ON MATVT

Model	minADE_1	minFDE_1	MR_1	minADE_6	minFDE_6	MR_6
$\text{MaTVT}_{d_h=16}$	1.483	3.278	0.526	0.744	1.089	0.108
$\text{MaTVT}_{d_h=32}$	1.476	3.266	0.523	0.738	1.077	0.106
$\text{MaTVT}_{d_h=64}$	1.471	3.258	0.521	0.736	1.073	0.105
$\text{MaTVT}_{d_h=128}$	1.469	3.254	0.521	0.735	1.071	0.105
$\text{MaTVT}_{N_h=2}$	1.491	3.285	0.528	0.748	1.094	0.109
$\text{MaTVT}_{N_h=4}$	1.475	3.265	0.523	0.739	1.081	0.107
$\text{MaTVT}_{N_h=8}$	1.471	3.258	0.521	0.736	1.073	0.105
$\text{MaTVT}_{N_h=16}$	1.473	3.262	0.522	0.738	1.079	0.106

studies. Except for modified components, these variants utilize the same experimental setups as the complete model. During testing, this study records the sample size and processing time for each batch of test samples and utilizes the total number of samples and total processing time to calculate the average delay time per sample to measure the computational efficiency of MaTVT and its variants.

The first study alternately removes agent-agent encodings (AAE), agent-map encodings (AME), and global encodings (GE) from MaTVT to investigate their contributions. Table II indicates that AAE improves the prediction accuracy of MaTVT (minADE_1: -5.89%, minFDE_1: -6.70%, MR_1: -5.96%, minADE_6: -6.95%, minFDE_6: -6.04%, MR_6: -8.70%), indicating its ability to comprehend agent-agent interactions. The contributions of AME (minADE_1: -9.09%, minFDE_1: -11.44%, MR_1: -9.55%, minADE_6: -10.79%, minFDE_6: -11.83%, MR_6: -13.93%) and GE (minADE_1: -1.06%, minFDE_1: -1.51%, MR_1: -1.47%, minADE_6: -2.81%, minFDE_6: -3.67%, MR_6: -2.86%) are also confirmed, demonstrating that MaTVT utilizes them to capture features from map elements and global relationships to enhance prediction accuracy.

The second study investigates the impact of heterogeneous spatial-temporal features on model performance. Table II indicates that the increase in response delay T_r enhances the prediction accuracy of MaTVT, indicating that heterogeneous spatial-temporal features not only compensate for the response delay of agents but also filter out noise from long-term historical data. However, this improvement exhibits diminishing marginal returns and essentially reaches saturation when T_r is equal to 6. Meanwhile, the increase in T_r consecutively burdens the complexity of spatial-temporal dependency modeling and the computational cost of noise suppression, resulting in a superlinear decline in the computational efficiency of MaTVT. Therefore, MaTVT sets T_r to 6 to identify heterogeneous spatial-temporal features that optimally balance prediction accuracy and computational efficiency. The configuration enhances the interpretability of MaTVT by elucidating its performance boundaries and operational mechanisms.

The third study investigates the impact of the local encoding scope on MaTVT. Table II indicates that the predictive accuracy of MaTVT improves with increasing δ_1 until it exceeds 60 m. One possible explanation for this trend is that target agents focus on nearby agents and traffic conditions during navigation, while neglecting those farther away. Moreover, the average computation delay of MaTVT increases rapidly with increasing δ_1. Therefore, this study sets δ_1 at 60 m to ensure both accurate prediction and efficient computation.

4) Parameters Analysis: This study conducts two parameter studies to investigate the effects of the hidden dimension d_h and the attention heads N_h on the prediction performance of MaTVT. Except for these parameters, the MaTVT structure

and its remaining parameters and experimental settings remain unchanged. Table III shows that MaTVT's prediction performance remains largely stable in different d_h and N_h, indicating its strong robustness.

