Spatio-Temporal Context Graph Transformer Design for Map-Free Multi-Agent Trajectory Prediction

Zhongning Wang, Jianwei Zhang, Jicheng Chen, *Member, IEEE*, and Hui Zhang, *Senior Member, IEEE*

***Abstract*—Predicting the motion of surrounding vehicles is an important function of autonomous vehicles. However, most of the current state-of-the-art trajectory prediction models rely heavily on map information. In order to overcome the shortcomings of the existing models, our paper proposes a map-free trajectory prediction model and names it TR-Pred (Trajectory Relative two-stream Prediction). The trajectory stream employs LSTM to embedding the trajectory information of each agent. Subsequently, it utilizes graph neural networks (GNN) to extract latent traffic information in the current scenario, such as lane lines, drivable areas, and traffic control conditions. The relative stream utilizes temporal transformer to capture the local relative movement among agents. Subsequently, it employs GNN to extract the interaction information of all target agent. We augment the temporal transformer through refine initialization of its class token. This refined thereby enable enhanced modeling of inter-agent relative motion correlations between the agents. The decoder predicts the target agent by incorporating the historical interaction information among agents with latent traffic information. We validate TR-Pred on the Argoverse dataset, the highD dataset and the rounD dataset. The results show that TR-Pred performs better in the minimum Average Displacement Error compared to the main map-base model use in 2020, 2021. Experiments on Argoverse results show that our framework achieves a 16.3%/19.7%/24.4% improvement in minADE/minFDE(minimum Final Displacement Error)/MR(Miss Rate) compared to CRAT-Pred. The experimental results on highD and rounD show that, compared to the map-free version HiVT, our framework achieves improvements of 18.8%/18.8%/25.0% and 8.3%/8.5%/7.4% in minADE/minFDE/MR, respectively.**

***Index Terms*—Autonomous driving, trajectory prediction, motion forecasting, machine learning, deep learning, map-free trajectory prediction.**

I. Introduction

With the development and application of advanced autonomous vehicles (AVs) technologies, safety and efficiency have become more and more important in AV systems. In complex driving scenes such as intersections scenes, roundabouts scenes and highway on-ramp merging scenes, there are a large number of agents around AV. While encountering those scenarios, AV need to predict the future trajectory of agents around them and speculate on multi-agent interactions. However, because of uncertain future scenarios, agents might operate variably under the same scene [1], [2]. Hence, multi-modal trajectory prediction (MTP) needs to generate various feasible trajectories of an agent. By incorporating MTP, the performance of the planning module is greatly enhanced, ensuring a more robust and reliable AVs system. For these reasons, agent trajectory prediction has attracted constant interest in the area of autonomous vehicles.

With the development and gradual adoption of high-definition (HD) maps, more and more companies and institutions begin to incorporate HD maps into downstream tasks for AVs. The emergence of datasets like Argoverse further facilitates related research by the researchers [3]. In order to obtain better performance in trajectory prediction, researchers have begin to focus their research on how to represent HD map better [4], [5], [6], [7]. Wayformer [8] uses cross-attention to encode information such as maps, traffic control, and trajectory. LaneGCN [9] works by refining lane centreline information as nodes of a graph neural network (GNN) and using lane changes as edges of the graph. Vectornet [10] interacts with global information through transformers by treating each lane line, footpath and other element as a node of the graph. However, the above models do not have a well-designed module for extracting trajectories information. Once these models lose the semantic information of the HD map, they either do not work or their prediction accuracy is severely degraded.

In practice, the HD map faces enormous challenges. HD map are difficult for numerous applications because of their high acquisition and update costs and large storage overhead [11]. Moreover, the issues concerning precise positioning in AVs also influence the utilization of HD maps. Both high-rise buildings and elevated structures can cause signal interference. Similarly, in tunnels, the loss of satellite signals further exacerbates this issue, and HD maps are unavailable in such situations. Moreover, the sudden failure of HD maps after AV systems have been activated can severely deteriorate the performance of trajectory prediction. This deterioration of trajectory prediction can critically impair the safety of automated driving. Therefore, map-based trajectory prediction has some limitations in practice. The key difference between map-free and map-based trajectory prediction is that map-free methods do not rely on HD map and accurate vehicle localization. It does not need the HD map information

Manuscript received 22 September 2023; revised 22 October 2023; accepted 29 October 2023. Date of publication 3 November 2023; date of current version 23 February 2024. This work was supported by the National Natural Science Foundation of China under Grant U1964201. *(Corresponding author: Jianwei Zhang.)*

Zhongning Wang and Jianwei Zhang are with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130025, China (e-mail: wangzn21@mails.jlu.edu.cn; zhangjw@jlu.edu.cn).

Jicheng Chen and Hui Zhang are with the School of Transportation Engineering and Science, Beihang University, Beijing 100191, China (e-mail: jichengc@buaa.edu.cn; huizhang285@buaa.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2023.3329885.

Digital Object Identifier 10.1109/TIV.2023.3329885

as input. Map-free methods depend only on target information provided by sensors, giving them wider applicability compared to map-based methods. When AVs drive in scenarios where HD map is unavailable map-based trajectory prediction methods often cannot function. Thus, map-free methods have greater versatility for AVs in diverse environments lacking HD map.

For these reasons, it becomes necessary to develop a map-free model in trajectory prediction. In such cases, it becomes imperative to rely on the constraints derived from the surrounding agents' interaction and motion data to constrain the predicted trajectories. By using this crucial information, the system can accurately predict the trajectories of other agents. In this work, we propose a two-stream map-free prediction method. It can consider the multimodal information of the trajectories, the interaction and the relative motion information between agents. Because its two streams are the trajectory stream and the relative stream, we term it "Trajectory Relative two-stream Prediction (TR-Pred)." This work is dedicated to predicting the future trajectory of multi-agent scenes without map information. The framework is validated and tested on the ArgoverseV1.1 dataset. The results indicate that our model achieves state-of-the-art (SOTA) performance compared to previous map-free models. In summary, our main contributions are:

- We propose a map-free vehicle trajectory prediction method. This method represents the historical trajectories as a trajectory stream and a relative stream. In each stream of the TR-Pred, we use a GNN with the transformer mechanism for the information interaction between the agents.
- We propose a new temporal transformer. This module uses LSTM encoding to initialize the class token. We find that LSTM encoding helps the temporal transformer to focus on temporal motion from the input.
- TR-Pred achieves SOTA performance in map-free prediction method. Meanwhile, TR-Pred outperforms map-based prediction models, which are proposed in the previous two years, in the minADE metric.

The rest of this paper is organized as follows: Section II describes the related work on trajectory prediction research. Section III discusses the current research problem and gives detailed definitions. Section IV provides the structure of the model. Section V provides the ablation experiments, visualization results and comparison results. Finally, conclusions are presented in Section VI.

II. Related Work

Trajectory prediction methods include physics-based methods, classical machine learning-based methods, deep learning-based methods, and reinforcement learning-based methods. However, current long sequence forecasting predominantly leverages deep learning. Among these methods, we have categorized the representation of agents into three forms.

Rasterization-based approaches: These methods use Bird's-Eye-View (BEV) image and require complex representation rules to create the rasterization image [12], [13]. Yuning Chai et al. perform feature extraction and state analysis on different agents from the BEV perspective, and agents interact through multi-level Convolutional Neural Networks (CNN) [4]. Linhui Li et al. predict raster map and historical trajectories by constructing a raster map and subsequently employing a CNN network with a Multilayer Perceptron (MLP) [14]. In rasterized BEV maps, one has the flexibility to insert and use a variety of information, including both trajectories and maps, while using a variety of common image-processing backbone networks. Thomas Gilles et al. obtain a raster heat map containing future trajectory possibilities by encoding a rasterized BEV map through a replaceable CNN backbone network [15]. However, rasterization techniques have some limitations. For instance, while coarser raster grids may reduce computational complexity, they can also lead to substantial loss of information. Moreover, too detailed rasterization can lead to difficulties in data preprocessing and a rapid rise in computational complexity, leading to the inability to real-time [13]. Therefore, people begin to find a new way of processing information.

Node-based approaches: This method is inspired by the development of GNN and the problems in Rasterization-based approaches. Researchers use a graph to represent the information of agents and maps [9], [16]. This way solves the problem of redundant information and high computation brought on by rasterization [17]. Lots of approaches are proposed, representing each agent as a node and then aggregating context via GNN, including Graph Convolutional Networks (GCNs) [18], Graph Attention Networks (GATs) [19], [20], and transformers [21], [22]. Kunpeng Zhang et al. construct a motion graph of the agent at each moment directly after the video inputs [23]. Dongwei Xu et al. construct a traffic target map for each moment centered on the target agent and subsequently encode the information through a GNN [24]. Theodor Westny et al. obtain the target's future trajectory by constructing a traffic map for each moment, which is decoded by attentive Graph-GRU and Kalman filter [25]. However, Node-based approaches exhibit two core deficiencies: first, they are incapable of representing the relative motion between objects at each moment in time; second, they possess a relative lack of clarity in the representation of nodes other than the center object.

