Article | Received 6 November 2025; Revised 12 April 2026; Accepted 22 April 2026; Published 29 May 2026
https://doi.org/10.55092/aias20260005

Multimodal trajectory prediction based on dynamic scene encoding and relational reasoning

Linwei Song[1], Zhengyi Li[1], Zhonghua Xiong[1], Zhiwen Wei[1], Rui Zhao[2] and Hongyu Hu[1,*]

[1] State Key Laboratory of Automotive Chassis Integration and Bionics, Jilin University, Changchun, China

[2] College of Automotive Engineering, Jilin University, Changchun, China

* Correspondence author; E-mail: huhongyu@jlu.edu.cn.

Highlights:

- A novel query-based framework (DRTR) is proposed for multimodal trajectory prediction.
- A relational reasoning-based feature selection module reduces model learning difficulty.
- The proposed DRTR model achieves promising results on the Argoverse 1 dataset.

Abstract: Autonomous vehicles require effective prediction of potential future trajectories of surrounding agents. The current trajectory prediction methods have limitations, firstly, traditional feature fusion methods merge scene features sequentially in a simplistic manner, often overlooking the intricate interrelations among scene elements, leading to incomplete selection and insufficient utilization of useful features; secondly, in multimodal trajectory prediction, the mode collapse issue inherent to probabilistic approaches results in inadequate expression of agent intent uncertainty, while overly anchor-dependent proposal-based methods can generate implausible trajectories. To address these limitations, We present a Dynamic scene and Relational reasoning Transformer (DRTR), a novel multimodal trajectory prediction method based on dynamic scene encoding and relational reasoning. A pivotal aspect of DRTR is the dynamic closed-loop modeling framework that effectively combines scene features to output three dynamic features: dynamic traffic flow, dynamic agents, and interactions between agents. This innovative framework ensures a comprehensive capture of the dynamic scene and its intricate interrelations. Then, DRTR initializes a set of trajectory suggestions representing various modalities and carefully refines these suggestions by sequentially fusing and querying dynamic scene features, ensuring predictions are both accurate and reflect multimodality. To further enhance model expressiveness, we introduce a feature selection network based on relational reasoning, which can recognize deep relationships between scene elements and select beneficial contextual features. Experiments on the Argoverse 1 dataset indicate that DRTR exhibits superior performance, particularly in multimodal trajectory prediction.

Keywords: dynamic scene encoding; relational reasoning; multimodal prediction; trajectory prediction

1. Introduction

The advent of autonomous driving promises to completely transform our mode of travel, offering new solutions to traffic congestion and reducing traffic accidents [1,2]. Accurately predicting the future trajectories of surrounding agents is crucial for autonomous vehicles to fully understand their environment and make safe and efficient decisions. In addition to the dynamic constraints of the target agents, the limitations imposed by lane lines, traffic regulations, and interactions with surrounding agents all significantly impact the future trajectories of these agents. At the same time, due to the uncertainty of the intentions of intelligent agents, future behaviors are also uncertain, and even in the same scenario, intelligent agents can exhibit different behaviors. Therefore, how to effectively model the scene, fuse features, and output reasonable multimodal prediction trajectories has long been the core challenge of trajectory prediction.

The purpose of scene modeling is to efficiently express the scene and guide the learning direction of the [3]. Trajectory prediction methods leverage information such as historical trajectory data of agents, traffic rules, map data, and interaction data amongst agents [4]. Every piece of information is interrelated and the scene is dynamically changing. However, most existing feature fusion methods employ static fusion and do not consider all correlations [5–7]. For example, some methods incorporate map features as context features into agent motion features after encoding, which only provides the agent with map information. Still, the map information remains isolated and static. Another method is directly merging scene elements in one go-through composition [8–10]. These simple modeling methods all lead to incomplete scene modeling and insufficient feature utilization issues. Therefore, this paper proposes dynamic closed-loop modeling of the scene, emphasizing the dynamic correlation between all elements. This approach to dynamic fusion features and obtains three characteristics: dynamic traffic flow, dynamic agent, and dynamic agent interaction features.

The widely used methods for integrating the aforementioned key cues usually involve directly merging all features [8,11–13]. These features are both correlated and redundant. If a neural network learns without an explicit basis, it becomes challenging to discern the relationships between different features, leading to learning difficulties and parameter wastage. Therefore, this paper proposes a feature selection network based on relational reasoning. It infers the relationships between base and context features based on prior knowledge. Then, it selects features based on the inferred relationships, allowing the neural network to understand the relationships between features, and effectively select useful information from the features. This process reduces learning difficulty and enhances prediction performance.

In addressing multimodal trajectory prediction issues, models need to learn how to model the uncertainty of trajectories and capture the underlying multimodal distribution. Current methods are primarily categorized into probabilistic methods and proposal-based methods. Probabilistic methods [11,14–16] define potential models as latent variables, subsequently mapping the merged features or directly constraining them onto a known probability distribution, thereby achieving implicit uncertainty modeling. This approach heavily depends on the mapping method and probability distribution, leading to potential issues such as optimization instability and mode collapse. Proposal-based methods, on the other hand, require predefined candidate points or trajectories [8,17–20]. Although these methods mitigate some drawbacks of probabilistic methods,

they still heavily rely on the selection of candidate points. Furthermore, the input used for decoding in these methods is the feature obtained from the concatenation process during the feature fusion stage, making it challenging for the decoder to fully utilize cues and eliminate redundant features. To better leverage each cue and more accurately capture potential multimodal trajectories, this paper proposes a multimodal trajectory query network. The multimodal trajectories are initialized as multiple query vectors. Each feature obtained during the dynamic scene modeling process is sequentially fused with the query vectors, which are finally input into the decoder to obtain the corresponding modal trajectories. Specifically, the fusion process adopts a feature selection network based on relational reasoning.