5) Sensitivity Analysis: This study conducts two additional experiments to investigate the impact of the learning rate and batch size on convergence during MaTVT training. In these experiments, this study presets a series of learning rates and batch sizes and uses minADE_6 to measure the convergence of MaTVT in different training configurations. Fig. 6(a) compares the training effects of MaTVT at different learning rates and batch sizes and reveals that MaTVT achieves optimal performance when the learning rate is set to 0.001. Fig. 6(b) presents the training effects of MaTVT with optimal learning rates across different batch sizes, determining 16 as the optimal batch size. Fig. 6(c) records the effect of MaTVT training after each training epoch with the appropriate learning rate and batch size. The MaTVT training curve initially drops sharply and then gradually stabilizes, demonstrating its strong convergence.

VI. Conclusion

This study integrates GAT and Transformer architecture to develop MaTVT, a novel multi-agent trajectory prediction method for complex traffic scenarios. It constructs a dual-level encoder: the lower-level encoder captures local encodings from historical data and map information, while the high-level encoder converts inter-agent relationships into global encodings. On this basis, MaTVT integrates local and global encodings into fused encodings, which are decoded into multi-modal trajectory predictions for target agents. This study selects complex traffic scenarios from the Argoverse dataset to train and validate MaTVT, confirming its outstanding accuracy, interpretability, robustness, and computational efficiency.

The experimental results indicate three key advantages of MaTVT. Firstly, MaTVT develops the hierarchical fusion mechanism that extracts local encodings from motion features, agent-agent interactions, and agent-infrastructure constraints and fuses them with global encodings representing spatial relationships between agents for multi-agent trajectory prediction. Secondly, MaTVT constructs a heterogeneous spatial-temporal feature extractor that not only achieves an optimal balance between prediction accuracy and computational efficiency by compensating for agent response delays but also indirectly reveals its attention bias towards historical spatial-temporal data to enhance model interpretability. Thirdly, MaTVT features a lightweight architecture with strong adaptability, enabling accurate multi-modal trajectory predictions for multiple agents simultaneously in various traffic scenarios.

Future research will focus on MaTVT in three key aspects. Firstly, MaTVT will be further optimized to improve its prediction accuracy and computational efficiency. Secondly, MaTVT will classify different driving styles and utilize them to guide local encoding generation, illuminating their impact on multi-modal trajectory prediction. Thirdly, MaTVT will be packaged as a functional module and integrated into AVs as a safety constraint for their predictive cruise control systems.

References

[1] Z. Huang, J. Wu, and C. Lv, "Driving behavior modeling using naturalistic human driving data with inverse reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10239–10251, Aug. 2022, doi: 10.16383/j.aas.c211108.

[2] A. Li, Z. Xu, W. Li, Y. Chen, and Y. Pan, "Urban signalized intersection traffic state prediction: A spatial–temporal graph model integrating the cell transmission model and transformer," *Appl. Sci.*, vol. 15, no. 5, 2025, Art. no. 2377. [Online]. Available: https://www.mdpi.com/2076-3417/15/5/2377

[3] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transp. Res. Part C, Emerg. Technol.*, vol. 60, pp. 416–442, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X15003447

[4] J. Guo, Y. Luo, and K. Li, "Adaptive coordinated collision avoidance control of autonomous ground vehicles," *Proc. Inst. Mech. Engineers, Part I, J. Syst. Control Eng.*, vol. 232, no. 9, pp. 1120–1133, 2018.

[5] Z. Liu, J. Chen, H. Xia, and F. Lan, "Quasi-critical collision-avoidance strategy for autonomous vehicles in complex traffic scenarios based on exclusive area of relative velocity vector algorithm," *Robot. Auton. Syst.*, vol. 153, 2022, Art. no. 104049. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889022000197

[6] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 7, no. 3, pp. 652–674, Sep. 2022, doi: 10.1109/TIV.2022.3167103.

[7] H. Yin, Y. Wen, and J. Li, "A survey of vehicle trajectory prediction based on deep-learning," in *Proc. 3rd Int. Conf. Neural Netw., Inf. Commun. Eng.*, Guangzhou, China, Feb. 2023, pp. 140–144, doi: 10.1109/NNICE58320.2023.10105706.

