Multi-Agent Trajectory Prediction with Difficulty-Guided Feature Enhancement Network

Guipeng Xin, Duanfeng Chu*, *Member, IEEE*, Liping Lu, Zejian Deng, Yuang Lu and Xigang Wu

***Abstract*—Trajectory prediction is crucial for autonomous driving as it aims to forecast the future movements of traffic participants. Traditional methods usually perform holistic inference on the trajectories of agents, neglecting the differences in prediction difficulty among agents. This paper proposes a novel Difficulty-Guided Feature Enhancement Network (DGFNet), which leverages the prediction difficulty differences among agents for multi-agent trajectory prediction. Firstly, we employ spatio-temporal feature encoding and interaction to capture rich spatio-temporal features. Secondly, a difficulty-guided decoder is used to control the flow of future trajectories into subsequent modules, obtaining reliable future trajectories. Then, feature interaction and fusion are performed through the future feature interaction module. Finally, the fused agent features are fed into the final predictor to generate the predicted trajectory distributions for multiple participants. Experimental results demonstrate that our DGFNet achieves state-of-the-art performance on the Argoverse 1&2 motion forecasting benchmarks. Ablation studies further validate the effectiveness of each module. Moreover, compared with SOTA methods, our method balances trajectory prediction accuracy and real-time inference speed. The code is available at https://github.com/XinGP/DGFNet.**

***Index Terms*—Autonomous driving, Trajectory Prediction, Behavioral Heterogeneity, Feature Enhancement Network.**

I. Introduction

TRAJECTORY prediction is a crucial component of current autonomous driving systems. Its goal is to infer the future trajectory distribution of agents based on historical information within the scene. In addition to the historical motion information of the agents, various complex factors must be considered, such as lane markings and traffic light constraints in the map, as well as social interactions between agents. These considerations ensure that the future trajectory distribution can accurately reflect the agents' movement trends, thereby ensuring the proper functioning of the autonomous driving system.

*This work is supported in part by the National Key R&D Program of China (2021YFB2501104), the Natural Science Foundation of Hubei Province for Distinguished Young Scholars (2022CFA091), and Wuhan Science and Technology Major Project (2022013702025184). (*Corresponding author: Duanfeng Chu.*)

Guipeng Xin, Duanfeng Chu, Yuang Lu and Xigang Wu are with the Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China (e-mail: xinguipeng@whut.edu.cn; chudf@whut.edu.cn; luyuang@whut.edu.cn; wxg_whut_1701@whut.edu.cn).

Liping Lu is with the School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, China (e-mail: luliping@whut.edu.cn).

Zejian Deng is with the Mechatronic Vehicle Systems Lab at the University of Waterloo,Waterloo, ON N2L3G1, Canada(e-mail: z49deng@uwaterloo.ca).

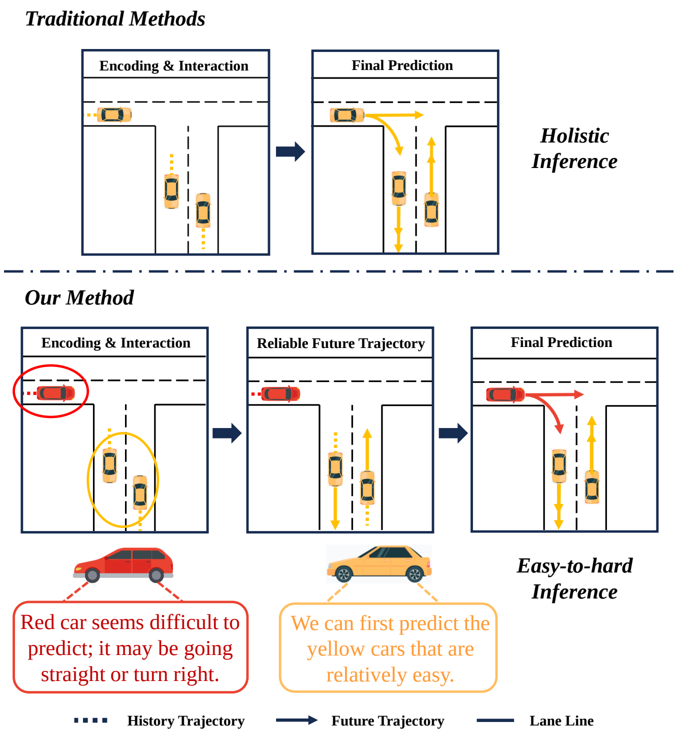


Fig. 1. In contrast to traditional trajectory prediction methods, we have incorporated an intermediate step to obtain reliable future trajectories. The lower part of the figure illustrates the varying prediction difficulty among different vehicles in a sample traffic scenario. The future driving trajectory of the yellow vehicle is relatively easy to predict.

Currently, mainstream prediction model architectures can be roughly divided into three steps: feature encoding, feature interaction, and feature decoding. Additionally, some methods improve the accuracy and robustness of prediction models through feature enhancement. These methods involve intermediate modeling of future intentions or trajectories and interact and fuse these with the original encoded features. Although these methods significantly improve model performance compared to mainstream model architectures, they fail to consider the inherent heterogeneity in prediction difficulty among agents, i.e., the natural differences in prediction difficulty for different agents within the scene.

To address potential hazards in driving scenarios, during the decision-making and planning stages of autonomous driving, it is common to consider the heterogeneity of traffic participants' behaviors [1]–[4]. This is primarily manifested in driver social behaviors, intent uncertainty, and scenarios characterized by long-tail distributions [1], [5], [6]. They categorize surrounding agents from different measurement perspectives to enhance the safety and comfort of decision-making and planning tasks. Recently, some studies have incorporated long-tail detection into trajectory prediction tasks [4], [7]. Specifically, they use methods such as contrastive learning or clustering to measure similarity, aiming to reduce the distance in feature space between samples that share similar scene characteristics. Compared to the difficulty of driving scenarios, our method explores the influence of prediction difficulty of different agents within a single scene on the network.

In real driving scenarios, human drivers often predict the future behaviors of most agents subconsciously, relying on intuition. However, when faced with agents that may pose risks and exhibit high prediction difficulty, human drivers tend to carefully consider their behavior and motivations [8]. Inspired by human drivers, we aim to investigate the effectiveness of prediction methods that follow the principle of easy-to-hard in multi-agent trajectory prediction tasks. To mimic the prediction approach of human drivers, our first step is to initially predict the potential intentions and future trajectories of all agents in the scene based on historical trajectories and map information. This step resembles the rough subconscious prediction of human drivers. Then, the prediction difficulty differentiation of agents is performed based on the difficulty coefficient modeling. Additionally, based on difficulty differentiation, we use reliable future trajectories as prior information and utilize feature interaction and fusion for feature enhancement. The significance of this step is that the future behaviors of easily predictable agents impose motion constraints on those agents with higher difficulty. Finally, to better capture local motion features and global relationships, we combine two approaches for scene representation.

As illustrated in the Fig. 1, traditional trajectory prediction methods can be broadly summarized into two stages: Encoding&Interaction and Final Prediction. Initially, contextual encoding is employed to extract scene features from various input information. These features are processed through an interaction module, which aids the model in understanding and capturing the interactions and relationships between different entities (such as vehicles, lane markings, traffic lights, etc.). Finally, these features are passed to a decoding module to obtain the final predicted trajectories and their corresponding probability distributions. The primary difference in our pipeline is the addition of an intermediate module that generates reliable future trajectories, thereby better prediction results can be achieved for agents with higher prediction difficulty.