In summary, we propose a model, named Dynamic scene and Relational reasoning Transformer (DRTR), that uses dynamic scene information and relational reasoning to predict the future multimodal trajectories of agents. The proposed model is composed of an encoder, a feature fusion network, a multimodal trajectory query network, and a decoder. The encoder is divided into agent encoding and map information encoding. The agent encoding uses a multi-head attention mechanism and convolutional blocks, while the map feature encoding borrows from the LaneGCN [11]. The feature fusion network completes the dynamic coupling of all scene elements, and outputs three features: dynamic traffic flow, dynamic agent, and dynamic agent interaction features. At the same time, we propose a feature selection network based on relational reasoning, which eliminates a large amount of redundant information while effectively selecting useful information from the features. The multimodal trajectory query network randomly initializes suggestions representing multimodality, then combines them with the feature selection network based on relational reasoning to efficiently query scene dynamic information and refine multimodal suggestions. The decoder uses refined multimodal trajectory suggestions as input to generate multimodal trajectories and their probabilities. Finally, comparative experiments and ablation experiments are conducted to demonstrate the advantages of the proposed model, and some instances are visualized to qualitatively explain the prediction results.

The main contributions of this work can be summarized as follows:

- We propose a query-based multimodal trajectory prediction framework *(i.e.*, DRTR) that effectively models scene interaction relations and integrates dynamic scene context information for multimodal prediction.
- We introduce a feature selection module based on relational reasoning, allowing base features to purposefully and hierarchically select the context features. This network is applied in the dynamic feature fusion module and the multimodal trajectory query network, reducing the learning difficulty of DRTR.
- Experimental results demonstrate that the proposed model framework and feature selection network bring about actual improvements. Our model has achieved promising results on the Argoverse 1 dataset.

The remainder of the paper is organized as follows. Section 2 summarizes the methods and current progress in multimodal trajectory prediction research. In Section 3, the proposed approach is presented. Section 4 explains the experiments conducted using the Argoverse 1 dataset to validate the performance of the proposed method. Conclusions are presented in Section 5.

2. Related works

This section focuses on trajectory prediction methods based on deep learning, and briefly reviews the key technologies and methods involved, including Extracting Scene Contextual Information, Feature Fusion, and Multimodal Trajectory Prediction.

2.1. Extracting scene contextual information

Trajectory prediction necessitates the integration of rich contextual information from the scene. Inspired by convolutional neural networks (CNNs), early works utilized bird's-eye view raster images as inputs for learning [16,18,19,21]. However, the performance of these methods is constrained by the spatial resolution of raster images, which are easily disturbed by irrelevant raster areas. Additionally, raster images suffer from lossy rendering and high costs. Consequently, encoding schemes based on raster images have been gradually phased out in favor of more efficient vector-based encoding methods. Vector-based encoding describes topological information and ignores irrelevant details, particularly in scenarios like long, straight highways that contain a lot of redundant information. This significantly compensates for the shortcomings of raster image encoding methods. VectorNet [9] represents agents and roads in vector form for the first time, avoiding information loss and the need for dense convolution-related calculations in raster image rendering. Graph convolution methods, represented by LaneGCN [11], have demonstrated the powerful capability of vectorization in feature expression [22–24].

2.2. Feature fusion

Vector-based methods effectively aggregate information in traffic scenes, with many advanced trajectory prediction models [10,25–27] employing attention mechanisms for feature encoding and fusion. Most models traditionally encode map and agent features first, then integrate map information into agent information, and subsequently perform inter-agent feature fusion [5–7]. Another method is simply to complete the fusion of all features in a single layer [8–10]. This type of method also has the commonality of outputting only feature sets that aggregate all elements. These methods all lead to feature fusion obtaining static scene information and failing to fully express and integrate scene features. To address these issues, our method proposes dynamic closed-loop fusion, achieving a complete fusion of scene features.

Currently, deep learning models typically aggregate features by directly fusing the hidden states among various features, such as [14,28–30]. Other models use basic priors such as distance to filter elements of interest in the scene. For instance, the multi-head attention-based Long Short-Term Memory (LSTM) encoder/decoder structure proposed by Messaoud *et al.* [31] maps agent positions to an attention matrix and visualizes their relative importance around the ego-vehicle; similarly, STAG [32] infers the relative importance of surrounding vehicles through distance information. These simplistic approaches that do not meticulously select predictive features from prior knowledge limit the expressive capacity of the models, failing to leverage their potential advantages fully. Recent studies have explored the relationships among traffic elements in greater depth. For instance, Liao *et al.* [33] proposed the behavior-aware trajectory prediction model BAT, which characterizes driving behaviors in dynamic geometric graphs

by introducing centrality measures as prior knowledge, thereby capturing potential interactions among vehicles. HLTP [34] proposed a novel vision-aware pooling mechanism that simulates the dynamic adaptation of a driver's visual sector with vehicle speed: as the speed increases, the visual sector becomes narrower to focus more on the area ahead, whereas as the speed decreases, the visual sector becomes wider to enhance the perception of the surroundings. Xu *et al.* [35] propose GroupNet, a multiscale hypergraph neural network, which explicitly reasons some relational factors including the interaction category, interaction strength, and interaction function; Xu *et al.* [36] introduced the Joint-Relation Transformer, which has achieved significant results by using explicit relational data such as the distance between two individuals, as well as skeletal and joint information of the individuals, as prior knowledge for feature fusion. Inspired by this, this paper introduces a feature selection network based on relational reasoning. This network takes the distance and feature correlation between base and context features as inputs to infer the deep internal relationships and carefully selects key predictive features in the context based on these relationships, significantly improving the model's expressive power.