Vector-based approaches: Vector representations are demonstrated to effectively encode map information. Inspired by the OpenDrive [26], researchers use a vector to represent the information of agents. Moreover, the vectorized information is permutation invariant so that it can be easily combined with transformers, GCN, and other GNN mechanisms. So, much SOTA work is carried out using vectorized information in combination with GNN [27]. For instance, The work by Jiyang Gao et al. pioneered the application of vectorized trajectory and lane boundary information to the task of trajectory prediction [10]. Zhibo Wang et al. encode and distillation learns the inputs by vectorized agent history trajectories and vectorized center lanes [28].

The above-mentioned methods apply to both map-based approaches and map-free approaches. However, among all SOTA models tend to use a combination of trajectory and map information. Yuxuan Han et al. achieve better results by combining the map with traffic rules and historical vehicle trajectories [29]. After graphing the HD map, Xing Gao et al. reach SOTA by combining historical trajectories and HD map [30]. The work of Zhou et al., which win in the competitions for the ArgoverseV1.1 and ArgoverseV2, fully accounts for map and trajectories and uses interactions between agents and map [31]. However, HD maps have started to reveal some issues in their usage. Similar to other AVs systems, researchers also begin investigating how to perform trajectory prediction without relying on HD maps. Julian Schmidt et al., in their work, achieves good results in the ArgoverseV1.1 competitions only using historical trajectories [32]. Jing lian et al. achieve better results than crat-pred using only historical trajectories through GAT, IDCNN, and multi-attention mechanisms [27]. Our work is related to that of Julian Schmidt et al. [32] and Zhou et al. [33]. In our framework, the interactions between each agent at every time step are fully taken into account, and the corresponding traffic rules can be inferred from the trajectories of surrounding agents.

Due to the multimodality of agent trajectory prediction, researchers tend to make a priori assumptions about the probability of trajectories. The earliest models tend to use the Gaussian mixture model (GMM) [4], [34], [35]. Recently, there have been frameworks that use Laplace distribution for fitting [33]. Both models have been shown to perform well in multimodal prediction of trajectories. Also, both probabilistic models are equally well adapted to different datasets, such as Argoverse, Waymo [3], [36]. Therefore, in our work, we use the Laplace distribution.

III. Problem Formulation

In this paper, there are N traffic participants, so we can use

$$P_{hist} = \{\rho_{h_1}, \rho_{h_2}, \ldots, \rho_{h_M}, \ldots, \rho_{h_N}\} \tag{1}$$

for the traffic participants, where P_{hist} denotes the set of historical trajectories for all agents in the current prediction task, ρ_{h_N} represents the trajectory of the ego vehicle, $[\rho_{h_1}, \ldots, \rho_{h_M}]$ represent the historical trajectories of agents that need to be predicted, M is the number of prediction targets. Each historical trajectory in P_{hist} can be represented by :

$$\rho_{h_i} = \left\{\rho_{i_c}, \rho_{i_g}, \rho_i^t, \rho_i^{t+1}, \ldots, \rho_i^{T_f}\right\}, \tag{2}$$

where ρ_{h_i} denotes the history trajectory of the ith agent, t means the start frame of the ith agent, T_f means the end frame of the ith agent, and ρ_{i_c} denotes the difference in coordinates between the agent and the ego at frame t, ρ_{i_g} denotes the agent's position in the global coordinate system at frame t. With the introduction of ρ_{i_c} and ρ_{i_g}, our network has Permutation Invariance. Each frame of the ith agent is

$$\rho_i^t = [x_t, y_t], \tag{3}$$

where (x_t, y_t) denotes the centroid position of the agent in the local coordinate system at timestamp t.

Some studies have found that outputting a single predicted trajectory can be deemed unreasonable. A single trajectory does not yet accurately express the likely future trajectory of the target and, thus, the target's intentions. Therefore, during the trajectory prediction task, we conduct multimodal trajectory predictions for all agents at the same moment. So when we predict the trajectory P_p, we can represent results as

$$P_p = \{\rho_{p_1}, \rho_{p_2}, \ldots, \rho_{p_M}\}. \tag{4}$$

where M is the number of targets for which single object predictions can be made. The prediction for the ith agent can be denoted by:

$$\rho_{pi} = \{\rho_{pi}^1, \rho_{pi}^2, \ldots, \rho_{pi}^U\}, \tag{5}$$

where U denotes the total number of trajectories we need to predict. Each predicted trajectory

$$\rho_{pi}^j = \{(x_0, y_0), (x_1, y_1), \ldots, (x_H, y_H)\}. \tag{6}$$

where j is the jth predicted trajectory, (x_i, y_i) is the estimated coordinates and H denotes the number of steps we need to predict. In the prediction, since the output time step is determined according to the dataset or actual usage requirements, it is not necessary to predict the time for each step.

IV. Methodology

A. Overall Model

The general structure for the trajectory prediction network is shown in Fig. 1. TR-Pred comprises two streams: a trajectory stream and a relative stream. The trajectory stream includes two modules: the trajectory feature extraction module and the global trajectory interaction module. The core of the trajectory feature extraction module is an LSTM encoder. The input of the trajectory feature extraction module is a single trajectory. The core of the global trajectory interaction module is a GNN module using the transformer mechanism. It uses the output from the trajectory feature extraction module to get U global interactivity feature for each agent. In this module, the main role of the GNN is to infer the latent traffic context of the current road from the trajectory of each vehicle. This includes the drivable area, traffic control, and some lane information. This enables the token corresponding to the target vehicle to contain information about the relevant traffic states. The main modules in the relative stream are the local encoder module and the relative motion global interaction module. The local encoder consists of a graph feature encoder, an Agent relative motion encoder (AR encoder) module and a temporal transformer module. The GNN module uses a transformer mechanism containing edge information for local information aggregation and the temporal transformer for extracting full-time features of the center target. The relative interaction module is a GNN module using the transformer mechanism. It uses the output from the relative encoder module to get the U global interactivity feature for each agent. In this module, the primary role of GNN is to aggregate information from each local scene. Consequently, the model can obtain global interactive information between each target agent and other agents. This enables the target agent token to access information about the target vehicle's interaction with other vehicles. Finally, an MLP decodes use the inputs from both streams and directly generates the trajectories within the next H steps.

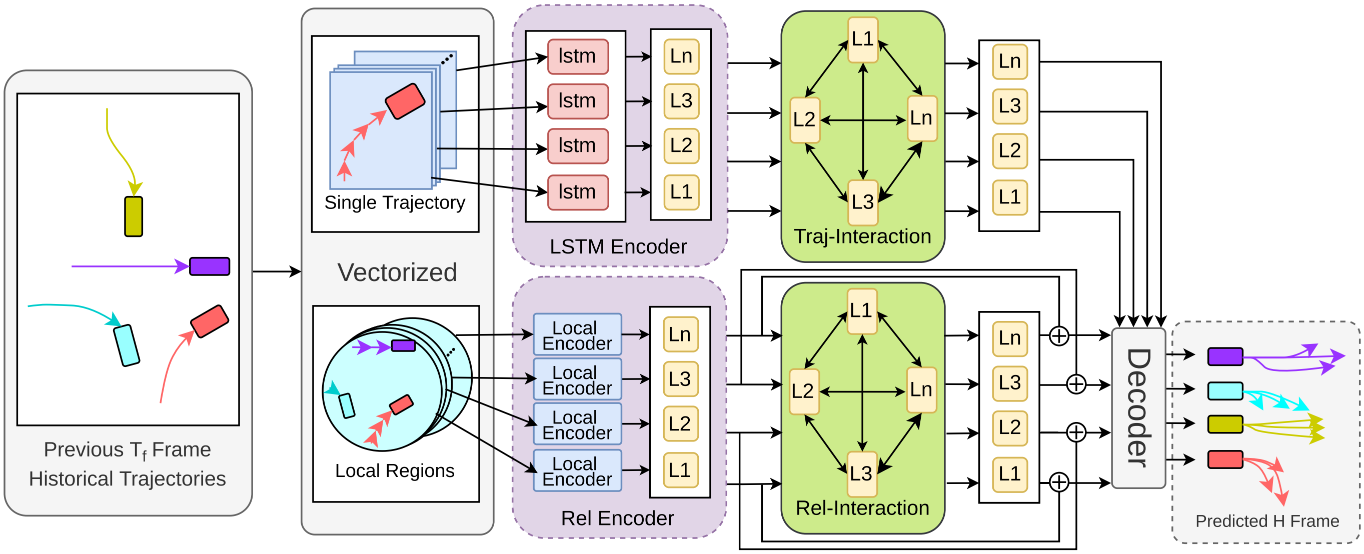

Fig. 1. Architecture of TR-Pred. This figure illustrates the two-stream structure of our model, where data passes through two distinct encoders and contextual aggregators prior to decoding. Key: Rel encoder: relative encoder; Rel-Interactivity: relative interactivity; Traj-Interactivity: global trajectory interactivity.