The main contributions of this paper can be summarised as follows:

- We introduce a novel network architecture that combines the strengths of two scene representations and leverages the differences in prediction difficulty among agents for multi-agent trajectory prediction, thereby improving overall prediction accuracy.
- We propose a difficulty-guided decoder that adjusts the flow of future trajectories based on prediction difficulty, ensuring that reliable trajectories are utilized in subsequent prediction steps.
- Inspired by human drivers, our method follows the principle of easy-to-hard. By initially predicting the easier-to-predict agents and using their reliable future trajectories for feature interaction and fusion, we enhance the prediction results for agents with higher prediction difficulty.

The rest of this paper is organized as follows: We briefly review the related work in Section II. We present the technical details of our proposed method in Section III. Then, extensive experiments and analysis are presented in Section IV. Finally, we conclude the paper in Section V.

II. Related Work

Heterogeneity in Trajectory Prediction. The inherent heterogeneity in driving trajectories can provide rich additional information for decision making tasks, leading to safer and more flexible decision outcomes. Recent research in trajectory prediction tasks has taken such heterogeneity into account to provide personalised predictions. For example, Chen et al. [9] proposed a distribution discrimination method to predict personalised motion patterns by distinguishing latent distributions. Xing et al. [10] introduced a joint time-series modelling approach that considers different driving styles to predict the trajectory of lead vehicles. In contrast to the emphasis on heterogeneity in the aforementioned works, our research focuses primarily on the prediction difficulty of the agents. For example, Makansi et al. [4] addressed the problem of long-tailed data distribution by exploiting the effects of feature embeddings. The use of contrastive learning in feature embeddings aggregates rare and difficult samples, contributing to improved final prediction performance. Qian et al. [11] introduced a teacher learning strategy based on the difficulty of scene prediction. They predefined the prediction difficulty of the scene and trained the proposed model by first training on easier samples and then on more difficult samples. However, both methods did not consider potential differences in difficulty between agents in multi-agent prediction scenarios. In this work, we designed a difficulty estimation module to estimate potential prediction difficulties in multi-agent prediction scenarios. The future trajectories of agents with lower prediction difficulties are integrated into the scene to enhance information.

Future Feature Enhancement. Compared to conventional pipelines, methods that enhance features using future information allow networks to incorporate future constraints and interactions. These methods can be broadly categorized into two approaches: intention and future trajectories. TNT [12] and PECNet [13] model vehicle intention as latent variables and generate different prediction trajectories conditioned on vehicle intentions, forming predicted trajectories under certain constraint conditions. GANet [14] and ADAPT [15] integrate and fuse vehicle intentions for future feature enhancement through feature interaction. Similarly, Prophnet [16] and FFINet [17] model intermediate latent variables in the form of trajectories, where the output of intermediate decoders is one or multiple trajectories. Additionally, there is a method that models future trajectories through planner sampling, followed by feature enhancement [18], [19].

Most relevant to our work is M2I [20], which exploits potential game relationships between vehicles. It uses a Relation Predictor and a Marginal Trajectory Predictor to input the future trajectories of the leader in the game relationship as prior information into the Conditional Trajectory Predictor. However, we find that this method is highly dependent on the accuracy of the inferred game relationships and future trajectories. Accurately identifying game relationships between two vehicles in long-term predictions proves to be very challenging. Therefore, we use the more easily extractable prediction difficulty as a basis for personalized classification. Our approach is an end-to-end inference model, abandoning the multi-model inference method used by M2I.

Scene Representation. Inspired by the application of convolutional neural networks in image processing, early trajectory prediction methods used rasterized images as the main form of scene representation and then extracted 2D image features through convolution [21]. However, this approach required extensive image rendering processing and suffered from occlusion issues between trajectories and map scenes. To address this, Vectornet [22] introduced a unified scene representation by vectorizing trajectories and map scenes, and utilized MLP to extract scene information.

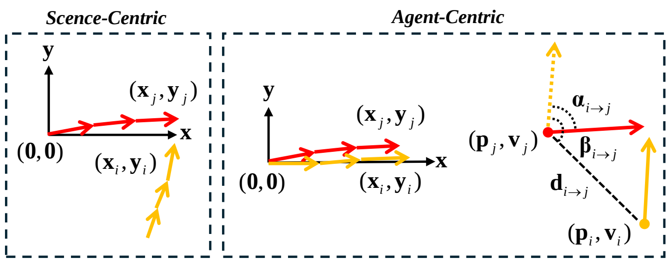
Broadly, vectorization can be categorized into two types: scene-centric and agent-centric. Since the pioneer of vectorization, Vectornet, was designed for single-agent trajectory prediction tasks, it naturally took the last frame coordinates of the vehicle to be predicted as the scene center, and expressed the trajectories and map vector coordinates of all other agents around the scene center. The benefit of this approach is that it enables the preservation of global relative pose features of various objects in the scene during feature extraction, and facilitates the embedding of some prior information or trajectory anchors [12], [16].

However, due to the demands of multi-agent trajectory prediction tasks, there has been a shift towards agent-centric scene representation methods. Building upon agent-centric approaches, some methods aim to preserve as much information as possible about the mutual spatial relationships of various objects in the scene [23]–[26]. To achieve this, they incorporate local pairwise-relative pose features and extract and fuse the relative pose information through networks such as GNNs [27] and Transformers [28]. However, such methods suffer from the limitation that historical and predicted trajectories only retain local motion characteristics, making it inconvenient to embed anchor and prior information. Therefore, we aim to preserve the advantages of each method by integrating the two approaches.

III. Method

A. Overview

Recent works [10], [11] have been dedicated to associating similar scenes through contrastive learning or clustering, linking scenes based on their similarity, and enhancing features during inference according to the degree of scene similarity. We propose DGFNet, a network designed for multi-agent trajectory prediction using a Difficulty-Guided Feature Enhancement module for asynchronous interaction between historical trajectories and reliable future trajectories. As shown in Fig. 3, the architecture of DGFNet can be divided into three components: spatio-temporal feature encoding, feature interaction, and trajectory decoding.


Scence-Centric
Agent-Centric
y
x
(0,0)
$(\mathbf{x}_j,\mathbf{y}_j)$
$(\mathbf{x}_i,\mathbf{y}_i)$
y
x
(0,0)
$(\mathbf{x}_j,\mathbf{y}_j)$
$(\mathbf{x}_i,\mathbf{y}_i)$
$\boldsymbol{\alpha}_{i\to j}$
$(\mathbf{p}_j,\mathbf{v}_j)$
$\boldsymbol{\beta}_{i\to j}$
$\mathbf{d}_{i\to j}$
$(\mathbf{p}_i,\mathbf{v}_i)$
Fig. 2. For the same scenario, the left side represents the scene-centric approach, which only requires using coordinate points. On the right side, the agent-centric approach necessitates expressing through local coordinate points and pairwise-relative poses.

(1)Spatio-temporal Feature Extraction. This component extracts spatio-temporal features separately from agent-centric historical trajectory information and scene-centric information using both agent-centric and scene-centric approaches. **(2)Feature Interaction.** The difference with spatio-temporal feature extraction is that the feature extraction component is compatible for both scenarios. However, feature interaction is not. The feature interaction component is categorized into two types: multi-head attention module and multi-head attention module with edge features. The two interaction methods are respectively applicable to scene-centric and agent-centric scenario representations. **(3)Trajectory Decoder.** This component is divided into a difficulty-guided decoder and a final decoder. The difficulty-guided decoder operates during the intermediate process, while the final decoder outputs the final prediction results.