[8] M. Schreiber, S. Hoermann, and K. Dietmayer, "Long-term occupancy grid prediction using recurrent neural networks," in *Proc. Int. Conf. Robot. Automat.*, Montreal, QC, Canada, May 2019, pp. 9299–9305, doi: 10.1109/ICRA.2019.8793582.

[9] M. Schreiber, V. Belagiannis, C. Gläser, and K. Dietmayer, "Dynamic occupancy grid mapping with recurrent neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xian, China, May 2021, pp. 6717–6724, doi: 10.1109/ICRA48506.2021.9561375.

[10] A. Zyner, S. Worrall, and E. Nebot, "A recurrent neural network solution for predicting driver intention at unsignalized intersections," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1759–1764, Jul. 2018, doi: 10.1109/LRA.2018.2805314.

[11] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1341–1352, Feb. 2020, doi: 10.1109/TVT.2019.2960110.

[12] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "CoverNet: Multimodal behavior prediction using trajectory sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2020, pp. 14062–14071, doi: 10.1109/CVPR42600.2020.01408.

[13] R. Chandra, U. Bhattacharya, A. Bera, and D. Manocha, "Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 8475–8484, doi: 10.1109/CVPR.2019.00868.

[14] M. Liang et al., "Learning lane graph representations for motion forecasting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Glasgow, U.K., Aug. 2020, pp. 541–556, doi: 10.1007/978-3-030-58536-5_32.

[15] J. Gu, C. Sun, and H. Zhao, "DenseTNT: End-to-end trajectory prediction from dense goal sets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, BC, Canada, Oct. 2021, pp. 15283–15292, doi: 10.1109/ICCV48922.2021.01502.

[16] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *Proc. 25th Int. Conf. Pattern Recognit.*, Milan, Italy, Jan. 2021, pp. 10335–10342, doi: 10.1109/ICPR48806.2021.9412190.

[17] Z. Huang, X. Mo, and C. Lv, "Multi-modal motion prediction with transformer-based neural network for autonomous driving," in *Proc. Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, May 2022, pp. 2605–2611, doi: 10.1109/ICRA46639.2022.9812060.

[18] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 6000–6010. [Online]. Available: https://api.semanticscholar.org/CorpusID:13756489

[19] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "HIVT: Hierarchical vector transformer for multi-agent motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, Jun. 2022, pp. 8813–8823, doi: 10.1109/CVPR52688.2022.00862.

[20] S. Matija, "The reaction times of drivers aged 20 to 80 during a divided attention driving," *Traffic Inj. Prevention*, vol. 17, no. 8, pp. 810–814, 2016, doi: 10.1080/15389588.2016.1157590.

[21] M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Los Angeles, CA, USA, Jun. 2019, pp. 8740–8749, doi: 10.1109/CVPR.2019.00895.

[22] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 528–540, Dec. 2006, doi: 10.1109/TITS.2006.883115.

[23] M. Brännström, E. Coelingh, and J. Sjöberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 658–669, Sep. 2010, doi: 10.1109/TITS.2010.2048314.

[24] P. Yuyan Annie, J. Guo, Y. Chen, Q. Cheng, W. Li, and Y. Liu, "A fundamental diagram based hybrid framework for traffic flow estimation and prediction by combining a Markovian model with deep learning," *Expert Syst. Application*, vol. 238, 2024, Art. no. 122219, doi: 10.1016/j.eswa.2023.122219.

[25] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *Proc. IEEE 5th Int. Conf. Intell. Comput. Commun. Process.*, Cluj-Napoca, Romania, Aug. 2009, pp. 417–422, doi: 10.1109/ICCP.2009.5284727.

[26] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 549–562, Sep. 2007, doi: 10.1109/TITS.2007.903439.

[27] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *Proc. IEEE Proc. Intell. Veh. Symp.*, Las Vegas, NV, USA, Jun. 2005, pp. 319–324, doi: 10.1109/IVS.2005.1505122.