B. LSTM Encoder

In order to improve the generalization performance, we rotate the trajectory by a suitable angle. So, We use the rotated and normalized agent historical trajectory as the LSTM encoder input. However, the agent's trajectory does not always start at timestamp 0. In contrast to other networks [9], [32], we incorporate relative moments as inputs so that trajectories that are not at the moment 0 can also be added to the input. Since the sampling period of the dataset is not always uniformly distributed, the network cannot simply take in the number of sampling steps as input like other networks [32], [33]. We need to also incorporate the timestamp as inputs for learning the acceleration, deceleration, and other behaviors of the agents. The concrete representation of the input information can be represented by

$$\rho_{hist} = \left\{\rho_i^t, \rho_i^{t+1}, \ldots, \rho_i^{T_f}\right\}, \tag{7}$$

with ρ_{hist} denotes the input trajectory, ρ_i^j being the relative position at moment j and $\rho_i^j = [x_j; y_j]$. Based on this information, a single layer LSTM encoder can be used to capture temporal information for the agent itself, and the encoder weights are shared. The LSTM can be denoted by:

$$h_i^t = LSTM(\rho_i^j, h_i^{t-1}, c_i^{t-1}) \tag{8}$$

with output h_i^t, hidden state c_i^{t-1} and previous moment output h_i^{t-1} are vector of size dm. dm is 128, and LSTM finally output id h_i.

C. Trajectory Interactivity

Obtaining spatial interactions is a crucial challenge in trajectory prediction. The GAT method cannot flexibly extract interaction information of interest. Therefore, inspired by the Graph Transformer [37], we use a transformer with relative positional embedding to extract interaction features. Since we have normalized the trajectories of each agent, it is necessary to select a key feature point within each trajectory point for the construction of edges in GNN. We select the feature points through:

$$\rho_{i_c} = \begin{cases} \rho_i^t & \text{if } t < T_f; \\ \rho_i^{T_f} & \text{otherwise.} \end{cases} \tag{9}$$

Similar to the Graph Transformer, we extend the transformer to add information about the edges between nodes. We use MLP $\Psi_{rel}(\cdot)$ to obtain edge embeddings e_{ij} from agent i to agent j. So, we construct information about the edges between two different trajectories i and j by $\rho_{i_c}, \rho_{j_c}, \Delta\theta_{ij}$, where $\Delta\theta_{ij}$ denotes $\theta_i - \theta_j$, $Rot_i \in \mathbb{R}^{2\times 2}$, $e_{ij} \in \mathbb{R}^{dm}$, θ_i and θ_j denotes the angle of rotation.

$$e_{ij} = \Psi_{rel}\left(\left[Rot_i^\top\left(\rho_{i_c}^\top - \rho_{j_c}^\top\right); \cos\left(\Delta\theta_{ij}\right); \sin\left(\Delta\theta_{ij}\right)\right]\right). \tag{10}$$

In (11), $x \in [1, 2, \ldots, dk]$, $W_x^{Q^{\text{Traj}}} \in \mathbb{R}^{dm\times dh}$, $W_x^{K^{\text{Traj}}} \in \mathbb{R}^{2dm\times dh}$, $W_x^{V^{\text{Traj}}} \in \mathbb{R}^{2dm\times dh}$ are the learnable matrices, where $dh = dm/dk$. dk represents the number of attention heads used in the transformer.

$$\begin{aligned} \tilde{q}_i^x &= h_i W_x^{Q^{\text{Traj}}}, \\ \tilde{k}_{ij}^x &= [h_j; e_{ij}]\, W_x^{K^{\text{Traj}}}, \\ \tilde{v}_{ij}^x &= [h_j; e_{ij}]\, W_x^{V^{\text{Traj}}}, \end{aligned} \tag{11}$$

We then put the $\tilde{q}_i^x, \tilde{k}_{ij}^x, \tilde{v}_{ij}^x$ into the scaled dot product attention module to obtain the aggregated information. The scaled dot product attention module is formulated as follows:

$$m_i^x = \sum_{j\in\mathcal{N}_i} \operatorname{softmax}\left(\frac{\tilde{q}_i^x}{\sqrt{d_k}} \cdot \left[\left\{\tilde{k}_{ij}^{x\top}\right\}_{j\in\mathcal{N}_i}\right]\right) \tilde{v}_{ij}^x \tag{12}$$

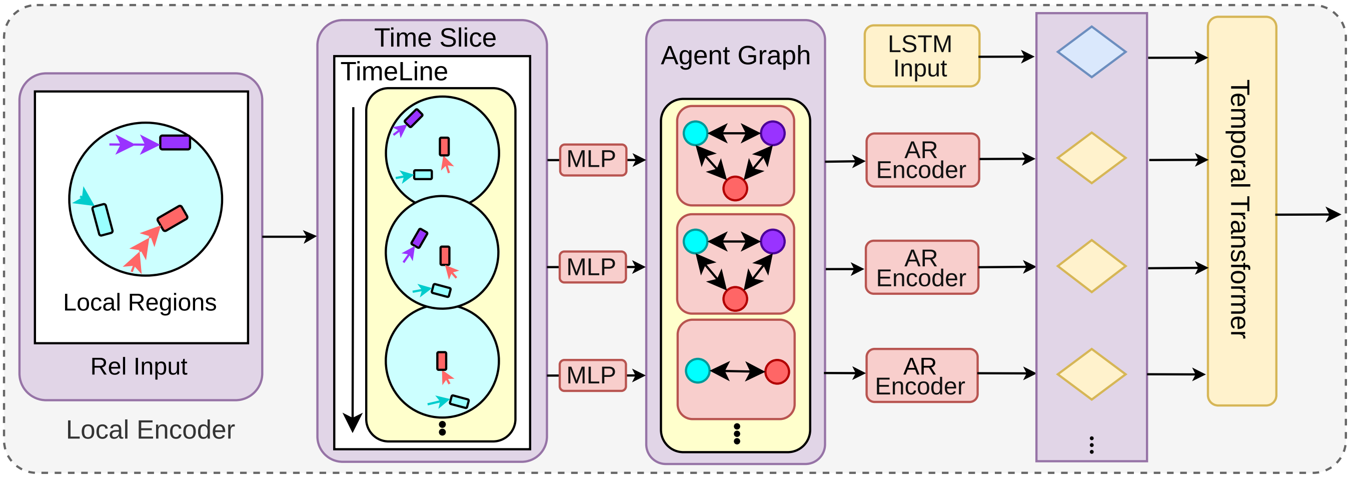

Fig. 2. Architecture of the Local Encoder.

where $\mathcal{N}_i$ is the set of all trajectories except i, $m_i^x \in \mathbb{R}^{dh}$.

$$m_i^{traj} = concat(m_i^1, m_i^2, \ldots, m_i^{dk}) + h_i \tag{13}$$

The MLP module is followed by the dot product attention module.

$$l_i = Relu(MLP(m_i^{traj})) + m_i^{traj} \tag{14}$$

l_i is vector of dm. Notably, we applied layer normalization before formulas (11) and (14).

D. Related Encoder

The related encoder consists of one or more local encoder components in parallel. Fig. 2 provides a detailed illustration of the structure of each module in Local Encoder. The local encoder has four main modules:

- Time Slice
- Graph Build
- Agent Relative Encoder, AR Encoder
- Temporal Transformer

Time Slice is aimed at providing Rotate-Translation Invariance. Rotate-Translation Invariance [33] is essential for enabling our network to have the simultaneous output of trajectory information from multiple agents. In the module, we choose the appropriate rotation matrix Rot based on the trajectory of the center agent. Then, rotate both the center agent i and the surrounding agents accordingly. So we can be represented input Γ_{p_i} by:

$$\Gamma_{p_i} = \left\{ Rot, \gamma_{p_i}^0, \gamma_{p_i}^1, \ldots, \gamma_{p_i}^{T_f} \right\}. \tag{15}$$

where $\gamma_{p_i}^t$ denotes the set of agent local coordinates within the center agent's region at t time step, the origin of the local coordinate system is situated at the center agent's position at time 0. In Γ_{p_i},

$$\gamma_{p_i}^t = \left\{ t, \Phi_i^t, \{\Phi_{ij}^t\}_{j \in \mathcal{A}_i} \right\}, \tag{16}$$

$\mathcal{A}_i$ denotes the local coordinates of surrounding agents within the center agent's region, Φ_i^t represents the position displacement of the center agent from time 0 to the current time step t, and $\Phi_{ij}^t = (x_j^i, y_j^i)$, where (x_j^i, y_j^i) is the local coordinate for agent j.

Graph Build consists of two blocks, namely, the center agent embedding and the surrounding agent embedding. Two different MLPs are used to obtain each embedding. In these MLPs, $(\Phi_i^t - \Phi_i^{t-1})$ and $(\Phi_j^t - \Phi_j^{t-1})$ is the motion information of the agent between adjacent moments, $(\Phi_j^t - \Phi_i^t)$ is the relative distance coordinates between the surrounding agent and the center agent at the current moment. ai and aj are semantic attributes of agent i and agent j, respectively. Each MLP can be represented by:

$$z_i = \Psi_{\mathrm{c}} \left(\left[Rot_i^{\top} \left(\Phi_i^t - \Phi_i^{t-1} \right)^{\top} ; a_i \right] \right), \tag{17}$$

$$z_{ij} = \Psi_{\mathrm{n}} \left(\left[Rot_i^{\top} \left(\Phi_j^t - \Phi_j^{t-1} \right)^{\top} ; Rot_i^{\top} \left(\Phi_j^t - \Phi_i^t \right)^{\top} ; a_j \right] \right), \tag{18}$$

where, $z_i^t \in \mathbb{R}^{dm}$, $z_{ij}^t \in \mathbb{R}^{dm}$, $\Psi_{\mathrm{c}(\cdot)}$ and $\Psi_{\mathrm{n}(\cdot)}$ are different MLPs.