B. Problem formulation

Building on the majority of previous work, such as [1], [12], [22], [25], [29], we assume that upstream perceptual tasks can provide high-quality 2D trajectory tracking data for agents in a coordinate system. At the same time, localisation and mapping tasks can provide accurate self-vehicle positioning data and comprehensive HD map data for the urban environment. In other words, for j agents within a given scene, we obtain x and y positions, denoted as $X^{0:j}_{-t_h:0}$, corresponding to the time stamp horizon $\{-t_h, ..., 0, 1, ..., t_f\}$. The trajectory prediction task consists of predicting the future trajectories $Y^{0:j}_{1:t_f}$ of the agents by using the HD map information M (including road coordinates, topological links, traffic lights and other relevant information) within the given scene and the historical trajectories output by the tracking task.

In accordance with the encoding methodology employed by VectorNet and LaneGCN, we represent the trajectory informa-

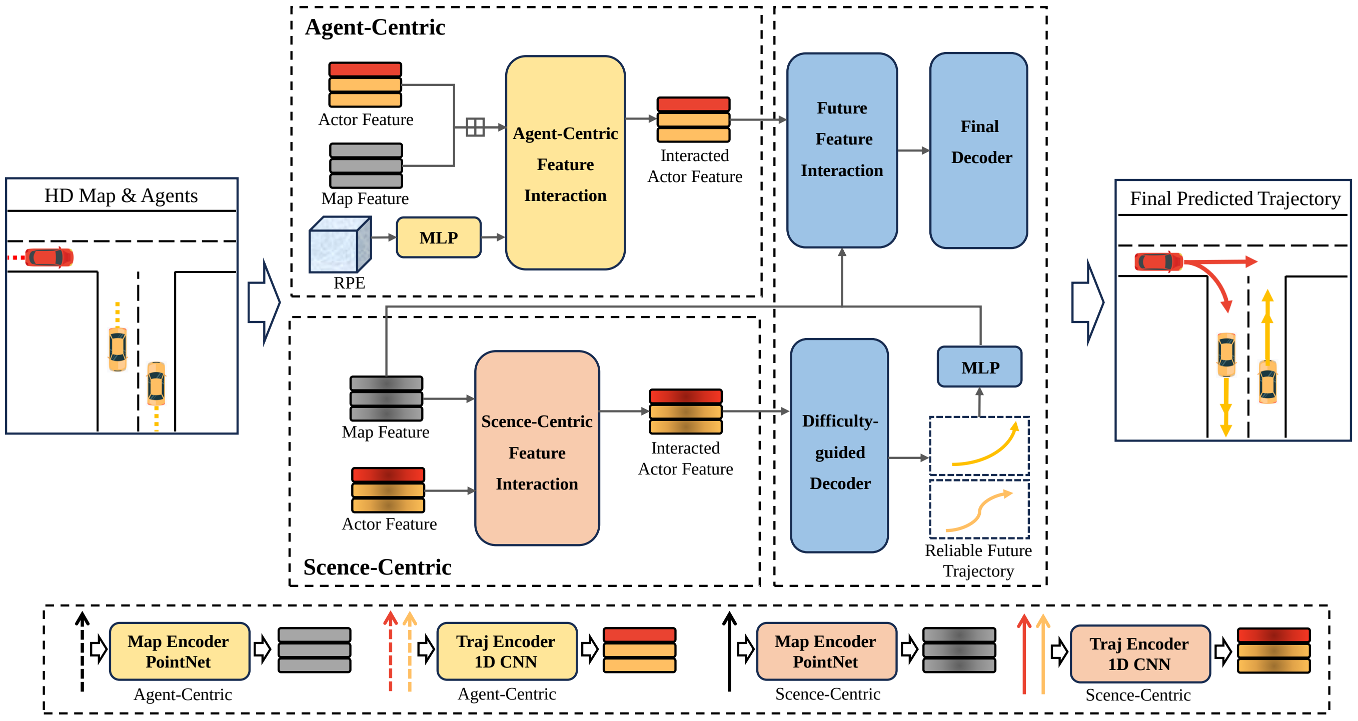


Fig. 3. Our spatio-temporal feature extraction includes two sets of independent encoders, which extract features based on scene representations (bottom). Subsequently, the extracted features pass through their respective feature interaction modules to obtain interacted Actor features. Following this, we perform future feature enhancement by interacting reliable future trajectories with Actor and Map features. Finally, these inputs are fed into the final predictor to obtain predicted trajectories and their corresponding probabilities.

tion of agents and lane details in vectorized form. To elaborate further, for a given agent i within the scenario, its trajectory information, denoted as X^i, is represented as a matrix $X^i_{-t_h:0}$ comprising a spatio-temporal sequence$\{s^i_{-t_h}, s^i_{-t_h+1}, ..., s^i_0\}$ over the past t_h time step. Similarly, map information is also segmented into predefined sequence vectors to capture various scene details. Analogous to trajectory vectors, map information is represented as $M_{1:N} = \{s_1, s_2, ..., s_N\}$, where N denotes the total vector length. Within each lane vector s_i, there is an inclusion of lane slice coordinates, lane type (e.g., straight or left-turn lane), and map lane details such as traffic signals. This encapsulates diverse map information pertaining to each lane.

It should be noted that this method involves two types of scenario representations. As shown in Fig. 2, the scene-centric representation involves only one coordinate system, where all agents and lane markings are expressed within this coordinate system. For simplicity, we only use uppercase notation to denote the trajectory and map tensor as X and M, respectively. In the agent-centric representation, multiple coordinate systems exist. Initially, all agents and lane markings are rotated using matrix rotation so that their orientations align with the x-axis. Subsequently, we use the original coordinates p and velocity orientation v to calculate the relative position distance $d_{i\to j}$ and the angular difference $\alpha_{i\to j}$, $\beta_{i\to j}$. We denote the agent-centric trajectory and map tensor as A and L, respectively. Pairwise-relative pose is represented as $RPE = \{||d_{i\to j}||, \sin(\alpha_{i\to j}), \cos(\alpha_{i\to j}), \sin(\beta_{i\to j}), \cos(\beta_{i\to j})\}$.

C. Spatio-temporal feature extraction

In the problem formulation, we explain the process of representing agent and high-definition (HD) map data as vectors, establishing a corresponding mapping between continuous trajectories, map annotations, and sets of vectors. This vectorization method allows us to encode trajectory and map information as vectors. Following the trajectory feature extraction method of LaneGCN, our trajectory feature extraction module mainly uses 1D CNN and downsampling techniques to encode the historical trajectories of all vehicles in the scene, thus obtaining encoded historical trajectory features. For map tensor, we use the PointNet [30] to encode lane nodes and structural information to obtain map features. It is worth mentioning that when processing scene-centric vector information, we do not mix vehicle historical trajectories and lane line information. Our view is that vehicle trajectories, as dynamic vectors, need to be distinguished from static maps to better capture the local motion characteristics of the vehicles. For the extraction of historical trajectory features are described by this:

$$Z_i = \textbf{Conv1d}(\textbf{Res1d}(X_i)) \quad \text{for } i = 0, 1, \ldots, n_{\text{fpn}} - 1 \tag{1}$$

$$\hat{X}_i = \textbf{Inter}(Z_{i+1}, \text{scale\_factor} = 2) + Z_i \tag{2}$$

after Res1d and Conv1d encoding, feature alignment is performed by scaling and interpolation operations. For the agent-centric trajectory tensor, the same feature extraction is used to obtain $\hat{A}$.