[28] Y. Wang, Z. Liu, Z. Zuo, Z. Li, L. Wang, and X. Luo, "Trajectory planning and safety assessment of autonomous vehicles based on motion prediction and model predictive control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8546–8556, Sep. 2019, doi: 10.1109/TVT.2019.2930684.

[29] S. Klingelschmitt, M. Platho, H.-M. Groß, V. Willert, and J. Eggert, "Combining behavior and situation information for reliably estimating multiple intentions," in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 388–393, doi: 10.1109/IVS.2014.6856552.

[30] J. Joseph, F. Doshi-Velez, A. S. Huang, and N. Roy, "A Bayesian nonparametric approach to modeling motion patterns," *Auton. Robots*, vol. 31, pp. 383–400, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:3020880

[31] Q. Tran and J. Firl, "Online maneuver recognition and multimodal trajectory prediction for intersection assistance using non-parametric regression," in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 918–923, doi: 10.1109/IVS.2014.6856480.

[32] G. S. Aoude, B. D. Luders, K. K. H. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Funchal, Portugal, Sep. 2010, pp. 1855–1862, doi: 10.1109/ITSC.2010.5625287.

[33] P. Kumar, M. Perrollaz, S. Lefèvre, and C. Laugier, "Learning-based approach for online lane change intention prediction," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Gold Coast, QLD, Australia, Jun. 2013, pp. 797–802, doi: 10.1109/IVS.2013.6629564.

[34] H. Berndt, J. Emmert, and K. Dietmayer, "Continuous driver intention recognition with hidden Markov models," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Beijing, China, Oct. 2008, pp. 1189–1194, doi: 10.1109/ITSC.2008.4732630.

[35] Y. Wang, C. Wang, W. Zhao, and C. Xu, "Decision-making and planning method for autonomous vehicles based on motivation and risk assessment," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 107–120, Jan. 2021, doi: 10.1109/TVT.2021.3049794.

[36] L. Sun, W. Zhan, and M. Tomizuka, "Probabilistic prediction of interactive driving behavior via hierarchical inverse reinforcement learning," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, Maui, HI, USA, Nov. 2018, pp. 2111–2117, doi: 10.1109/ITSC.2018.8569453.

[37] D. Xu et al., "Learning from naturalistic driving data for human-like autonomous highway driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7341–7354, Dec. 2021, doi: 10.1109/TITS.2020.3001131.

[38] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Los Angeles, CA, USA, Nov. 2017, pp. 204–211, doi: 10.1109/IVS.2017.7995721.

[39] S. Choi, J. Kim, and H. Yeo, "TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning," *Transp. Res. Part C, Emerg. Technol.*, vol. 128, no. 0968–090X, 2021, Art. no. 103091, doi: 10.1016/j.trc.2021.103091.

[40] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, Jun. 2021, pp. 7573–7582, doi: 10.1109/CVPR46437.2021.00749.

[41] H. Liu, Z. Huang, X. Mo, and C. Lv, "Augmenting reinforcement learning with transformer-based scene representation learning for decision-making of autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 9, no. 3, pp. 4405–4421, Mar. 2024, doi: 10.1109/TIV.2024.3372625.

[42] B. He and Y. Li, "Multi-future transformer: Learning diverse interaction modes for behaviour prediction in autonomous driving," *IET Intell. Transport Syst.*, vol. 16, no. 9, pp. 1249–1267, 2022, doi: 10.1109/TIV.2024.3372625.

[43] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, BC, Canada, Oct. 2021, pp. 9793–9803, doi: 10.1109/ICCV48922.2021.00967.

[44] J. Zhao, X. Li, Q. Xue, and W. Zhang, "Spatial-channel transformer network for trajectory prediction on the traffic scenes," 2021, *arXiv:2101.11472*. [Online]. Available: https://api.semanticscholar.org/CorpusID:231718998

[45] K. Zhang, X. Feng, L. Wu, and Z. He, "Trajectory prediction for autonomous driving using spatial-temporal graph attention transformer," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22343–22353, Nov. 2022, doi: 10.1109/TITS.2022.3164450.