2.3. Multimodal trajectory prediction

In trajectory prediction tasks, the future state of the scene is unpredictable, and the intentions of agents are highly uncertain, predicting diverse multimodal trajectories is more advantageous in addressing the uncertainty of agent behaviors compared to unimodal predictions. Currently, there are two main methods for decoding multimodal trajectories. The first method decodes a set of discrete trajectories directly using features from the model's scene encoding module [11,14–16,37]. This approach is simple to implement and allows for the output of multiple trajectory hypotheses, but it heavily depends on the encoder's performance and the decoder's ability to extract useful features. Given that only one real trajectory can be observed in the training data, this method poses significant challenges. The second method is based on anchors, which utilizes predefined operations and candidate trajectories [8,17–20] to achieve multimodal predictions. This method is clearly influenced by the quality of the anchors, and an excessive focus on candidate trajectories can overlook the potential meanings of scene features. Our proposed multimodal trajectory query network has been improved in two aspects: feature utilization and anchor refinement. The specific implementation is to randomly initialize trajectory suggestions and gradually refine them based on dynamic scene features to obtain reasonable multimodal trajectories.

3. Development of proposed model

This section begins by introducing the trajectory prediction problem, detailing the inputs and prediction objectives. It continues with a comprehensive examination of the various sub-modules within DRTR and the associated loss functions. Additionally, this section includes schematic diagrams of the overall model structure and the feature selection network based on relational, facilitating a deeper understanding of the model's operating mechanism.

3.1. Problem formulation

To enhance the accuracy of trajectory prediction, all information within the environment that could potentially impact future trajectories is incorporated as inputs to the model, specifically categorized into

agent information and scene information.

The set of agents is denoted as $\mathscr{A} = \{A_0, A_1, ..., A_{N_s}\}$, where A_0 is the target agent, and N_s represents the number of other perceived agents besides the target agent. $A_i = \{a_i^{-t_h+1}, a_i^{-t_h+2}, ..., a_i^0\}$, where t_h is the length of the perceived time series, $i = 0, 1, ..., N_s$, and $a_i^t \in \mathbb{R}^{f_a}$, f_a is the number of features per agent. The coordinate system of the scene is fixed at the position of the target agent at time $t = 0$ (i.e., the current moment), with the driving direction of the target agent defined as the x-axis and a 90° counterclockwise rotation defined as the y-axis. Meanwhile, we select the features of the vehicle at time t as $a_i^t = \{{}_\Delta x_i^t, {}_\Delta y_i^t, class_i, flag_i^t\}$. Note that $t = -t_h+1, -t_h+2, ..., -1, 0$, where Δx_i^t and Δy_i^t represent the differences in the x and y coordinates of the agent between times t and $t-1$, $class_i$ represents the type of agent (such as vehicle, non-motor vehicle, or pedestrian), and $flag_i^t$ is a placeholder indicating whether the agent is perceived at the current moment. To facilitate the relational reasoning module and feature selection network, we also input the coordinates d of each agent at time $t = 0$.

Scene information $\mathscr{S}$ includes lane markings, traffic signs and traffic lights. Lane markings are vectorized and structured into a graph, with lane nodes denoted as $\mathscr{L} = \{L_0, L_1, ..., L_{N_l}\}$ and the adjacency matrix as $\mathscr{M} = \{M_0, M_1, ..., M_{N_l}\}$. The number of lane vector elements in the scene is $N_l + 1$. For convenience in feature representation, information such as traffic lights is integrated into the lane features. Specifically, $L_j = \{{}_\Delta x_j, {}_\Delta y_j, heading_j, turn_j, light_j\}$ and $j = 0, 1, ..., N_l$, where Δx_j and Δy_j represent the difference between the endpoint and starting point of the lane node on the x-axis and y-axis, $heading_j$ is the orientation of lane nodes, $turn_j$ indicates the permissible directions of the lane (straight, left turn, or right turn), and $light_j$ describes the traffic light control affecting the lane. The adjacency matrices $M_j = \{M_{j,p}^{1:6}, M_{j,s}^{1:6}, M_{j,r}, M_{j,l}\}$, $M_j \in \mathbb{R}^{(N_l+1)\times(N_l+1)}$, where $M_{j,p}^{1:6}, M_{j,s}^{1:6}, M_{j,r}, M_{j,l}$ represents the predecessors, successors, right neighbors, and left neighbors of the lane, respectively, and 1:6 considers the 6 predecessors or successors of the current lane line.

The network proposed in this study aims to predict the trajectories $\hat{Y}_{i,k}^t = (\Delta\hat{x}_{i,k}^t, \Delta\hat{y}_{i,k}^t)^\mathsf{T}$ of all agents in the scene within a future time span of t_f, in K different modes, along with the corresponding probabilities $\hat{p}_i^k$ of each trajectory, where, $i = 0, 1, ..., N_s$, $k = 0, 1, ..., K-1$, and $t = 1, 2, ..., t_f$.

3.2. Proposed model

The proposed network consists of four main components: an encoder, a feature selection network based on relational reasoning, a multimodal trajectory query network, and a decoder. Initially, the network performs feature encoding of both agent information and scene information. It then proceeds to fuse multi-source heterogeneous features and to query and refine multimodal trajectory features. Ultimately, the decoder outputs the multimodal trajectories. The model structure is shown in Figure 1, and the following sections will provide a detailed explanation of each module's functionality and significance.