For map feature extraction is described by this:

$$H_1 = \mathbf{PAB}_1\left(\mathbf{ReLU}(\mathbf{LayerNorm}(MW_p + b_p))\right) \tag{3}$$

$$\hat{M} = \mathbf{PAB}_2(H_1) \tag{4}$$

where $M \in \mathbb{R}^{N_{\text{lane}} \times 10 \times C_{\text{in}}}$ are the input lane features, $W_p \in \mathbb{R}^{C_{\text{in}} \times h}$ and $\mathbf{b}_p \in \mathbb{R}^h$ are the weights and biases for the projection layer, and PAB represents the PointAggregateBlock.

Similarly, we can obtain the agent-centric map feature $\hat{L}$. For RPE we use MLP for feature extraction to get $\hat{RPE}$.

D. *Feature interaction*

Our intention was to utilize the global representation of scence-centric scenes to embed future trajectory features. Interestingly we found that even without any future enhancement, just concatenating the Actor features after the two interactions gives a better boost. Specific results can be seen later in the ablation experiments. Here we will present the two interaction methods: multi-head attention module and multi-head attention module with edge features, respectively.

We first introduce a generic feature interaction block. In contrast to the general method, which uses simple attention operations for feature interaction, we believe that multi-head attention mechanisms are often better suited to handle complex input sequences and capture a wider range of hierarchical and diverse information. We opt for Multi-Head Attention Blocks (MHAB) instead of the previously used simple attention operations, as used for the features [31]. Specifically, we use self-attention encoders and feedforward networks (FFN) for self-interaction and cross-attention encoders and FFN for cross-interaction:

$$\mathbf{MHA}(Q, K, V) = \mathbf{softmax}\left(\frac{QK^T}{\sqrt{dim_k}}\right)V \tag{5}$$

$$Q, K, V = W^q, W^k, W^v \tag{6}$$

where $\mathbf{MHA}$ denotes the Multi-Head Attention operation. Q, K, V are computed by linear projections W^q, W^k, W^v applied to input vectors. The attention mechanism computes scaled dot-product attention using Q and K, and applies it to V after softmax normalization.

The feature interaction module for carrying edge features is described as follows:

$$\mathbf{MHA}(Q', K', V') = \mathbf{softmax}\left(\frac{Q'K'^T}{\sqrt{dim_k}}\right)V' \tag{7}$$

$$Q', K', V' = Q + W^q\text{edge}', K + W^k\text{edge}', V + W^v\text{edge}' \tag{8}$$

here, Q', K', V' are the query, key, and value tensors after linear transformations that incorporate edge features, and edge$'$ is the edge feature after linear transformation.

Finally, the normalized output after attention and dropout application is:

$$\hat{x} = \mathbf{LayerNorm}(x + \mathbf{Drop}(\mathbf{MHA}(Q', K', V'))) \tag{9}$$

For ease of expression, we denote regular multi-head attention as $\mathbf{MHA}$ and multi-head attention with edge features as $\mathbf{MHA}_{edge}$. Our Scene-centric Feature Interaction can be denoted as follows:

$$\hat{X}^{(l)} = \mathbf{MHA}_{MA}(\hat{M}^{(l-1)}, \hat{X}^{(l-1)}) \tag{10}$$

$$\hat{M}^{(l)} = \mathbf{MHA}_{MM}(\hat{M}^{(l-1)}) \tag{11}$$

$$\hat{M}^{(l)} = \mathbf{MHA}_{AM}(\hat{X}^{(l)}, \hat{M}^{(l-1)}) \tag{12}$$

$$\hat{X}^{(l)} = \mathbf{MHA}_{AA}(\hat{X}^{(l-1)}) \tag{13}$$

Here, l denotes the current layer index, and $\hat{X}^{(0)} = \hat{X}$ and $\hat{M}^{(0)} = \hat{M}$ are the initial inputs.

Our Agent-centric Feature Interaction can be denoted as follows:

$$\hat{A}^{(k)} = \mathbf{MHA}_{edge}(\hat{A}^{(k-1)}, \hat{L}^{(k-1)}, \hat{RPE}^{(k-1)}) \tag{14}$$

Similarly we complete multiple rounds of interactions by looping through multiple levels of updates.

E. *Difficulty-guided decoder*

To take advantage of the different prediction difficulties of different agents in the scenario, we developed the Difficulty Guided Feature Enhancement Module, which consists of a Difficulty Guided Trajectory Decoder and a Future Feature Interaction Module.

To capture features of plausible future trajectories in the scene, we introduce a Difficulty-Guided Decoder to obtain the future trajectories of agents that are relatively easier to predict. Firstly, we decode the actor features $\hat{A}^{(k)}$ in Agent-centric, which have undergone multiple layers of feature interactions. Each agent receives six predicted trajectories. Following the ADAPT [15] trajectory decoder to get higher quality decoded trajectories by predicting endpoints for refinement. The decoding process can be expressed as follows:

$$\hat{E} = \mathbf{MLP}_{\text{end}}(\hat{X}^{(l)}) + \mathbf{MLP}_{\text{refine}}(\text{Cat}[\hat{X}^{(l)}, \hat{E}]) \tag{15}$$

$$\hat{P} = \text{Cat}[\mathbf{MLP}_{\text{traj}}(\text{Cat}[\hat{X}^{(l)}, \hat{E}]), \hat{E}] \tag{16}$$

Where endpoints $\hat{E}$ are predicted and refined by concatenating with the agent features $\hat{X}^{(l)}$, then the trajectories $\hat{P}$ are predicted and merged with the refined endpoints.

The Argoverse 1 validation and train sets show that the predicted trajectories of most of the vehicles in the scenarios exhibit a high degree of concentration, with more than 90% of the samples of straight ahead trajectories in each case [32]. This suggests that the motion patterns of most agents can be easily captured, reflecting real-world traffic scenarios. In order to prevent higher difficulty prediction trajectories from entering the subsequent modules and generating cumulative errors, we introduced a difficulty masker to filter the initial prediction results. ScenceTransformer [33], inspired by recent approaches to language modeling, has pioneered the idea of using a masking strategy as a query to the model, enabling it to invoke a single model to predict agent behavior in multiple ways (marginal or joint prediction). This mechanism in this

Algorithm 1 Difficulty Masker for Predicted Trajectories

```
Input: Predicted trajectories $\hat{P}$, Predicted endpoints $\hat{E}$, threshold $\tau$
Output: Masked trajectories $\hat{P}_{masked}$
1: $\hat{P}_{masked} = set()$
2: $\bar{E} \leftarrow \hat{E}.mean()$
3: for $i = 1$ to $N$ do
4:     $D = \|\hat{E} - \bar{E}\|_2$
5:     $\bar{D} \leftarrow D.mean()$
6:     if $\bar{D} \leq \tau$ then
7:         $\hat{P}_{masked}$.add($\hat{P}_l$)
8:
9: return $\hat{P}_{masked}$
```

paper is intended to mask out the initial prediction trajectories of the more difficult agents, thus providing better control over the prediction accuracy. This process can be formalized as Algo. 1.