[46] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "WayFormer: Motion forecasting via simple & efficient attention networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, London, U.K., May 2023, pp. 2980–2987, doi: 10.1109/ICRA48891.2023.10160609.

[47] Z. Zhou, J. Wang, Y. Li, and Y. Huang, "Query-centric trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, Jun. 2023, pp. 17863–17873, doi: 10.1109/CVPR52729.2023.01713.

[48] X. Jia, L. Sun, H. Zhao, M. Tomizuka, and W. Zhan, "Multi-agent trajectory prediction by combining egocentric and allocentric views," in *Proc. Conf. Robot Learn.*, London, U.K., Nov. 2021, pp. 1434–1443. [Online]. Available: https://api.semanticscholar.org/CorpusID:246037656

[49] M. Kofinas, N. S. Nagaraja, and E. Gavves, "Roto-translated local coordinate frames for interacting dynamical systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 6417–6429. [Online]. Available: https://api.semanticscholar.org/CorpusID:240070754

[50] B. Varadarajan et al., "Multipath: Efficient information fusion and trajectory aggregation for behavior prediction," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 7814–7821.

[51] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *Proc. Int. Conf. Mach. Learn.*, Graz, Austria, Jul. 2021, pp. 4651–4664. [Online]. Available: https://api.semanticscholar.org/CorpusID:232110866

[52] M. Tancik et al., "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2020, pp. 1–11, doi: 10.5555/3495724.3496356.

[53] L. Thiede and P. Brahma, "Analyzing the variety loss in the context of probabilistic trajectory prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, Oct. 2019, pp. 9953–9962.

[54] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, USA, May 2015, pp. 1–15. [Online]. Available: https://hdl.handle.net/11245/1.505367

[55] M. Ye, T. Cao, and Q. Chen, "TPCN: Temporal point cloud networks for motion forecasting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, Jun. 2021, pp. 11313–11322, doi: 10.1109/CVPR46437.2021.01116.

[56] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "GOHOME: Graph-oriented heatmap output for future motion estimation," in *Proc. Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, May 2022, pp. 9107–9114, doi: 10.1109/ICRA46639.2022.9812253.

[57] J. Ngiam et al., "Scene transformer: A unified architecture for predicting future trajectories of multiple agents," in *Proc. Int. Conf. Learn. Representations, Virtual-Only*, Apr. 2022, pp. 1–22. [Online]. Available: https://api.semanticscholar.org/CorpusID:251648671
[58] M. Wang et al., "GANet: Goal area network for motion forecasting," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2022, pp. 1609–1615. [Online]. Available: https://api.semanticscholar.org/CorpusID:252383191
[59] M. Ye, J. Xu, X. Xu, T. Cao, and Q. Chen, "DCMS: Motion forecasting with dual consistency and multi-pseudo-target supervision," 2022, *arXiv:2204.05859*. [Online]. Available: https://api.semanticscholar.org/CorpusID:263864486
[60] X. Wang, T. Su, F. Da, and X. Yang, "ProphNet: Efficient agent-centric motion forecasting with anchor-informed proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21995–22003.
[61] X. Tang, M. Kan, S. Shan, Z. Ji, J. Bai, and X. Chen, "HPNet: Dynamic trajectory forecasting with historical prediction attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 2024, pp. 15261–15270.

Anran Li is currently working toward the Ph.D. degree in transportation engineering with the College of Metropolitan Transportation, Beijing University of Technology, Beijing, China. He is also with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing, and Beijing Key Laboratory of Transportation Engineering, Beijing University of Technology. His research interests include intelligent transportation systems and autonomous vehicle technologies, particularly in microscopic traffic prediction, vehicle trajectory prediction, and cooperative planning, and control strategies for intelligent connected vehicles.