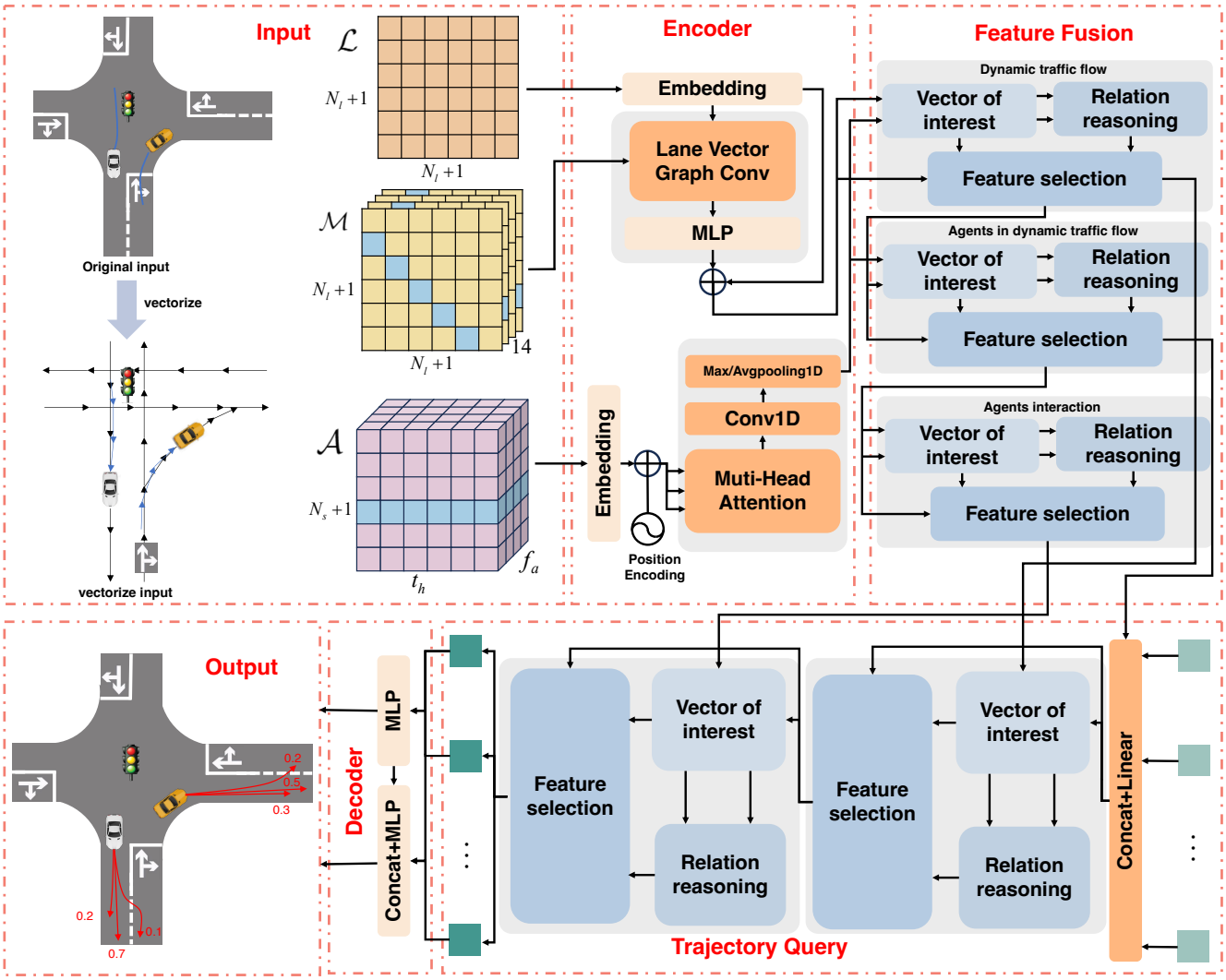


Figure 1. Schematic of DRTR. The input to the Encoder is shown by Input. The network is mainly composed of the Encoder, Feature Fusion Network, Trajectory Query Network, and Decoder. The output of DRTR is shown by Output.

3.2.1. Encoder

The primary task of the encoder is to encode the features of agents and lane lines. The features of agents include time-series information and thus are encoded using a sequence feature encoding [38]. In contrast, lane line features do not contain time-series information and are structured as graphs, necessitating a different encoding strategy.

The encoder adopts the scaled dot-product attention mechanism [39] and convolutional neural networks. And stack N_a-layers. To effectively utilize the temporal information of agent $\mathscr{A}$ historical features, we employ Position Embedding (PE) layers, similar to those in the original Transformer. Subsequently, the agent's historical trajectory features are processed using a Multi-Head Self-Attention mechanism, and the outputs from all heads are concatenated. Let the input to the attention module be denoted as $X_A \in \mathbb{R}^{L \times d_{model}}$, X_A is the result of embedding A_i, the sequence length as L, with the observation time length t_h in the first convolutional pooling unit, followed by the output sequence length of the previous convolutional pooling unit, and the input feature dimension as d_{model}. Then, the output of the multi-head attention can be represented as follows:

$$X_{MA} = concat(MHSA(X_{A,h} + PE)) \tag{1}$$

where $h = 1, 2, ..., H$, and H represents the number of heads. Subsequently, the output of the multi-head attention is added to the original input and layer normalized to integrate the features:

$$X_A^m = \phi(X_A + X_{MA}) \tag{2}$$

where ϕ is the normalization function. Then, the feedforward layer is adopted as:

$$X_A^f = \phi(X_A + \mathrm{ReLU}(X_A^m W_{A1}^m + b_{A1}^m) W_{A2}^m + b_{A2}^m) \tag{3}$$

where $W_{A1}^m \in \mathbb{R}^{d_{model} \times 4d_{model}}$, $b_{A1}^m \in \mathbb{R}^{4d_{model}}$, $W_{A2}^m \in \mathbb{R}^{4d_{model} \times d_{model}}$, $b_{A2}^m \in \mathbb{R}^{d_{model}}$, representing is the weights and biases of linear layers.

After implementing the multihead self-attention mechanism, to further extract useful features, convolutional pooling units are employed. The first $N_a - 1$ layers consist of two one-dimensional convolution layers, a normalization layer, and a non-linear activation function layer. Following these processes, the data is then sent to a max pooling layer for further feature extraction:

$$X_A^{c1} = \mathrm{ReLU}(\varphi(Conv1d(X_A^f))) \tag{4}$$

$$X_A^{c2} = \mathrm{ELU}(\varphi(Conv1d(X_A^{c1}))) \tag{5}$$

$$X_A^{out} = MaxPool(X_A^{c2}) \tag{6}$$

where X_A^{c1} is the output of the first convolution block, X_A^{c2} is the output of the second convolution block, $X_A^{out} \in \mathbb{R}^{L/2 \times d_{model}}$ is the output of max pooling, Maxpool is the Max Pooling Layer. In the final convolutional pooling unit, we switch from max pooling to average pooling to compress the sequence length to 1, thereby obtaining the final output:

$$X_A^{out} = AvgPool(X_A^{c2}) \tag{7}$$

where $X_A^{out} \in \mathbb{R}^{d_{model}}$, AvgPool is the Average Pooling Layer.