F. Future feature enhancement

For the reliable future trajectories we first flatten them by performing the flatten operation on multiple future trajectories for feature dimension alignment. Then future trajectory feature extraction is performed by a simple MLP layer. These two operations can be formulated as:

$$\hat{F} = \mathbf{MLP}(\mathbf{flatten}(\hat{P}_{\text{masked}})) \tag{17}$$

Subsequently, we interact the future trajectory feature $\hat{F}$ with the agent-centric interactive actor feature $\hat{A}^{(k)}$ via multi-head attention. This operation aims to impose certain constraints on the prediction of other agents through the reliable future trajectory features. Finally, we need to perform one last interaction using the original map features $\hat{M}$. This step is crucial as it aims to refocus on the reachable lane lines of the map after imposing social constraints through the future trajectory features [34]. These two operations can be formulated as:

$$H = \mathbf{MHA}_{AF}(\hat{A}^{(k)}, \hat{F}) \tag{18}$$

$$\hat{O} = \mathbf{MHA}_{AM}(H, \hat{M}) \tag{19}$$

G. Final prediction

We perform feature fusion operations on the final features $\hat{O}$ and the scence-centric feature $\hat{X}^{(l)}$, expanding the decoding dimension to 256. Our final predictor can generate multimodal trajectories for all agents in the scene and their corresponding probabilities in a single inference. This process can be described as follows:

$$A_{fusion} = \text{Cat}(\hat{O}, \hat{X}^{(l)}) \tag{20}$$

$$E = \mathbf{MLP}_{\text{end}}(A_{fusion}) + \mathbf{MLP}_{\text{refine}}(\text{Cat}[A_{fusion}, E]) \tag{21}$$

$$K = \mathbf{softmax}(\mathbf{MLP}_{\text{cls}}(\text{Cat}[A_{fusion}, E])) \tag{22}$$

$$P = \text{Cat}[\mathbf{MLP}_{\text{traj}}(\text{cat}[A_{fusion}, E]), E] \tag{23}$$

Here, P represents the multimodal prediction trajectory and its corresponding probability K that we finally obtain.

Model Training. Although our feature enhancement component outputs trajectories during the intermediate process, our training process remains end-to-end. Our DGFNet obtains initial and final predictions in a single pass. Similar to the previous method [12], [22], [25], [29], we supervise the output trajectories P and $\hat{P}$ during the training process by smoothing the L1 loss, which is expressed as $\mathcal{L}_{reg}$ and $\mathcal{L}^{c}_{reg}$. For the categorization loss in K, we use the maximum marginal loss, which is expressed as $\mathcal{L}_{cls}$. Our overall loss expression is as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{reg} + \beta\mathcal{L}_{cls} + \lambda\mathcal{L}^{c}_{reg} \tag{24}$$

where α, β and λ are three constant weights.

IV. Experiments

A. Experimental setup

Datasets. Our model DGFNet is tested and evaluated on a very challenging and widely used self-driving motion prediction dataset: the Argoverse 1&2 [47], [48]. Both motion prediction datasets provide agent tracking trajectories and semantically rich map information at a frequency of 10Hz over a specified time interval. The prediction task in Argoverse 1 is to predict trajectories for the next 3 seconds based on the previous 2 seconds of historical data. The dataset contains a total of 324,557 vehicle trajectories of interest extracted from over 1000 hours of driving. In contrast, Argoverse 2 requires a higher level of robustness and generalization of the predictive model, with 250,000 of the most challenging scenarios officially filtered from the self-driving test fleet. Argoverse 2 predicts the next 6 seconds from the first 5 seconds of historical trajectory data, and Argoverse 2 provides heading data for traffic targets as well as traffic target categories, in order to ensure fair comparisons between models, the dataset has official data partitioning and the test set is evaluated using the Eval online test server.

Evaluation Metrics. We have adopted the standard testing and evaluation methodology used in motion prediction competitions to assess prediction performance. Key metrics for individual agents include Probabilistic minimum Final Displacement Error (p-minFDE), Minimum Final Displacement Error (minFDE), Minimum Average Displacement Error (minADE), Miss Rate (MR) and Drivable Area Compliance (DAC). Where p-minFDE, MR, and minFDE reflect the accuracy of the predicted endpoints, and minADE indicates the overall bias in the predicted trajectories. DAC, reflects the compliance of the predicted outcomes, which is incorporated into the bias when the predicted outcomes are outside of the drivable region.

Implementation Details. We generate lane vectors for lanes that are more than 50 meters away from any available agent. The number of layers l and k of interaction features are set to 3. All layers except the final decoder have 128 output feature channels. In addition, the weight parameters in the loss function are set to α=0.7, β=0.1 and λ=0.2. The hyperparameter τ in the difficulty masker was set to 5. The

TABLE I
COMPARISONS WITH THE STATE-OF-THE-ART METHODS LISTED ON THE LEADERBOARD OF THE ARGOVERSE 1 MOTION FORECASTING TEST SET. THE NUMBERS HIGHLIGHTED IN BOLD REPRESENT THE BEST-PERFORMING RESULTS AMONG THEM.

Inference	Methods	Year	p-minFDE ↓ (K=6)	minFDE ↓ (K=6)	MR ↓ (K=6)	minADE ↓ (K=6)	minFDE ↓ (K=1)	minADE ↓ (K=1)	DAC ↑
Single model	LaneGCN [29]	2020	2.05	1.36	0.162	0.87	3.76	1.70	0.9812
	DenseTNT [35]	2021	1.98	1.28	0.126	0.88	3.63	1.68	0.9875
	THOMAS [36]	2021	1.97	1.44	**0.104**	0.94	3.59	1.67	0.9781
	SceneTransformer [33]	2021	1.89	1.23	0.126	0.80	4.06	1.81	0.9899
	HiVT-128 [25]	2022	1.84	1.17	0.127	0.77	3.53	1.60	0.9888
	FRM [37]	2023	1.94	1.27	0.143	0.82	3.75	1.71	0.9878
	Macformer [38]	2023	1.83	1.22	0.120	0.82	3.72	1.70	0.9906
	LAformer [39]	2023	1.84	1.16	0.125	0.77	3.45	1.55	0.9897
	HeteroGCN [40]	2023	1.84	1.19	0.120	0.82	3.52	1.62	-
	GANet [14]	2023	1.79	1.16	0.118	0.81	3.46	1.59	0.9899
	DGFNet(single model)	-	**1.74**	**1.11**	0.108	**0.77**	**3.34**	**1.53**	**0.9909**
Ensembled model	HOME+GOHOME [41], [42]	2022	1.86	1.29	**0.085**	0.89	3.68	1.70	0.9830
	Multipath++ [43]	2022	1.79	1.21	0.132	0.79	3.61	1.62	0.9876
	DCMS [44]	2023	1.76	1.14	0.109	0.77	**3.25**	1.48	0.9902
	HeteroGCN-E [40]	2023	1.75	1.16	0.117	0.79	3.41	1.57	0.9886
	Macformer-E [38]	2023	1.77	1.21	0.127	0.81	3.61	1.66	0.9863
	Wayformer [45]	2023	1.74	1.16	0.119	0.77	3.66	1.64	0.9893
	ProphNet [16]	2023	1.69	1.13	0.110	0.76	3.26	**1.49**	0.9893
	QCNet [46]	2023	1.69	**1.07**	0.106	**0.73**	3.34	1.52	0.9887
	FFINet [17]	2024	1.73	1.12	0.113	0.76	3.36	1.53	0.9875
	DGFNet(ensembled model)	-	**1.69**	1.11	0.107	0.75	3.36	1.53	**0.9902**

model was trained 50 epochs on an Nvidia RTX 3090 with a batch size of 16. For the experiments on Argoverse 1&2, we used a segmented constant learning rate strategy: up to the 5th epoch we used 5×10^{-5}; from the 6th epoch to the 25th, we used 5×10^{-4}; from the 26th epoch to the 40th epoch, we used 3×10^{-4}; and thereafter, we used 5×10^{-5} until training was complete. Similar to LaneGCN [29], data augmentation was achieved by applying random scaling in the range [0.75, 1.25].