Hongsheng Yu received the master's degree in transportation engineering from the Shanghai University of Engineering Science, Shanghai, China, in 2022. He is currently an Assistant Researcher with the Institute of Electronic Computing Technology, China Railway Science Research Institute Group Company, Ltd. He has participated in two overseas projects and 11 provincial and ministerial research projects. His main research interests include rail transit engineering, railway information technology, traffic safety, and satellite navigation.

Yuyan Pan received the Ph.D. degree in transportation engineering from the Beijing University of Technology, Beijing, China, in 2023. She is currently a Postdoctoral Scholar with the Department of Civil and Environmental Engineering, Pennsylvania State University, University Park, PA, USA. Her work combines fundamental traffic theory with AI-based methods to enhance traffic system efficiency and resilience. She has authored or coauthored more than 20 peer-reviewed papers. Her research interests include connected and automated vehicles, electric vehicle charging, traffic flow theory, and data-driven modeling. She is a Reviewer of top journals such as *Transportation Research Part B/C/E*.

Zhenlin Xu received the M.Sc. degree in transport, infrastructure, and logistics from the Delft University of Technology, Delft, Netherlands, in 2025. He is currently a Junior Researcher with the Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology. His research interests include intersection of extended reality (virtual reality and augmented reality), digital twins, automated vehicles, and behavior analysis of road users.

Huibo Bi (Member, IEEE) received the Ph.D. degree in electrical and electronic engineering from Imperial College London, London, U.K., in 2017, under the supervision of Prof. Erol Gelenbe. He is currently a Lecturer with the College of Metropolitan Transportation, Beijing University of Technology, Beijing, China and Visiting Scholar with the Department of Electronic Engineering, Tsinghua University, Beijing, under the supervision of Prof. Xiaoming Tao. He has authored or coauthored more than 40 academic articles in peer-reviewed journals and conferences. His research interests include intelligent network systems, emergency management, machine learning, and energy conservation. He was the recipient of Beijing High-Level Youth Talent Support Program.

Bolin Gao received the B.E. and M.E. degrees in vehicle engineering from Jilin University, Changchun, China, in 2007 and 2009, respectively, and the Ph.D. degree in vehicle engineering from Tongji University, Shanghai, China, in 2013. He is currently an Associate Research Professor with the School of Vehicle and Mobility, Tsinghua University, Beijing, China. His research interests include the theoretical research and engineering application of the dynamic design and control of intelligent and connected vehicles, especially collaborative perception and tracking methods in cloud control systems, intelligent predictive cruise control systems on commercial trucks with cloud control mode, and test and evaluation of intelligent vehicle driving systems.

Keqiang Li received the B.Tech. degree from Tsinghua University Beijing, China, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering from the Chongqing University of China, Chongqing, China, in 1988 and 1995, respectively. He is currently a Professor with the School of Vehicle and Mobility, Tsinghua University. He is leading the national key project on intelligent and connected vehicles, China. He has authored more than 200 articles. He is a Co-Inventor of more than 80 patents in China and Japan. His main research interests include automotive control systems, driver assistance systems, and networked dynamics and control. He was a fellow member of the Society of Automotive Engineers of China, Chairperson of Expert Committee of China Industrial Technology Innovation Strategic Alliance for ICVs (CAICV), and CTO of China ICV Research Institute Company Ltd.. He was on the Editorial Boards of the *International Journal of Vehicle Autonomous Systems*. He was the recipient of Changjiang Scholar Program Professor and National Award for Technological Invention in China.

Yanyan Chen received the M.S. degree from the School of Civil Engineering, Zhengzhou University, Zhengzhou, China, in 1994, and the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 1997. She is currently the Dean with Metropolitan Transportation College, Beijing University of Technology, Beijing, China. She has authored or coauthored more than 100 peer-reviewed papers and ten books. She has authorized 14 China invention patents. Her research interests include transportation Big Data, transportation planning and management, and intelligent transportation. She is also the Vice-Chairperson of the Transportation Branch of China Highway and Transportation Society.