Lane vectors $\mathscr{L}$ are encoded using LaneGCN [11]. Given that agents typically have high longitudinal speeds, which allow them to cover considerable distances in a short period, there is an increased reliance on additional lane vectors along the direction of the lane. Consequently, it is necessary to convolve more predecessor and successor lane vectors. LaneGCN incorporates dilated convolution operators to capture more dependencies along the direction of the lane lines.

$$X_L^m = LW_L + \sum_{i \in \{\mathrm{l,r}\}} M_i LW_i + \sum_{c=1, j \in \{\mathrm{s,p}\}}^{C} (M_j^{k_c} LW_j) \tag{8}$$

where $W_{\cdot}$ is the weight of LaneConv, and k_c is the cth dilation size. Normalization and rectified linear units are adopted after each LaneConv and linear layer. Subsequently, it is passed through a connected multi-layer perceptron (MLP), followed by a skip connection and another non-linear activation:

$$X_L^n = \mathrm{ReLU}(\phi(X_L^m)) \tag{9}$$

$$X_L^{out} = \mathrm{ReLU}(f_{MLP}(X_L^n) + L) \tag{10}$$

where, ϕ represents the normalization function, and f_{MLP} refers to the function of the multi-layer perceptron. It is important to note that to enhance the extraction of lane line features, we stack N_l layers of LaneGCN in the model.

3.2.2. Feature fusion module

After completing the initial feature encoding, to facilitate dynamic closed-loop modeling of the scene, a series of feature fusions are performed. This includes three feature fusion modules: agent-to-lane line, lane line-to-agent, and agent-to-agent, which respectively yield features of dynamic traffic flow, agent features after integrating dynamic traffic flow, and interaction features between agents. These feature fusion processes are implemented using a feature selection network based on relational reasoning, which is detailed in Section 3.2.3.

3.2.3. Feature selection network based on relational reasoning

After encoding through the encoder, both agent features and lane vector features are encoded. During feature fusion, agents or lane lines and unrelated features that do not require attention complicate network learning and diminish the model's expressiveness. For instance, in the process of feature fusion between agents, it is unreasonable to consider all agents in the scene, as those far away will not impact the future trajectory of the target vehicle. Moreover, even after selecting agents to focus on based on distance, determining the attention weight for each agent requires network learning. Additionally, high-dimensional features often contain a lot of redundant elements, which also need to be identified and eliminated by the network. Failure to address these issues results in learning difficulties and parameter waste. To enable agents to selectively focus on relevant scene elements purposefully and to assign greater weight to important elements and features, this study proposes a feature selection network based on relational reasoning, primarily implemented through a sparse attention mechanism.

The input to the sparse attention mechanism includes the base features (Base) and the context features (Context), which, under the premise of prior knowledge and relational reasoning, enables the base features to aggregate useful context features. Initially, the Euclidean distance between the base and context features is used as prior knowledge to preliminarily filter the set of context features of interest, denoted as X_{se}, and to record these Euclidean distances D_{se}:

$$D_{se} = \{d_{uv} \mid d_{uv} = d_u - d_v, \|d_{uv}\|_2 < \delta_d\} \tag{11}$$

$$X_{se} = \{x_v \mid \|d_{uv}\|_2 < \delta_d\} \tag{12}$$

where d_u, d_v are the coordinates of the basic and the context element in the local coordinate system respectively, $u = \{0, 1, ..., N_t\}, v = \{0, 1, ..., N_c\}$, $N_t + 1$ and $N_c + 1$ are the number of base and context elements and δ_d is the Euclidean distance threshold of the area of interest.

Next, the base feature x_u, each element of interest x_v, and the corresponding Euclidean distance d_{uv} are fed into linear layers, resulting in the processed base feature X_{ba}, context feature X_{ctx}, and distance information D_{bc}. These are then concatenated and passed into another linear layer. After normalization, the Exponential Linear Units (ELU) are used as the activation function, resulting in the input for the feature selection network based on relational reasoning:

$$V = \mathrm{ELU}(\phi(concat(X_{ba}, X_{ctx}, D_{bc})W_V + b_V)) \tag{13}$$

where W_V and b_V are the weights and biases of the linear layer respectively.

This study initially establishes the relationships between base features and context features through relational reasoning and then employs a gating mechanism to select useful features from the context.

The initial step involves obtaining the raw features through a linear layer:

$$I = VW_I + b_I \tag{14}$$

where W_I and b_I are the weights and biases of the linear layer respectively. In the relational reasoning part, this study bases its reasoning of foundational features on feature correlation and distance. For feature correlation R_a, inspired by the dot-product attention mechanism [39], we input the base feature X_{ba} and the context feature X_{ctx} into a linear layer, then calculate their Hadamard product (element-wise multiplication), followed by one-dimensional convolution. For distance feature R_d, it is directly mapped using a linear layer:

$$R_a = Conv1d((X_{ba}W_{at} + b_{ab}) \odot (X_{ctx}W_{ac} + b_{ac})) \tag{15}$$

$$R_d = X_d W_d + b_d \tag{16}$$

At this point, we have obtained the initial features R_a and R_d representing the relationships between the base features and context features. Next, we will further extract and express these relationships to infer and obtain the deep relational features R:

$$R_{ad} = concat(R_a, R_d)W_{R_{ad}} + b_{R_{ad}} \tag{17}$$

$$R = f(R_{ad}) + f(f(X_{ba} + R_{ad}) \odot f(X_{ctx} + R_{ad})) \tag{18}$$

Where f is the linear layer. Following this, we employ a gating mechanism based on the deep relational features R to select useful features from the context.

$$X_s = \sigma(I + R) \odot V \tag{19}$$

$$X_g = \mathrm{ReLU}(\phi(V + X_s)) \tag{20}$$

where $\sigma(\cdot)$ is the sigmoid function. Then, the selected features are added to the base features and normalized:

$$X^m = \phi(X_{ba} \oplus X_g) \tag{21}$$

where $\oplus$ indicates an index-add. Subsequently, the data is fed into a feedforward layer consistent with Equation (3) to produce the output X^{out}. It is important to note that each step of the feature fusion process involves the use of N_b layers of the feature selection network based on relational reasoning, stacked together.