B. Quantitative Results

Quantitative evaluations on the Argoverse 1&2 datasets demonstrate that DGFNet exhibits significant performance advantages across various metrics compared to state-of-the-art methods. Our method remains competitive even when compared to ensemble strategies used by other top models like QCNet and ProphNet. Through an ablation study, we validate the effectiveness of scene feature fusion, future feature interaction, and a difficulty masking mechanism, resulting in substantial improvements in prediction accuracy. Computational efficiency analysis shows that DGFNet achieves high prediction accuracy with relatively fewer parameters, ensuring its practical applicability in real-time autonomous driving scenarios.

Comparison with State-of-the-Arts in Agoverse 1. We conducted performance comparisons between our method and the state-of-the-art methods on the test set of Argoverse 1 over the past three years. It should be noted that in the table, LaneGCN [29], DenseTNT [35], THOMAS [36], SceneTransformer [33], HiVT-128 [25], LAformer [14], GANet [14], R-Pred [49] performed inference as single models. HOME+GOHOME [41], [42], Multipath++ [43], DCMS [44], Wayformer [45], ProphNet [16], QCNet [46], and FFINet [17] used ensemble inference involving 5 or 10 models. Where Macformer [38] and HeteroGCN [40] have both single and ensemble inference results.

According to the supplementary materials of relevant papers, the ensemble methods used for the results with model ensembles are inconsistent. Generally, ensemble methods involve using K-means to match the closest trajectories from multiple models, followed by weighted computation to obtain the final result [50]. We obtained 5 models by setting different initial random seeds and performed ensemble inference using the aforementioned method. The results are shown in Tab.I. Compared to single-model methods, our approach consistently ranks at the top in almost all metrics. Even when compared with methods using ensemble strategies, our approach remains highly competitive, achieving prediction accuracy comparable to the leading models such as QCNet and ProphNet in the leaderboard. However, our method has advantages in terms of parameter size and inference time, which will be detailed in the following sections.

TABLE II
COMPARISONS WITH THE STATE-OF-THE-ART METHODS LISTED ON THE LEADERBOARD OF THE ARGOVERSE 2 SINGLE AGENT FORECASTING.

Method	Year	p-minFDE ↓ (K=6)	minADE ↓ (K=6)	minFDE ↓ (K=6)	MR ↓ (K=6)
LaneGCN [29]	2020	2.64	0.91	1.96	0.30
THOMAS [36]	2021	2.16	0.88	1.51	0.20
FRM [37]	2023	2.47	0.89	1.81	0.29
HDGT [23]	2023	2.24	0.84	1.60	0.21
HPTR [26]	2023	2.03	0.73	1.43	0.19
GoReal [24]	2022	2.01	0.76	1.48	0.22
GANet [14]	2023	1.96	0.72	**1.34**	0.17
DGFNet	-	**1.94**	**0.70**	1.35	**0.17**

TABLE III
COMPARISON OF PREDICTION PERFORMANCE (MINFDE) ON HIGH-DIFFICULTY SAMPLES IN THE ARGOVERSE 1 VALIDATION SET.

Method	**Top 1%**	**Top 2%**	**Top 3%**	**Top 4%**	**Top 5%**	**ALL**
LaneGCN	11.52	8.24	6.98	6.67	5.65	1.08
Our base	9.12↓21%	6.86↓17%	5.74↓18%	5.05↓24%	4.57↓19%	0.94↓13%
DGFNet	7.26↓**37%**	5.21↓**36%**	4.36↓**37%**	3.85↓**42%**	3.51↓**30%**	0.89↓**17%**

Comparison with State-of-the-Arts in Agoverse 2. In our comparison of performance with several recent single-model inference methods on the Argoverse 2 test set, we found that while the advantage of our method is not as pronounced as in Argoverse 1, it remains reasonably acceptable. We speculate that this is primarily due to the increased prediction horizon in the Argoverse 2 dataset, which has led to less robust initial prediction results. QCNet [46] was the first to propose mitigating the instability of long-term predictions through recurrent step. In future work, we will also explore similar approaches to improve robustness. Moreover, due to the inclusion of categories such as pedestrians and cyclists in Argoverse 2, the effectiveness of the parameter settings for our difficulty masker has also been impacted. Overall, our model demonstrates highly competitive performance within an acceptable range of model parameters across both datasets.

Comparison of Prediction Performance on High-difficulty Samples. Tab.III shows the highest minFDE errors of each method on the Argoverse 1 validation set. We reasoned on the validation set with LaneGCN's pre-trained model to obtain some sample sets with the highest errors, and then reasoned on these sample sets using our base model and DGFNet, respectively. The results show that our method DGFNet has significantly lower minFDE errors than the comparison methods in all percentage ranges, which fully demonstrates the significant advantage of our method in mitigating the long-tail problem in trajectory prediction.

Quantitative Results and Visualization. Tab.4 shows the qualitative visualization of DGFNet on the Argoverse 1&2 validation set. The model provides reasonable multimodal predictions for multiple target agents in different traffic scenarios. The eight selected scenarios have high prediction difficulty and strong representativeness. In Seq.1, Seq.2, and Seq.3, the model generates accurate predictions for each agent in a busy and strongly interacting intersection scenario. In Seq.3 and Seq.5, the model gives robust predictions in the face of vehicle turning tendencies. As a comparison, in Seq.6, Seq.7 and Seq.8, the model gives more scattered prediction results when the intention of the vehicle is not clear. In addition, the model captures the final landing very accurately in both turning and straight ahead situations.

C. Analysis and Discussion

Ablation Study. We conducted an ablation study on the Argoverse 1 validation set, which consists of 39,472 sequences. The initial model only used agent-centric historical trajectories and maps for feature extraction. The second model incorporated scene-centric features into the initial model, achieving some performance improvement through Scene Feature Fusion (SFF). The third model added the Future Feature Interaction (FFI) module, enabling interaction between extracted historical features and future features. The final model used the Difficulty Masker (DM) to mask out future trajectories that are difficult to predict, ensuring that only reliable trajectories are used for future interaction. Of concern is that without adding Difficulty Masker, the performance of direct future enhancement drops dramatically, failing because the second stage model may find shortcuts during optimization if the results given in the first stage are good enough [51]. It tends to replicate the results of the first stage rather than trying to extract richer information from future features. According to Tab.IV, comparing the base model with DGFNet, the improvement in predictive performance is evident. And we also did ablation experiments on the effect of parameter τ on the model. According to Tab.V, the model performance obtained by setting parameter τ to 5 is the best.

TABLE IV
THE ABLATION STUDY RESULTS OF DIFFERENT COMPONENTS ON THE ARGOVERSE 1 VALIDATION SPLIT, INCLUDING DIFFICULTY MASKER (DM), FUTURE FEATURE INTERACTION (FFI), AND SCENE FEATURE FUSION (SFF).

DM	FFI	SFF	p-minFDE ↓ (K=6)	minADE ↓ (K=6)	minFDE ↓ (K=6)
×	×	×	1.559	0.657	0.941
×	×	✓	1.525	0.646	0.916
×	✓	✓	1.654	0.706	1.052
✓	✓	✓	**1.492**	**0.632**	**0.892**

TABLE V
PERFORMANCE VERSUS τ PARAMETER OF DGFNET EVALUATED ON THE ARGOVERSE 1 VALIDATION SET.

τ	p-minFDE ↓ (K=6)	minADE ↓ (K=6)	minFDE ↓ (K=6)
7	1.510	0.642	0.907
5	**1.492**	**0.632**	**0.892**
3	1.499	0.634	0.897
1	1.508	0.639	0.905

TABLE VI
COMPARISON ON COMPUTATIONAL PERFORMANCE WITH OTHER METHODS.