AR Encode aggregates information of the constructed graph through the graph transformer module. In this module, we use the center agent embedding to generate the query vector and use the surrounding agents to generate the key and value vectors. These are denoted as $q_{i_x}^{ar}, k_{ij_x}^{ar}, v_{ij_x}^{ar}$. In (19), $W^{Q^{\mathrm{AR}}}{}_x \in \mathbb{R}^{dm \times dh}$, $W_x^{K^{\mathrm{AR}}} \in \mathbb{R}^{dm \times dh}$, $W_x^{V^{\mathrm{AR}}} \in \mathbb{R}^{dm \times dh}$ are the learnable matrices.

$$q_{i_x}^{ar} = z_i W_x^{Q^{\mathrm{AR}}}, \quad k_{ij_x}^{ar} = z_{ij} W_x^{K^{\mathrm{AR}}}, \quad v_{ij_x}^{ar} = z_{ij} W_x^{V^{\mathrm{AR}}}, \tag{19}$$

Then, we feed these values into the scaled dot product attention mechanism to compute:

$$m_{i_x}^{ar} = \sum_{j \in \mathcal{A}_i} \operatorname{softmax} \left(\frac{q_{i_x}^{ar}}{\sqrt{d_k}} \cdot \left[\left\{ k_{ij_x}^{ar \top} \right\}_{j \in \mathcal{A}_i} \right] \right) v_{ij_x}^{ar} \tag{20}$$

In the previous research, it is found that the gating unit has a specific effect enhancement for information extraction [38], so we also included the gating unit as the input and the fusion of the extracted information.

$$m_i^{ar} = concat(m_{i_1}^{ar}, m_{i_2}^{ar}, \ldots, m_{i_{dk}}^{ar}) \tag{21}$$

$$g_i = \text{sigmoid}\left(\left[z_i; m_i^{ar}\right] W^{\text{gate}}\right), \tag{22}$$

$$\hat{z}_i = g_i \odot W^{\text{self}} z_i + (1 - g_i) \odot m_i^{ar}, \tag{23}$$

where $W^{\text{gate}} \in \mathbb{R}^{2dm \times 1}$, $W^{\text{self}} \in \mathbb{R}^{dm \times dm}$ are the learnable matrices. The MLP module is followed by the dot product attention module.

$$S_i = Relu(MLP(\hat{z}_i)) + \hat{z}_i \tag{24}$$

S_i is vector of dm. Notably, we apply layer normalization before formulas (19) and (24).

Temporal Transformer is used to obtain the temporal features of the center agents. This is because the input vector S_i is just the positional feature at a single timestamp. Our module differs from Detr and Hivt in that we did not use learnable parameters [33], [39]. We use LSTM embedding to initialize class tokens instead of learnable parameters. In (25), we add the positional embedding (T_A) with the S_i.

$$\hat{S}_i = S_i + T_A, \tag{25}$$

where $\hat{S}_i \in \mathbb{R}^{(T_f+1)\times dm}$, $T_A \in \mathbb{R}^{(T_f+1)\times dm}$, T_A is a learnable parameter.

$$q_{i_x}^{time} = \hat{S}_i W_x^{Q^{\text{time}}}, k_{i_x}^{time} = \hat{S}_i W_x^{K^{\text{time}}}, v_{i_x}^{time} = \hat{S}_i W_x^{V^{\text{time}}}, \tag{26}$$

where $W_x^{Q^{\text{time}}} \in \mathbb{R}^{dm\times dh}, W_x^{K^{\text{time}}} \in \mathbb{R}^{dm\times dh}, W^{V^{\text{time}}}{}_x \in \mathbb{R}^{dm\times dh}$, are learnable matrix.

$$\tilde{S_{i_x}} = \text{softmax}\left(\frac{q_{i_x}^{time} {k_{i_x}^{time}}^\top}{\sqrt{d_k}} + Mask\right) v_{i_x}^{time}, \tag{27}$$

$$m_{uv} = \begin{cases} -\infty & \text{if } u < v; \\ 0 & \text{otherwise,} \end{cases} \tag{28}$$

$$Mask = \begin{bmatrix} 0 & -\infty & \cdots & -\infty \\ 0 & 0 & \cdots & -\infty \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{(T_f+1)\times(T_f+1)} \tag{29}$$

where we use the $Mask$ to make the model pay more attention to the information from the previous timestamps. The MLP module is followed by the dot product attention module.

$$\tilde{S}_i = Relu(MLP(concat(\tilde{S_{i_1}}, \tilde{S_{i_2}}, \ldots, \tilde{S_{i_{dk}}}))) + \hat{S}_i \tag{30}$$

Same as the AR encoder, we apply layer normalization before formulas (26) and (30).

E. Global Interactivity

Region features are obtained through the local encoder, necessitating a global interaction module to capture the remote dependencies in the scene. Similar to the Trajectory Interaction module, we employ the graph transformer for global information aggregation. However, in contrast to the trajectory interaction module, each center agent must possess trajectory information in the final frame T of the historical trajectory. The geometric relationship between agent i and agent j can be represented in the following way:

$$E_{ij} = \Psi_r\left(\left[Rot_i^\top\left(\rho_j^{T_f} - \rho_i^{T_f}\right)^\top; \cos\left(\Delta\theta_{ij}\right); \sin\left(\Delta\theta_{ij}\right)\right]\right). \tag{31}$$

where $\Psi_r(\cdot)$ is MLP layer, $E_{ij} \in \mathbb{R}^{dm}$.

$$q_{i_x}^g = \tilde{S}_i W_x^{Q^{\text{G}}}, k_{ij_x}^g = [\tilde{S}_i; E_{ij}] W_x^{K^{\text{G}}}, v_{ij_x}^g = [\tilde{S}_i; E_{ij}] W_x^{V^{\text{G}}}, \tag{32}$$

where $W_x^{Q^{\text{G}}} \in \mathbb{R}^{dm\times dh}, W_x^{K^{\text{G}}} \in \mathbb{R}^{2dm\times dh}, W^{V^{\text{G}}}{}_x \in \mathbb{R}^{2dm\times dh}$, are learnable matrix.

$$\tilde{M}_{i_x} = \sum_{j\in\mathcal{N}_i} \text{softmax}\left(\frac{{q_{i_x}^{g}}^\top}{\sqrt{d_k}} \cdot \left[\left\{k_{ij_x}^g\right\}_{j\in\mathcal{N}_i}\right]\right) v_{ij_x}^g \tag{33}$$

In (34) $\mathcal{N}_i$ donate the set of all trajectories except i. It is worth noting that the $\mathcal{N}_i$ here is sometimes inconsistent with the $\mathcal{N}_i$ of the Trajectory Interaction Module, as not all trajectories can be represented with center embeddings.

$$L_i = Relu(MLP(concat(\tilde{M_{i_1}}, \tilde{M_{i_2}}, \ldots, \tilde{M_{i_{dk}}}))) + \tilde{S}_i \tag{34}$$

Same as the AR encoder, we apply layer normalization before formulas (32) and (34).

F. Decoder

The most important characteristic of the future trajectory of a vehicle is its multimodality. Therefore, the output of multimodal trajectories is the core of trajectory prediction tasks. We use the Laplace distribution to fit the uncertainty of the trajectory. The model predicts trajectories using an MLP where the inputs are concatenated with the features extracted from the two streams.

$$\hat{P}_p = Relu(MLP([l; L; \tilde{S}])) \tag{35}$$

The model outputs $\hat{P}_p$ a tensor of size $[U, M, H, 4]$, where U denotes the number of outputs required for multimodal outputs, M denotes the number of agents in the current scene that can meet the requirements for predicting a trajectory, and H represents the number of steps that need to be predicted in the contemporary scene. To aid in training, we use another MLP and a softmax function to predict the mixing coefficients for each agent's mixing model in the shape of $[U, M]$. The specific model architecture of the decoder is shown in the Fig. 3.

G. Training

The loss function consists of two parts. One part is a regression loss function, and the other part is a classification loss function. The negative log-likelihood function of the Laplace distribution serves as the regression loss function. The cross-entropy loss serves as the classification loss function. Their ratio is 1:1.

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{cls}}, \tag{36}$$

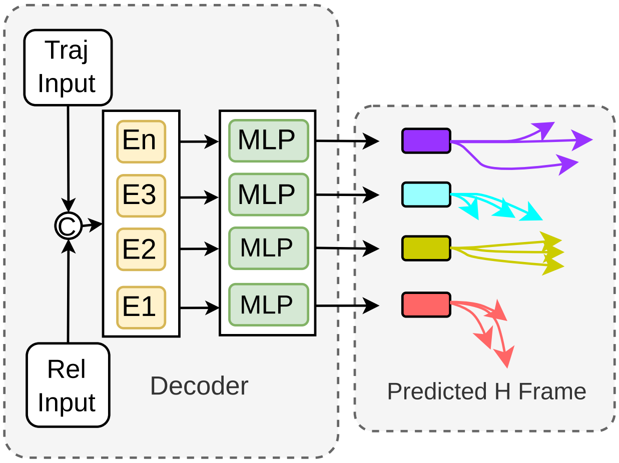

Fig. 3. Specific structure of the decoder is shown to obtain the final output trajectory through the inputs of the two modes.