Through the feature selection network based on relational reasoning, base features are fused with selected useful features, thereby reducing learning complexity and achieving efficient feature integration. This network serves as a submodule applied in the feature fusion processes for agent-to-lane lines, dynamic traffic flow to agents, agent-to-agent interactions, and in the map and interaction query modules of multimodal trajectory queries network described in the next section. Specifically, in the agent-to-lane

line module, the base features are lane line features X_L^{out}, and the context features are agent features X_A^{out}, aiming to integrate dynamic vehicle information into the lane line data to obtain dynamic traffic flow features X_{dl}^{out}. In the dynamic traffic flow to agent module, the base features are agent features X_A^{out}, and the context features are dynamic traffic flow X_{dl}^{out}, intended to focus the agent on the dynamic traffic and obtain dynamic agent features X_{da}^{out}. In the agent-to-agent module, both the base and context features are dynamic agent features X_{da}^{out}, aiming to facilitate dynamic interactions between agents and obtain dynamic agent interaction features X_{di}^{out}, where is further divided into target agent feature X_{tar}^{out} and surrounding agent X_{sur}^{out}. In the query network, the base features are query vectors, while the context features include dynamic traffic flow X_{dl}^{out} and agent interaction features X_{di}^{out}, enabling the query vectors to select useful features for feature fusion. Figure 2 illustrates this with an example of agent interaction feature fusion.

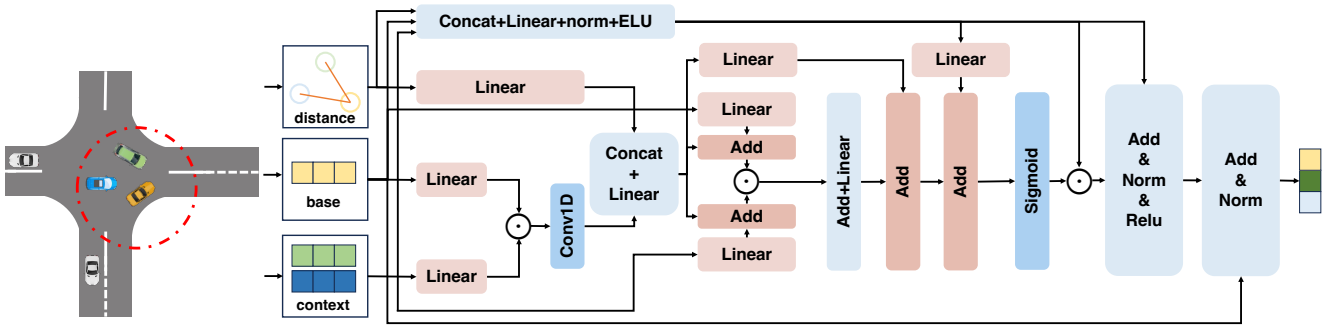


Figure 2. Schematic of feature selection network based on relational reasoning.

3.2.4. Multimodal trajectory query network

This study aims to predict multimodal trajectories of target agents, and integrating multi-source information poses significant challenges. Inspired by DETR [40], we define Transformer queries as multimodal trajectory suggestions. This approach allows each suggestion to independently integrate information from the feature fusion module without interference, ensuring that each trajectory suggestion corresponds directly to its respective modality. During execution, features from the ego-vehicle, map, and interactions are fused hierarchically. Each layer updates and refines the trajectory suggestions, with the refined suggestions from one layer directly serving as the input for the next layer. Given that different modal suggestions may need to focus on different scene features, the feature selection network based on relational reasoning is also employed to enable trajectory suggestions to select important features for trajectory updating and refinement. Initially, trajectory suggestions are randomly initialized and their gradients are frozen:

$$\mathscr{Q} = \{Q_i | i = 0, 1, ..., N_s, grads = False\} \tag{22}$$

where $Q_0 = Q_1 = ... = Q_{N_s} \in \mathbb{R}^{K \times d_{\text{model}}}$, $N_s + 1$ is the number of agent. Following the previously described hierarchical feature aggregation method, we first aggregate the features of the target agent, followed by the aggregation of lane line information, and finally, the aggregation of interaction features:

$$Q_T = concat(X_{tar}^{out}, Q)W_T + b_T \tag{23}$$

$$Q_M = RE(Q_T, X_{dl}^{out}) \tag{24}$$

$$Q_I = RE(Q_M, X_{di}^{out}) \tag{25}$$

where Q_T, Q_M, and Q_I represent the refined trajectory suggestions after feature aggregation of the target dynamic agent, dynamic traffic flow feature aggregation, and interaction feature aggregation, respectively. W_T is the weight of the fully connected layer, b_T is the bias of the fully connected layer, and $RE(\cdot,\cdot)$ refers to the relational reasoning and feature selection network function described in Section 3.2.3.

3.2.5. Decoder

The decoder module is divided into two branches: a multimodal trajectory generator and a probability generator for the corresponding trajectories. The multimodal trajectory generator employs a three-layer MLP to generate predicted trajectories:

$$\mathscr{Y} = \{\hat{Y}_i | \hat{Y}_i = f_{MLP}(Q_I), i = 0, 1, ..., N_s\} \tag{26}$$

where $\hat{Y}_i \in \mathbb{R}^{K\times T\times 2}$. To enhance the performance of trajectory classification, the predicted value $\hat{Y}_i^{t_f} \in \mathbb{R}^{K\times 2}$ at the last moment t_f is used alongside the distance d_{by} from the current position of the target agent as context features. These features are concatenated with the trajectory suggestions Q_I, then decoded through a three-layer MLP. Finally, the probabilities of each predicted trajectory are output through a Softmax layer:

$$\mathscr{P} = \{\hat{p}_i \mid \hat{p}_i = Softmax(f_{MLP}(concat(Q_I, \hat{Y}_i^{t_f})))\} \tag{27}$$

where $\hat{p}_i \in [0,1]^{K\times 1}$, $i = 0, 1, ..., N_s$.