Method	p-minFDE ↓ (K=6)	Param(M) ↓	Infer time(ms) ↓
LaneGCN [29]	2.054	3.68	17.2
GANet [14]	1.789	5.21	25.3
QCNet* [46]	1.693	7.30	-
HiVT-128 [25]	1.842	**2.47**	**13.2**
LAformer [39]	1.835	2.59	42.5
FFINet* [17]	1.729	6.20	-
Prophnet* [16]	1.694	15.20	-
DGFNet	**1.693**	4.53	32.1

Computational Performance. We compared the parameters of each method and the average inference time per scenario on the same device. As shown in Tab.VI, our inference time is slightly higher than LaneGCN and close to the lighter models HiVT-128 and LAformer. However, our method has a much higher prediction accuracy than these two methods. Compared with the methods with the highest prediction accuracy, our method has fewer parameters (note that those marked with * use ensemble strategies), which means that our model achieves high prediction accuracy with lower computational cost. Then, we recorded the inference time on the same experimental machine equipped with an NVIDIA GeForce RTX 3090. Specifically, we adjusted the batch size of all models to 1 and recorded their average time on the validation set. The model inference time indicates that the real-time latency of DGFNet is within an acceptable range. Since LAformer uses a two-stage inference approach, its inference time is longer despite having fewer parameters than our method.

V. Conclusion

In this paper, we propose a novel Difficulty-Guided Feature Enhancement Network (DGFNet) for muti-agent trajectory prediction. Distinguishing from general future enhancement networks, our model emphasizes filtering future trajectories by masking out and retaining only reliable future trajectories for feature enhancement, which greatly improves the prediction performance. Extensive experiments on the Argoverse 1&2 benchmarks show that our method outperforms most state-of-the-art methods in terms of prediction accuracy and real-time processing. In addition, we integrate agent-centric and scence-centric scene features in a concise and effective way, and demonstrate that the fusion of the two features is effective in improving the model prediction accuracy.

Emulating the predictive habits of human drivers is an intriguing direction for future research. This involves emulating human strategies when encountering different vehicles and making effective assumptions in the presence of incomplete perceptual information. Furthermore, we will integrate Difficulty-Guided feature enhancement into other prediction frameworks to further validate the efficacy of this module in optimizing multi-agent trajectory prediction problems.

References

[1] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[2] J. Wiederer, A. Bouazizi, M. Troina, U. Kressel, and V. Belagiannis, "Anomaly detection in multi-agent trajectories for automated driving," in *Conference on Robot Learning*. PMLR, 2022, pp. 1223–1233.

[3] W. Zhou, Z. Cao, Y. Xu, N. Deng, X. Liu, K. Jiang, and D. Yang, "Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 1275–1282.

[4] O. Makansi, Ö. Cicek, Y. Marrakchi, and T. Brox, "On exposing the challenging long tail in future prediction of traffic actors," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 147–13 157.

[5] X. Huang, S. G. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 9718–9724.

[6] B. Tang, Y. Zhong, U. Neumann, G. Wang, S. Chen, and Y. Zhang, "Collaborative uncertainty in multi-agent trajectory forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 6328–6340, 2021.

[7] Y. Wang, P. Zhang, L. Bai, and J. Xue, "Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1400–1409.

[8] A. Gulati, S. Soni, and S. Rao, "Interleaving fast and slow decision making," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1535–1541.

[9] G. Chen, J. Li, N. Zhou, L. Ren, and J. Lu, "Personalized trajectory prediction via distribution discrimination," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 580–15 589.

[10] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1341–1352, 2019.

[11] T. Qian, Y. Xu, Z. Zhang, and F. Wang, "Trajectory prediction from hierarchical perspective," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6822–6830.

[12] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "Tnt: Target-driven trajectory prediction," in *Conference on Robot Learning*. PMLR, 2021, pp. 895–904.

[13] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 759–776.

[14] M. Wang, X. Zhu, C. Yu, W. Li, Y. Ma, R. Jin, X. Ren, D. Ren, M. Wang, and W. Yang, "Ganet: Goal area network for motion forecasting," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1609–1615.

[15] G. Aydemir, A. K. Akan, and F. Güney, "Adapt: Efficient multi-agent trajectory prediction with adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8295–8305.

[16] X. Wang, T. Su, F. Da, and X. Yang, "Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 995–22 003.

[17] M. Kang, S. Wang, S. Zhou, K. Ye, J. Jiang, and N. Zheng, "Ffinet: Future feedback interaction network for motion forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[18] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *Conference on Robot Learning*. PMLR, 2022, pp. 1035–1045.

[19] J. Schmidt, P. Huissel, J. Wiederer, J. Jordan, V. Belagiannis, and K. Dietmayer, "Reset: Revisiting trajectory sets for conditional behavior prediction," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–8.

[20] Q. Sun, X. Huang, J. Gu, B. C. Williams, and H. Zhao, "M2i: From factored marginal trajectory prediction to interactive prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6543–6552.

[21] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 international conference on robotics and automation (icra)*. IEEE, 2019, pp. 2090–2096.

[22] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 525–11 533.

[23] X. Jia, P. Wu, L. Chen, Y. Liu, H. Li, and J. Yan, "Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding," *IEEE transactions on pattern analysis and machine intelligence*, 2023.

[24] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun, "Gorela: Go relative for viewpoint-invariant motion forecasting," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7801–7807.