In (37) $P(\cdot|\cdot)$ denote the Laplace distribution, $\hat{\boldsymbol{\mu}}_i^t, \hat{b}_i^t$ are denoted the location of the optimal trajectory and uncertainty, respectively.

$$\mathcal{L}_{\text{reg}} = -\frac{1}{NH}\sum_{i=1}^{N}\sum_{t=T_f+1}^{T_f+H} \log\text{P}\left(Rot_i^\top\left(\rho_i^t - \rho_i^{T_f}\right)^\top \mid \hat{\boldsymbol{\mu}}_i^t, \hat{b}_i^t\right). \tag{37}$$

We compute $\mathcal{L}_{\text{cls}}$ utilizing the cross-entropy loss function. It is worth noting that we only optimize the best-predicted trajectory each time. The best trajectories are selected by 2-paradigm selection.

V. Experiments

A. Experimental Setup

1) Dataset: Our framework is trained and validated on the Argoverse dataset. We submit to Evalai's public leaderboard[1] (07/06/2023). The task of Argoverse Motion ForecastingV1.1 is to predict the target's trajectory for the next 3 seconds from the first 2 seconds of the agent's historical trajectory. The dataset contains 323,557 real-vehicle driving scenarios in Miami and Pittsburgh, divided into training, validation, and test sets roughly in a ratio of 5:1:2 (train: 205,942; val: 39,472; test: 78,143). In addition, experiments are also conducted on the highD [40] and RounD [41] datasets. The highD dataset is collected on German highways, with vehicle speeds generally above 80 km/h and maximum speeds exceeding 160 km/h. The RounD dataset is collected at German roundabouts using drones, with more complex traffic situations and increased interaction between agent motions compared to other scenarios. Since the sampling rates of these two datasets are 25 Hz, which does not meet trajectory prediction requirements, we resampled them to 12.5 Hz. We predict the next 30 sample points based on the first 20 points, for a total prediction time of 4 s. We spilt these two datasets into training, testing, and validation sets. The roundD dataset has 4,366, 541 and 569 data points in the training, testing, and validation sets, respectively. The highD dataset has 11,848, 1,536 and 1,564 data points in the training, testing, and validation sets, respectively. The spilting ratio for both datasets is 8:1:1.

2) Metrics: We choose standard metrics for motion prediction to evaluate our model. The standard metrics include minimum Average Displacement Error (minADE), minimum Final Displacement Error (minFDE), and Miss Rate (MR). Using these metrics, we need to predict K possible trajectories for the target. The minADE measures the accuracy of the predicted trajectory by calculating the l_2 distance between the best predicted trajectory $\hat{\tau}_{i,k}^t$ and the actual trajectory τ_i^t. The specific calculations are as follows

$$\min \text{ADE}_K = \frac{1}{N}\frac{1}{H}\min_{k=1}^{K}\sum_{i=1}^{N}\sum_{t=1}^{H}\left\|\hat{p}_{pi,k}^t - p_{pi}^t\right\|_2. \tag{38}$$

The minFDE is calculated as the minimum final displacement error between the best predicted endpoints and the actual endpoints over K predictions.

$$\text{minFDE}_K = \frac{1}{N}\min_{k=1}^{K}\sum_{i=1}^{N}\left\|\hat{p}_{pi,k}^H - p_{pi}^H\right\|_2 \tag{39}$$

In (38) and (39), N is the total number of agents, and K denotes that we generate K predictions for each agent.

MR is calculated as the ratio of the predicted endpoints less than 2 meters from the actual endpoint.

3) Implementation Details: The model is trained with 64 epochs by the ADAMw optimizer. The hidden layer size is 128, the batch size is 128, the initial learning rate is 10^{-4}, and the weight decay and drop-out rates are 10^{-4} and 0.1, respectively. We use the cosine annealing scheduler to attenuate the learning rate. In the trajectory stream, we use one layer of the LSTM module and three layers of the trajectory interaction module in the trajectory stream. In the relative stream, we use a layer of local modules, four layers of temporal transformer modules, and three layers of trajectory interaction modules in our relative stream. The number of heads for the multi-head attention is 8, and the number of predictive modes K is set to 6. The radius of all local regions is 50 meters. We train for 7 hours on a single RTX3090, with one epoch taking around 5 minutes.

B. Results

1) Qualitative Results: We present qualitative results for TRpred on the Argoverse validation set. For the sake of an intuitive presentation, we have visualized the predictions. We have selected a few representative scenarios for demonstration, which include vehicle acceleration, deceleration, turning, lane changing, and traveling straight. Interestingly, despite the absence of semantic information from the map, our model still accurately predicts the vehicle's actions, such as lane changing and lane centering. We can predict the acceleration and deceleration of vehicles at intersections even in the absence of map and traffic control information, as depicted in Fig. 4. Our model likewise accurately determines vehicle lane changes without relying on a

[1] [Online]. Available: https://eval.ai/web/challenges/challenge-page/454/overview

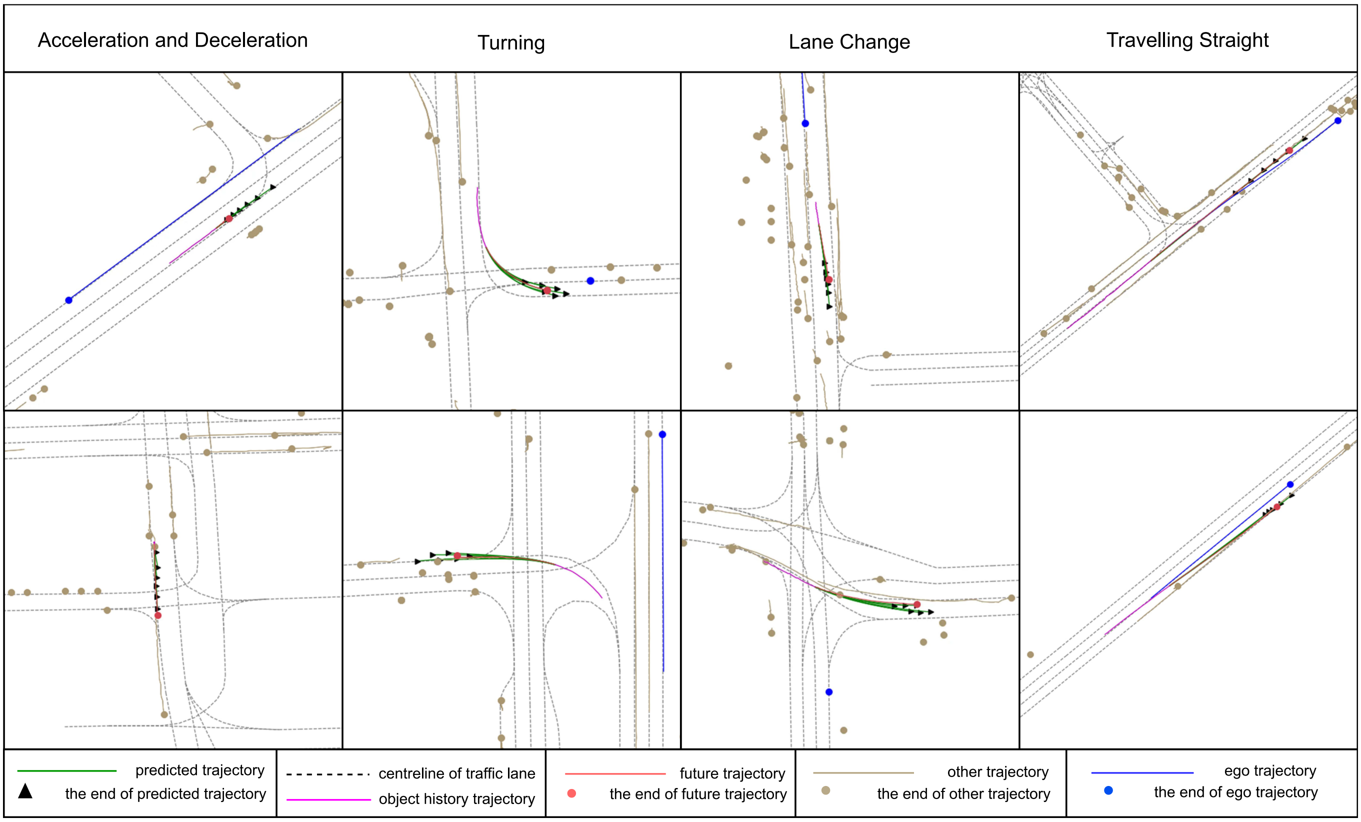
Acceleration and Deceleration
Turning
Lane Change
Travelling Straight
predicted trajectory
▲ the end of predicted trajectory
centreline of traffic lane
object history trajectory
future trajectory
● the end of future trajectory
other trajectory
● the end of other trajectory
ego trajectory
● the end of ego trajectory

Fig. 4. This figure demonstrates accurate prediction results for several common driving scenarios, including vehicle acceleration/deceleration, turning, lane changing, and traveling straight. In this depiction, various elements are color-coded for clarity: blue is the ego vehicle, light brown is the neighboring traffic participant, magenta is the historical trajectory, red is the true trajectory, green is the predicted trajectory, and the dotted line is the center line of the lane line. With the exception of the predicted vehicle, all agents drew only the first two seconds of trajectory.

map. Remarkably, we found that even without map information and despite the vehicle having no intention of changing lanes, our model infer the possibility of making a left or right turn in the current lane based on the trajectories of surrounding vehicles, as shown in Fig. 5. These observations indirectly validate the effectiveness of the interaction module in our model.