3.2.6. Loss function

The network proposed in this study is designed to predict the multimodal trajectories of agents and their corresponding probabilities. All modules of the network are differentiable, thus allowing for fully supervised end-to-end training. The loss components include probability classification loss for trajectories, trajectory regression loss, and multimodal classification loss.

For any agent, let $k^* = argmax(p)$ be the trajectory with the highest probability. To increase the confidence of the k^*th trajectory, we set k^* as the positive class and $k \neq k^*$ as the negative class, using the cross-entropy loss function for calculation:

$$\mathscr{L}_{\mathrm{C}} = -\frac{\sum_{n_s=0}^{N_s}\sum_{c=1}^{K} \tilde{p}_{n_s}^{k^*} \log \hat{p}_{n_s}^{c}}{N_s + 1} \tag{28}$$

To enhance the multimodal expression capability of the multimodal trajectory query network, we employ Max-margin loss as an auxiliary function. This loss function is designed to optimize the model to more effectively distinguish between different modalities, thereby improving the overall expressiveness of the model:

$$\mathscr{L}_M = \frac{\sum_{n_s=0}^{N_s}\sum_{k\neq k^*} \max(0, \hat{p}_{n_s}^{k} - \hat{p}_{n_s}^{k^*} + \varepsilon_M)}{(N_s+1)(K-1)} \tag{29}$$

For trajectory regression, we utilize the Smooth L1 Loss to calculate the discrepancy between predicted values and actual values:

$$\mathscr{L}_{\mathrm{R}}=\begin{cases}\dfrac{\sum_{n_s=0}^{N_s}\sum_{j=1}^{t_f}\frac{1}{2}\left\|\hat{Y}^{j}_{n_s,k^*}-Y^{j}_{n_s}\right\|_2^2}{N_s+1}, \text{if}\left\|\hat{Y}^{j}_{n_s,k^*}-Y^{j}_{n_s}\right\|<1\\ \dfrac{\sum_{n_s=0}^{N_s}\sum_{j=1}^{t_f}\left\|\hat{Y}^{j}_{n_s,k^*}-Y^{j}_{n_s}\right\|-\frac{1}{2}}{N_s+1}, \text{otherwise}\end{cases} \tag{30}$$

The final loss function is the sum of the above three items:

$$\mathscr{L}=\mathscr{L}_C+\mathscr{L}_M+\mathscr{L}_R \tag{31}$$

4. Experiment

4.1. Experimental setting

The Argoverse [41] public motion dataset used in this study contains agent information of cars, bicycles, and pedestrians, as well as High-Definition (HD)-map information collected on urban roads in Pittsburgh and Miami in the United States. All data were collected by vehicles equipped with LiDAR and cameras at a frequency of 10 Hz. The dataset was split into training, validation, and testing sets with 205,942, 39,472, and 78,143 sequences, respectively. The future trajectory of the test set was not provided, and there was no geographic overlap between sets. A 5-s period was selected to describe the trajectory of each vehicle: 2 s were used as historical input, and 3 s were used as the trajectory to be predicted.

All experiments were performed on an Intel Core(R) i9-13900K CPU, NVIDIA GeForce(R) RTX 4090 24-GB GPU with 64-GB of RAM running the Ubuntu 20.04 LTS edition. All program tasks were conducted on Python 3.7, and the deep learning framework was based on PyTorch.

4.2. Training setting and experimental metrics

The model processes input sequences of length $t_h = 20$ for the agent, and outputs predicted trajectories of length $t_f = 30$ across $K = 6$ modalities. The feature dimension d_{model} within the model is set to 128. In the multi-head attention mechanism, $H = 4$ is selected. The agent encoder employs the attention mechanism stacked $N_a = 4$ times, while the number of stacking operations for map fusion with agents, agents merging with dynamic traffic flow, and interactions among agents is set to $N_b = 2$. The model is trained for a total of 40 epochs with a batch size set to 48, utilizing the Adam optimizer. A variable learning rate strategy is adopted: the learning rate is set to 10^{-3} for the first 25 epochs, then reduced to 10^{-4} for the epochs from 26 to 35, and further decreased to 10^{-5} for the final five epochs. For the distance threshold in Equation (11), we set $\delta_d = 6$ in both the agent-to-lane feature fusion module and the dynamic traffic flow-to-agent feature fusion module, and $\delta_d = 100$ in the agent-to-agent interaction feature fusion module.

For prediction performance evaluation, average displacement error (ADE) and final displacement error (FDE) were adopted. ADE is the average L2 distance between the prediction trajectory and the ground truth, and FDE is the L2 distance between endpoint of the prediction trajectory and the ground

truth. In this study, minimum ADE (minADE) and minimum FDE (minFDE) at $K = 1$ and $K = 6$, respectively, were used as evaluation metrics to obtain the performance of the best predicted trajectory under single- and multi-modal cases.