[25] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the*

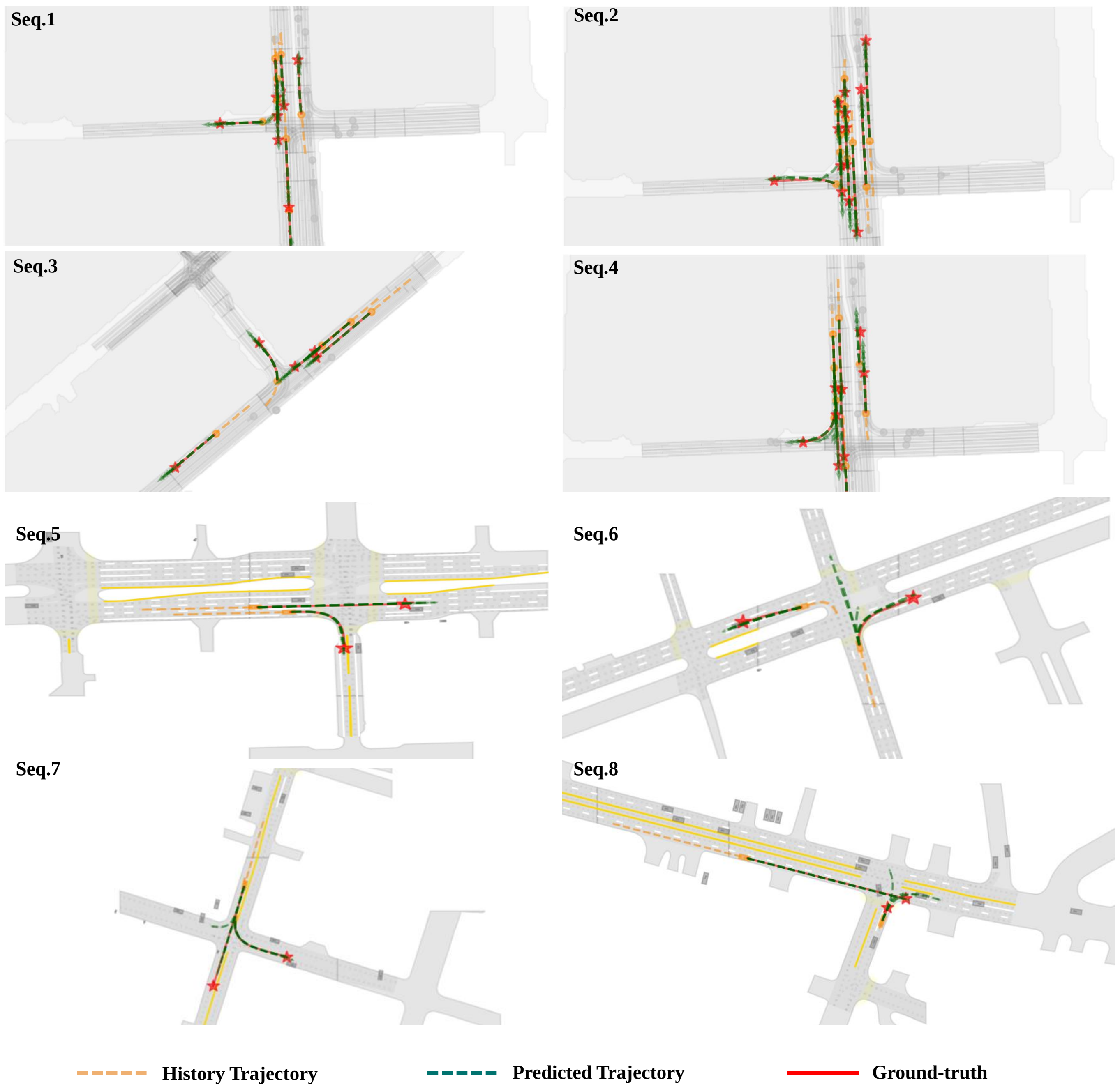


Fig. 4. Quantitative results of DGFNet on the Argoverse 1&2 validation set: observed trajectories of the proxy past are highlighted in orange, future trajectories of the ground truth are highlighted in red, and predicted trajectories are shown in green. The top four scenarios correspond to the Argoverse 1 dataset and the bottom four scenarios correspond to the Argoverse 2 dataset.

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8823–8833.

[26] Z. Zhang, A. Liniger, C. Sakaridis, F. Yu, and L. V. Gool, “Real-time motion prediction via heterogeneous polyline transformer with relative pose encoding,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[29] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, “Learning lane graph representations for motion forecasting,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 541–556.

[30] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[31] R. Girgis, F. Golemo, F. Codevilla, M. Weiss, J. A. D’Souza, S. E. Kahou, F. Heide, and C. Pal, “Latent variable sequential set transformers for joint multi-agent motion prediction,” *arXiv preprint arXiv:2104.00563*, 2021.

[32] S. Kim, H. Jeon, J. W. Choi, and D. Kum, “Diverse multiple trajectory

prediction using a two-stage prediction network trained with lane loss," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2038–2045, 2022.

[33] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.

[34] Y. Dong, H. Yuan, H. Liu, W. Jing, F. Li, H. Liu, and B. Fan, "Proin: Learning to predict trajectory based on progressive interactions for autonomous driving," *arXiv preprint arXiv:2403.16374*, 2024.

[35] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 303–15 312.

[36] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Thomas: Trajectory heatmap output with learned multi-agent sampling," *arXiv preprint arXiv:2110.06607*, 2021.

[37] D. Park, H. Ryu, Y. Yang, J. Cho, J. Kim, and K.-J. Yoon, "Leveraging future relationship reasoning for vehicle trajectory prediction," *arXiv preprint arXiv:2305.14715*, 2023.

[38] C. Feng, H. Zhou, H. Lin, Z. Zhang, Z. Xu, C. Zhang, B. Zhou, and S. Shen, "Macformer: Map-agent coupled transformer for real-time and robust trajectory prediction," *IEEE Robotics and Automation Letters*, 2023.

[39] M. Liu, H. Cheng, L. Chen, H. Broszio, J. Li, R. Zhao, M. Sester, and M. Y. Yang, "Laformer: Trajectory prediction for autonomous driving with lane-aware scene constraints," *arXiv preprint arXiv:2302.13933*, 2023.

[40] X. Gao, X. Jia, Y. Li, and H. Xiong, "Dynamic scenario representation learning for motion forecasting with heterogeneous graph convolutional recurrent networks," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2946–2953, 2023.

[41] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Home: Heatmap output for future motion estimation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 500–507.

[42] ——, "Gohome: Graph-oriented heatmap output for future motion estimation," in *2022 international conference on robotics and automation (ICRA)*. IEEE, 2022, pp. 9107–9114.

[43] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov *et al.*, "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.

[44] M. Ye, J. Xu, X. Xu, T. Wang, T. Cao, and Q. Chen, "Dcms: Motion forecasting with dual consistency and multi-pseudo-target supervision," *arXiv preprint arXiv:2204.05859*, 2022.

[45] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2980–2987.

[46] Z. Zhou, J. Wang, Y.-H. Li, and Y.-K. Huang, "Query-centric trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 863–17 873.

[47] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8748–8757.

[48] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arXiv preprint arXiv:2301.00493*, 2023.

[49] M. Ye, T. Cao, and Q. Chen, "Tpcn: Temporal point cloud networks for motion forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 318–11 327.

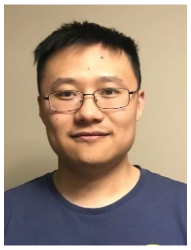
[50] Z.-H. Zhou and W. Tang, "Clusterer ensemble," *Knowledge-Based Systems*, vol. 19, no. 1, pp. 77–83, 2006.

[51] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

Guipeng Xin is currently pursuing his master's degree at the Intelligent Transportation Research Center, Wuhan University of Technology, China. His research interests include trajectory prediction and learning-based trajectory planning in autonomous driving.

Duanfeng Chu (Member) received the B.E. and Ph.D. degrees in mechanical engineering from Wuhan University of Technology, Wuhan, China, in 2005 and 2010, respectively. He has visited California PATH, University of California at Berkeley, Berkeley, USA, and the department of mechanical and aerospace engineering, The Ohio State University, USA, in 2009 and 2017, respectively. He is currently a Professor with the Intelligent Transportation Systems Research Center, Wuhan University of Technology. His research interests include automated and connected vehicle and intelligent transportation systems. Dr. Chu has been a reviewer for several international journals and conferences in the field of connected and autonomous vehicles.

Liping Lu received the B.E. degree in industrial automation control in 1996, the M.S. degree in automation control theory in 1999 from the Wuhan University of Technology, Wuhan, China, and the Ph.D. degree in computer science from the Institute National Polytechique de Lorraine, Nancy, France, in 2006. She is currently an Associate Professor with the School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China. Her research interests include autonomous vehicles and vehicular ad hoc networks.



Zejian Deng received his Ph.D. degree in Mechanical and Mechatronics Engineering from the University of Waterloo in 2022, the M.S. degree in Transportation Engineering and the B.S. degree in Automation from Wuhan University of Technology in 2018 and 2015. He is currently working as a postdoctoral fellow in Mechatronic Vehicle Systems Lab at the University of Waterloo. His research interests include driver model, driving safety validations, decision-making, planning and control in automated driving.

Yuang Lu is currently pursuing his master's degree at the Intelligent Transportation Research Center, Wuhan University of Technology, China. His research interests include learning-based trajectory prediction and learning-based perception in autonomous driving.

Xigang Wu received the M.S. degree and B.S. degree both in Transportation Engineering from Wuhan University of Technology, Wuhan, China, in 2024 and 2021, respectively. His research interests include trajectory prediction and decision-making in automated driving.