2) Comparison With State-of-The-Art: In order to validate the proposed framework, we compare the results obtained on the Argoverse dataset with those of several SOTA models from recent years. We uniformly compare the results obtained by these models on the validation set with the number of predicted trajectories, K set to 6. Our model is compared with the following map-free prediction methods:

- Nearest neighbors (NN) baseline [3].
- Crystal Graph Convolutional Neural Networks with Multi-Head Self-Attention (CRAT) [32].
- Encoding the intrinsic interaction information for vehicle trajectory prediction (Liannet) [27].
- Multi-Agent Tensor Fusion for Contextual Trajectory Prediction (MATF) [42].
- Diverse and Admissible Trajectory Forecasting through Multimodal Context Understanding (CAM) [43].
- Towards Safe Autonomy in Hybrid Traffic: Detecting Unpredictable Abnormal Behaviors of Human Drivers via Information Sharing (MEATP) [44].
- A Fast and Map-Free Model for Trajectory Prediction in Traffics (F ^{2} net) [45].

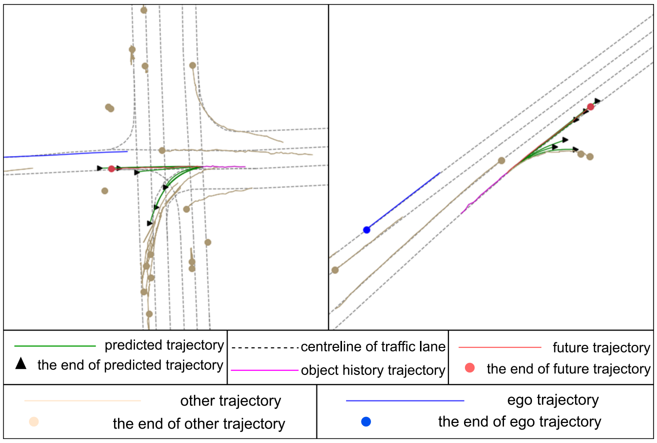
predicted trajectory
▲ the end of predicted trajectory
centreline of traffic lane
object history trajectory
future trajectory
● the end of future trajectory
other trajectory
● the end of other trajectory
ego trajectory
● the end of ego trajectory

Fig. 5. Picture shows an interesting result, which we have specially selected. Primarily, it illustrates the capacity of our model to leverage information from the surrounding agents. In this depiction, various elements are color-coded for clarity: blue is the ego vehicle, light brown is the neighboring traffic participant, magenta is the historical trajectory, red is the true trajectory, green is the predicted trajectory, and the dotted line is the center line of the lane line. With the exception of the predicted vehicle, all agents drew only the first two seconds of trajectory.

TABLE I
PERFORMANCE EVALUATION OF SEVERAL MAP-BASED PREDICTION METHODS ON THE ARGOVERSE TEST SET

Model	Source	Submission	minADE	minFDE	MR(%)
GOHOME [17]	ICRA	2022	0.943	1.45	**10.5**
HOME [15]	ITSC	2021	0.920	**1.36**	11.3
LaneRCNN [50]	IROS	2021	0.904	1.45	12.3
TNT [48]	CoRL	2020	0.910	1.45	16.5
DenseTNT [49]	ICCV	2021	0.882	1.28	12.6
TR-Pred	-	2023	**0.879**	1.50	19.5

Quantitative results on the Argoverse Motion Forecasting Leaderboard.

TABLE II
PERFORMANCE EVALUATION OF SEVERAL MAP-BASED PREDICTION METHODS ON THE ARGOVERSE VALIDATION SET

Model	Source	Submission	minADE	minFDE	MR(%)
TNT [48]	CoRL	2020	0.910	1.29	9.0
LaneRCNN [50]	IROS	2021	0.904	1.19	8.2
DenseTNT [49]	ICCV	2021	0.80	1.27	**7.0**
PiH [47]	TIV	2023	**0.70**	1.20	11
TR-Pred	-	2023	**0.70**	**1.14**	12.2

The validation scores reported in the referenced paper are adopted in the table.

TABLE III
PERFORMANCE EVALUATION OF SEVERAL MAP-FREE PREDICTION METHODS ON THE ARGOVERSE TEST SET

Model	Source	Submission	minADE	minFDE	MR(%)
NN [3]	CVPR	2019	1.71	3.3	53.7
MATF-GAN* [42]	CVPR	2019	1.35	2.48	-
EBMP* [46]	-	2023	1.26	2.27	-
CAM* [43]	ECCV	2020	1.13	2.50	-
MEATP* [44]	TCPS	2023	1.13	2.07	-
CRAT† [32]	ICRA	2022	1.05	1.87	25.8
F^{2}net* [45]	-	2023	0.93	1.59	21.4
TR-Pred	-	2023	**0.879**	**1.498**	**19.5**

(1) Quantitative results on the Argoverse Motion Forecasting Leaderboard.
(2) * indicates no test results on the Leaderboard are provided; we use test results from the paper instead.
(3) † denotes no Leaderboard test results are given, but we train the model using open-source code. Our results align closely with those state in the paper.

TABLE IV
PERFORMANCE EVALUATION OF SEVERAL MAP-FREE PREDICTION METHODS ON THE HIGHD TEST SET

Model	Source	Submission	minADE	minFDE	MR(%)
CAM [43]	ECCV	2020	0.37	0.74	5.12
CRAT [32]	ICRA	2022	0.23	0.45	2.24
LaneGCN‡ [9]	ECCV	2020	0.17	0.35	0.58
HiVT‡ [33]	CVPR	2022	0.16	0.32	0.60
TR-Pred	-	2023	**0.13**	**0.26**	**0.45**

‡ denotes a map-based model that can perform map-free predictions by excluding the map module.

- Learning Lane Graph Representations for Motion Forecasting (LaneGCN) [9].
- Efficient Baselines for Motion Prediction in Autonomous Driving (EBMP) [46].
- HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Predictions (HiVT) [33].

Additionally, our proposed model is also compared with the following map-based prediction methods:

- Planning-Inspired Hierarchical Trajectory Prediction Via Lateral-Longitudinal Decomposition for Autonomous Driving (PiH) [47].
- Heatmap Output for future Motion Estimation (HOME) [15].
- Graph-Heatmap Output for future Motion Estimation (GOHOME) [17].
- Target-driven Trajectory Prediction (TNT) [48].
- End-to-end Trajectory Prediction from Dense Goal Sets (DenseTNT) [49].
- Distributed Representations for Graph-Centric Motion Forecasting (LaneRCNN) [50].

The results in Tables I and III are from the 15 July 2023 Argoverse leaderboard. Our results are submitted on 7 July 2023.

Compared to conventional map-based approaches, our model attains superior performance on the minADE metric. Nevertheless, it exhibits suboptimal results on the minFDE and MR metrics. In light of this phenomenon, we conduct an analysis. Regarding the role of maps in trajectory prediction, we put forward the following hypothesis: In most cases, agents are traveling near the centerline of the road, except for situations such as lane changing or lane shifting. HD maps impose constraints on the agent's localization, enhancing the precision of the agent regarding the horizontal position and ameliorating errors. Moreover, map cues can help enhance the model's understanding of the agent's intention to turn and change lanes. However, our model has some advantages over other map-based methods regarding the quality of the generated trajectories. These results indicate that our model exhibits comparable performance to other map-based approaches in inferring the agent's intent. Our model can acquire the agent's intent through limited information and has stronger processing capabilities for the agent's trajectory information. Specific results are shown in Tables I and II.

Compared to recent map-free prediction methods, our proposed model achieves SOTA performance and surpasses other methods on minADE, minFDE, and MR metrics. Specific results are shown in Table III. Compared to the recently proposed method CRAT, our model demonstrates improvements of 0.171, 0.369, and 6.3 on the minADE, minFDE, and MR metrics, respectively. In particular, relative to F ^{2} net, another concurrently proposed map-free prediction method, our model achieves respective improvements of 5.5%, 5.8%, and 8.9% on the minADE, minFDE, and MR metrics.

For the experiments on the highD and rounD datasets, the models used for comparison are modified from their official git to allow training on these datasets. Our experiments results show that our model achieves the best performance on highway scene (highD dataset) and roundabout scene (RounD dataset). As shown in Table IV, our model achieves nearly zero loss rate on the highD dataset, with minFDE less than 0.25 m. Compared to the best performing map-free version of HIVT, our model

TABLE V
PERFORMANCE EVALUATION OF SEVERAL MAP-FREE PREDICTION METHODS ON THE ROUND TEST SET

Model	Source	Submission	minADE	minFDE	MR(%)
CAM [43]	ECCV	2020	0.68	1.51	23.77
CRAT [32]	ICRA	2022	0.52	1.20	14.43
LaneGCN‡ [9]	ECCV	2020	0.28	0.69	2.64
HiVT‡ [33]	CVPR	2022	0.24	0.59	2.82
TR-Pred	-	2023	**0.22**	**0.54**	**2.61**

TABLE VI
PERFORMANCE EVALUATION OF SEVERAL MAP-FREE PREDICTION METHODS ON THE ARGOVERSE VALIDATION SET

Model	Source	Submission	minADE	minFDE	MR(%)
CRAT* [32]	ICRA	2022	0.85	1.44	17.3
LaneGCN‡ [9]	-	2020	0.79	1.29	-
HiVT‡ [33]	CVPR	2022	0.77	1.25	14
Liannet* [27]	TIV	2023	0.74	1.16	12
F^2net* [45]	-	2023	0.74	1.18	**11.7**
TR-Pred	-	2023	**0.70**	**1.14**	12.2

improves minADE and minFDE by nearly 20%. On the more complex traffic conditions of the RounD dataset, our model achieves the best performance compared to other models on all three metrics of minADE, minFDE, and MR. Compared to the well-performing HIVT model, our model improves these three metrics by 8.3%, 8.5%, and 6.7% respectively. The performance of each model on the rounD dataset is presented in Table V.

3) Compare the Results of the Validation Set With the Results of the Test Set: During the experiments, we observe some interesting phenomena that indicate that our model has a stronger learning ability. We compare the results of the map-free prediction method on the validation set. Our model achieves the best results in terms of minADE and minFDE metrics. However, in the MR metric, our model performs even worse than F ^{2} net. Specific data is shown in Table VI. These results diverge from those attained on the test set. The problem of performance degradation of our model is minimal. For this reason, we specifically look at the difference between the test set, validation set, and training set. We find that scenarios overlap in the test set training set much more. The test has many different or even new scenarios. Most of the models suffer from more or less overfitting, resulting in models that cannot extract valid information when confronted with new scenarios. Our model has better generalization and can be used to extract more usable information from the data. More helpful information can be extracted when confronted with unfamiliar scenarios. This is due to our effective model design, which allows us to extract the features of the scene better. Despite utilizing both input modalities similar to F ^{2} Net, our two-stream architecture enables superior generalization capabilities and performance metrics. The specific difference between the validation set and test set data is shown in Table VII.

4) Inference Speed: The computational complexity of our model is approximately 0.139 GFLOPs. It is comparable to LaneGCN and about 25% of DenseTNT's complexity [51]. We evaluate the inference speed of the model on a part of the Argoverse validation set using an RTX 3090 GPU. The inference speed of our model for single-agent trajectory prediction tasks is comparable to that of conventional models. When predicting trajectories for multi-agents, the inference speed is much faster than models that cannot inference multi-agents trajectories at the same time. The inference speeds are shown in Table VIII.

TABLE VII
DIFFERENCE BETWEEN VALIDATION SET AND TEST SET

Model	Source	Submission	minADE	minFDE	MR(%)
CRAT [32]	ICRA	2022	0.20	0.43	8.5
F^2net [45]	-	2023	0.19	0.41	9.7
TR-Pred	-	2023	**0.18**	**0.36**	**7.3**

TABLE VIII
INFERENCE SPEED ON VALIDATION SET

Model	Speed(ms)		Input	minADE
	N = 1	N = 2		
DenseTNT [49]	137	274	Map+Traj.	0.73
LaneGCN [9]	**38**	78	Map+Traj.	0.79
HiVT‡ [33]	45	45	Traj.	0.73
CRAT [32]	40	80	Traj.	0.85
TR-Pred	44	**44**	Traj.	**0.70**

N: The number of objects requiring prediction.

TABLE IX
ABLATION EXPERIMENTS ON THE ARGOVERSE VALIDATION SET

LSTM	TI	LE	TT	RI	minADE	minFDE	MR
✓	✓				0.76	1.24	0.139
✓	✓	✓		✓	0.75	1.21	0.133
✓	✓	✓	✓		0.74	1.2	0.13
		✓	✓	✓	0.73	1.16	0.126
✓		✓	✓	✓	0.72	1.16	0.124
✓	✓	✓	✓	✓	**0.70**	**1.14**	**0.122**

TI: Tarj-Interaction; LE: Local Encoder; RI: Rel-Interaction; TT: temporal transformer

C. Ablation Studies

All ablation experiments are validated at Argoverse. Unless specified, the class token is initialized through the LSTM output in the temporal transformer. In this subsection, we conduct detailed ablation experiments to validate the effectiveness of each module. It is shown in Table IX.

1) Trajectory Stream: In the experiments with the trajectory stream, our relative stream uses the whole framework for embodying the adaptation of the model to the relative stream. For trajectory stream's ablation experiments, we investigated the effects of LSTM Encode and Trajectory-Interactions on model performance separately. In the case of the relative stream, LSTM can be a further performance improvement for the framework. The Global Information Interaction Module can further improve

TABLE X
ABLATION EXPERIMENTS FOR TRAJECTORY STREAM ON THE ARGOVERSE VALIDATION SET

LSRM Encoder	Traj-Int	minADE	minFDE	MR
		0.73	1.16	0.126
✓		0.72	1.16	0.124
✓*	✓*	0.76	1.24	0.139
✓	✓	**0.70**	**1.14**	**0.122**

* indicates no use of the relative stream.

TABLE XI
ABLATION EXPERIMENTS FOR RELATIVE STREAM ON THE ARGOVERSE VALIDATION SET

local Encoder	TT	RI	minADE	minFDE	MR
✓		✓	0.75	1.21	0.133
✓	✓		0.74	1.2	0.13
✓	✓	✓	**0.70**	**1.14**	**0.122**

TT: temporal transformer; RI: Rel-Interaction

the model's performance because it focuses on vehicle interaction information in a larger region. It is shown in Table X.

2) Relative Stream: In our experiments with relative stream, our trajectory stream uses to represent the adaptation of the model to the trajectory stream. We demonstrate the contribution of each module to the prediction performance by alternately removing one of the components. If maxout is used instead of the temporal transformer, we can find that the performance of the model decreases very seriously. This is because Temporal Transform can fully incorporate the long-distance trajectory information to fuse the information of the surrounding environment at each moment. Like the trajectory stream, removing the global interaction module leads to decreased model performance. This is due to the global interaction module's superior ability to capture relationships between different agents. It is shown in Table XI.

3) Relative Stream and Trajectory Stream: In this paragraph, our relative stream and trajectory stream utilize a full-framework model. It can be observed that a single stream exhibits adequate performance. The model's performance is also enhanced after the fusion between the two streams. This phenomenon demonstrates that the framework's performance can be fully augmented by sharing information between different modalities. It also provides evidence for the validity of representing relative motion temporally and prolonged motion trajectory. The specific performance of the model is shown in Table XII.

4) Class-Token Initialization: In this paragraph, we evaluate the impact of using two different class token initialization methods on model performance. We find that the method utilizing LSTM initialization achieved superior performance. Consequently, we posit the following conjecture. Learnable parameters can provide some a priori knowledge, as in Detr [39], and extract more appropriate features. Therefore, this form requires extensive training iterations and stacked Transformer modules to realize its full potential. With limited transformer stacks and epochs in our model, LSTM initialization achieves superior performance by reducing these requirements. The performance of the class-token initialization is shown in Table XIII.

TABLE XII
ABLATION EXPERIMENTS FOR TWO-STREAM ON THE ARGOVERSE VALIDATION SET

Traj-Stream	Rel-Stream	minADE	minFDE	MR
✓		0.76	1.24	0.139
	✓	0.73	1.16	0.126
✓	✓	**0.70**	**1.14**	**0.122**

Traj-Stream: trajectory stream; Rel-Stream: relative stream

TABLE XIII
ABLATION EXPERIMENTS FOR CLASS-TOKEN ON THE ARGOVERSE VALIDATION SET

Learnable-Para	LSTM Init	minADE	minFDE	MR
✓		0.72	1.15	0.124
	✓	**0.70**	**1.14**	**0.122**

Learnable-Para: Learnable parameters

5) Summaries: This subsection presents ablation studies to assess the contribution of each module and its influence on the holistic model performance. The results also demonstrate the effectiveness and rationality of our modifications to pertinent model components. The performance of all ablation experiments is shown in Table IX.

VI. CONCLUSION

This paper proposes a two-stream map-free trajectory prediction network that achieves excellent prediction results. The core idea of the framework is to correlate context information through two different modal representations of the same information. Although the two modes of agent trajectory can be independently predicted to achieve acceptable performance, the collective use significantly enhances the overall effectiveness, showing strong complementarity. In each stream, we use a message encoding followed by a global interaction for contextual messages. The extensive ablation studies further validate the efficacy of the proposed solutions in enhancing predictive performance. The context interaction module adopts a graph transformer. In the Argoverse dataset, map-free prediction methods achieve for the first time within 20% (19.5%) in MR and within 0.9 (0.88) in minADE. This result paves the way for potential applications of map-free prediction methods. In the highway and roundabout traffic scenarios, our model also exhibits good performance. Future research can proceed in the following directions: (1) knowledge distillation can be utilized to acquire enhanced constraint information; (2) incorporating interactive motions between agents during prediction may improve prediction quality.

REFERENCES

[1] 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.

[2] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, “Deep learning-based vehicle behavior prediction for autonomous driving applications: A review,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 33–47, Jan. 2022.

[3] M.-F. Chang et al., “Argoverse: 3D tracking and forecasting with rich maps,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8740–8749.

[4] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction,” in *Proc. Conf. Robot Learn.*, 2020, pp. 86–99.

[5] 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.*, 2021, pp. 7573–7582.

[6] Z. Zhou, J. Wang, Y. Li, and Y. Huang, “Query-centric trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17863–17873.

[7] M. Ye, T. Cao, and Q. Chen, “TPCN: Temporal point cloud networks for motion forecasting,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11313–11322.

[8] 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.*, 2023, pp. 2980–2987.

[9] M. Liang et al., “Learning lane graph representations for motion forecasting,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 541–556.

[10] J. Gao et al., “VectorNet: Encoding HD maps and agent dynamics from vectorized representation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11522–11530.

[11] G. Elghazaly, R. Frank, S. Harvey, and S. Safko, “High-definition maps: Comprehensive survey, challenges and future perspectives,” *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 527–550, 2023.

[12] L. Hou, S. E. Li, B. Yang, Z. Wang, and K. Nakano, “Integrated graphical representation of highway scenarios to improve trajectory prediction of surrounding vehicles,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1638–1651, Feb. 2023.

[13] J. Hong, B. Sapp, and J. Philbin, “Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8446–8454.

[14] L. Li, X. Wang, D. Yang, Y. Ju, Z. Zhang, and J. Lian, “Real-time heterogeneous road-agents trajectory prediction using hierarchical convolutional networks and multi-task learning,” *IEEE Trans. Intell. Veh.*, early access, May 11, 2023, doi: 10.1109/TIV.2023.3275164.

[15] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, “Home: Heatmap output for future motion estimation,” in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 500–507.

[16] Y. Sun, T. Xu, J. Li, Y. Chu, and X. Ji, “MMH-STA: A macro-micro-hierarchical spatio-temporal attention method for multi-agent trajectory prediction in unsignalized roundabouts,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11237–11250, Sep. 2023.

[17] 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.*, 2022, pp. 9107–9114.

[18] Y. Wang, J. Wang, J. Jiang, S. Xu, and J. Wang, “SA-LSTM: A trajectory prediction model for complex off-road multi-agent systems considering situation awareness based on risk field,” *IEEE Trans. Veh. Technol.*, early access, Jun. 20, 2023, doi: 10.1109/TVT.2023.3287227.

[19] Q. Meng, H. Guo, Y. Liu, H. Chen, and D. Cao, “Trajectory prediction for automated vehicles on roads with lanes partially covered by ice or snow,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 6972–6986, Jun. 2023.

[20] Y. Cai et al., “Environment-attention network for vehicle trajectory prediction,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11216–11227, Nov. 2021.

[21] X. Chen, H. Zhang, Y. Hu, J. Liang, and H. Wang, “VNAGT: Variational non-autoregressive graph transformer network for multi-agent trajectory prediction,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 10, pp. 12540–12552, Oct. 2023.

[22] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, “Attention based vehicle trajectory prediction,” *IEEE Trans. Intell. Veh.*, vol. 6, no. 1, pp. 175–185, Mar. 2021.

[23] K. Zhang, L. Zhao, C. Dong, L. Wu, and L. Zheng, “AI-TP: Attention-based interaction-aware trajectory prediction for autonomous driving,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 73–83, Jan. 2023.

[24] D. Xu, X. Shang, Y. Liu, H. Peng, and H. Li, “Group vehicle trajectory prediction with global spatio-temporal graph,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1219–1229, Feb. 2023.

[25] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, “MTP-GO: Graph-based probabilistic multi-agent trajectory prediction with neural ODEs,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 9, pp. 4223–4236, Sep. 2023.

[26] B. A. team, “Apollo: Open source autonomous driving,” 2017. [Online] Available: https://github.com/ApolloAuto/apollo

[27] J. lian, S. Li, D. Yang, J. Zhang, and L. Li, “Encoding the intrinsic interaction information for vehicle trajectory prediction,” *IEEE Trans. Intell. Veh.*, early access, Jun. 26, 2023, doi: 10.1109/TIV.2023.3288976.

[28] Z. Wang, J. Guo, H. Zhang, R. Wan, J. Zhang, and J. Pu, “Bridging the gap: Improving domain generalization in trajectory prediction,” *IEEE Trans. Intell. Veh.*, early access, Jul. 28, 2023, doi: 10.1109/TIV.2023.3299600.

[29] Y. Han, Q. Liu, H. Liu, B. Wang, Z. Zang, and H. Chen, “TP-FRL: An efficient and adaptive trajectory prediction method based on the rule and learning-based frameworks fusion,” *IEEE Trans. Intell. Veh.*, early access, May 25, 2023, doi: 10.1109/TIV.2023.3279825.

[30] X. Gao, X. Jia, Y. Li, and H. Xiong, “Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 5, pp. 2946–2953, May 2023.

[31] Z. Zhou, Z. Wen, J. Wang, Y.-H. Li, and Y.-K. Huang, “QCNeXt: A next-generation framework for joint multi-agent trajectory prediction,” 2023, *arXiv:2306.10508*.

[32] J. Schmidt, J. Jordan, F. Gritschneder, and K. Dietmayer, “CRAT-Pred: Vehicle trajectory prediction with crystal graph convolutional neural networks and multi-head self-attention,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 7799–7805.

[33] 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.*, 2022, pp. 8813–8823.

[34] X. Zhou, W. Zhao, A. Wang, C. Wang, and S. Zheng, “Spatiotemporal attention-based pedestrian trajectory prediction considering traffic-actor interaction,” *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 297–311, Jan. 2023.

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

[36] S. Ettinger et al., “Large scale interactive motion forecasting for autonomous driving : The waymo open motion dataset,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9690–9699.

[37] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, “Recipe for a general, powerful, scalable graph transformer,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 14501–14515.

[38] Y. Chai, S. Jin, and X. Hou, “Highway transformer: Self-gating enhanced self-attentive networks,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6887–6900.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[40] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, “The highd dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems,” in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst.*, 2018, pp. 2118–2125.

[41] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, “The round dataset: A drone dataset of road user trajectories at roundabouts in Germany,” in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–6.

[42] T. Zhao et al., “Multi-agent tensor fusion for contextual trajectory prediction,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12118–12126.

[43] S. H. Park et al., “Diverse and admissible trajectory forecasting through multimodal context understanding,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.

[44] J. Wang, L. Su, S. Han, D. Song, and F. Miao, “Towards safe autonomy in hybrid traffic: Detecting unpredictable abnormal behaviors of human drivers via information sharing,” *ACM Trans. Cyber- Phys. Syst.*, early access, Sep. 22, 2023, doi: 10.1145/3616398.

[45] J. Xiang, J. Zhang, and Z. Nan, “A fast and map-free model for trajectory prediction in traffics,” 2023, *arXiv:2307.09831*.

[46] C. Gómez-Huélamo, M. V. Conde, R. Barea, M. Ocaña, and L. M. Bergasa, “Efficient baselines for motion prediction in autonomous driving,” 2023, *arXiv:2309.03387*.

[47] D. Li et al., “Planning-inspired hierarchical trajectory prediction via lateral-longitudinal decomposition for autonomous driving,” *IEEE Trans. Intell. Veh.*, early access, Aug. 22, 2023, doi: 10.1109/TIV.2023.3307116.

[48] H. Zhaoet al., “TNT: Target-driven trajectory prediction,” in *Proc. Conf. Robot Learn.*, 2021, pp. 895–904.

[49] 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.*, 2021, pp. 15303–15312.

[50] W. Zeng, M. Liang, R. Liao, and R. Urtasun, “LaneRCNN: Distributed representations for graph-centric motion forecasting,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 532–539.

[51] 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.

Zhongning Wang received the B.Sc. degree in industrial design (body engineering) in 2019 from Jilin University, Changchun, China, where he is currently working toward the M.Sc. degree in vehicle engineering. His research interests include multi-information fusion, trajectory prediction and real-time trajectory planning.

Jianwei Zhang received the B.Sc. degree in vehicle engineering from the Harbin Institute of Technology, Harbin, China, in 1996, and the M.Sc. and Ph.D. degrees in vehicle engineering from Jilin University, Changchun, China, in 1999 and 2003, respectively. He is currently an Associate Professor with the State Key Laboratory of Automotive Simulation and Control, Jilin University. His research interests include vehicle dynamic control, autonomous vehicle control, and driving motor control.

Jicheng Chen (Member, IEEE) received the B.Sc. and M.Sc. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree from the Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada, in 2021. He is currently a Postdoctoral Researcher with Beihang University, Beijing, China. His main research interests include robust control, stochastic model predictive control, and event-based control.

Hui Zhang (Senior Member, IEEE) received the B.Sc. degree in mechanical design manufacturing and automation from the Harbin Institute of Technology, Weihai, China, in 2006, the M.Sc. degree in automotive engineering from Jilin University, Changchun, China, in 2008, and the Ph.D. degree in mechanical engineering from the University of Victoria, Victoria, BC, Canada, in 2012. He was the recipient of the 2017 IEEE Transactions on Fuzzy Systems Outstanding Paper Award, the 2018 SAE Ralph R. Teetor Educational Award, the IEEE Vehicular Technology Society 2019 Best Vehicular Electronics Paper Award, and the 2019 SAE International Intelligent and Connected Vehicles Symposium Best Paper Award. He is a Member of SAE International and ASME.