4.3. Comparison experiment

To validate the proposed model, its results were compared to those of models proposed by other excellent studies conducted in recent years:

- Nearest-neighbor (NN) regression (Argoverse Baseline) [41]: The baseline builds on NN and prunes the number of predicted trajectories based on how often they exit the drivable area.
- Target-driven trajectory (TNT) [8]: A prediction framework that contains three training stages.
- VectorNet [9]: A hierarchical graph neural network that first exploits the spatial locality of individual road components represented by vectors, then it models the high-order interactions among all components.
- LaneGCN [11]: A lane-graph CNN that captures the complex topology and long-range dependencies of the lane graph.
- LaneRCNN [24]: A graph-centric motion prediction model that learns each participant's LaneRol to encode its past motion and local map topology.
- mmTransformer [42]: A neural network prediction framework based on the stack Transformer structure models multi-modality at the feature level through a fixed set of independent proposals.
- DenseTNT [43]: An anchor-free, end-to-end trajectory prediction model that directly outputs a set of trajectories from dense goal candidates.
- HOME [44]: A framework that tackles the motion forecasting problem with an image output representing the probability distribution of the agent's future location.
- Autobot [45]:It is an encoder-decoder architecture that generates scene-consistent multi-agent trajectories based on the Latent Variable Sequential Set Transformer. Trajectory prediction is achieved by alternately perform equivariant processing across the temporal and social dimensions.
- DRTR: The model proposed in this study.

Table 1 presents the minFDE and minADE for $K = 1$ and $K = 6$. We see that the performance of the DRTR was slightly worse than that of DenseTNT on $K = 1$ minFDE, and the proposed method performed best comprehensively with multiple performance metrics.

VectorNet, LaneGCN, and DRTR are all based on vector representations. VectorNet solely connects features through graph convolutional networks. LaneGCN encodes agents' historical trajectories using a feature pyramid network and fuses features between agents and lane lines through an attention mechanism. The model proposed in this research demonstrates significant improvements across all metrics compared to the previous two models. A crucial factor contributing to these enhancements is the relational reasoning network introduced in this study, which pre infers the relationships between various features. This deeper learning of the relationships between feature vectors serves as explicit criteria for the neural network to select features, making the neural network more comprehensive, adequate, and efficient in feature fusion and propagation. Additionally, the decoding of multimodal trajectories in this study employs randomly

initialized anchors for progressive refinement, ensuring the predictions are diverse yet plausible.

Table 1. Metrics of each model in the 3 s prediction horizon on the Argoverse test set.

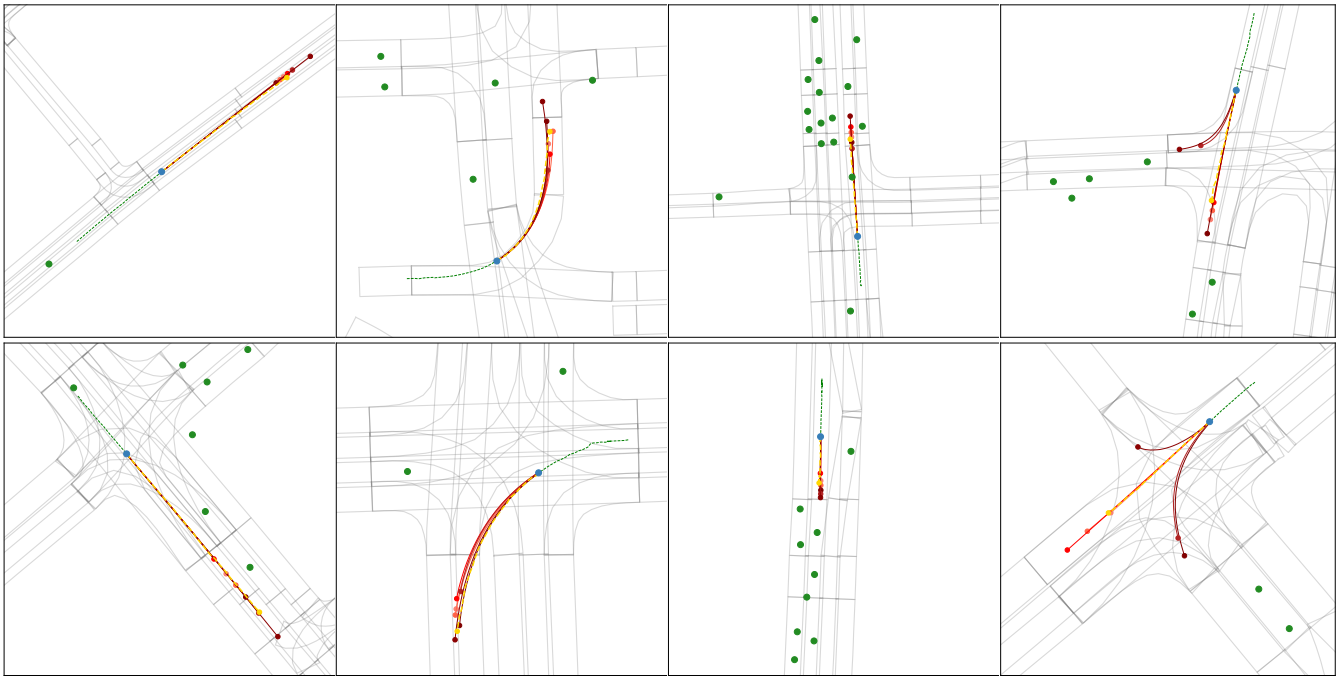
Model	K = 6		K = 1	
	min-ADE	min-FDE	min-ADE	min-FDE
NN (baseline) [41]	1.713	3.287	3.455	7.883
VectorNet [9]	3.11	6.723	3.11	6.723
LaneRCNN [24]	0.904	1.453	1.685	3.692
HOME [44]	0.89	1.292	1.699	3.681
TNT [8]	0.91	1.446	2.174	4.959
DenseTNT [43]	0.882	1.282	1.679	**3.632**
LaneGCN [11]	0.87	1.362	1.702	3.762
mmTransformer [42]	0.844	1.338	1.774	4.003
Autobot [45]	0.876	1.372	1.84	4.102
DRTR (Ours)	**0.825**	**1.267**	**1.664**	3.667

Note: VectorNet is a unimodal baseline. Therefore, the results under $K = 6$ coincide with those under $K = 1$, and do not indicate explicit multimodal distribution learning. We keep the $K = 6$ column to compare methods under the same output cardinality.

Autobot, mmTransformer, and DRTR all employ a framework similar to the DETR decoder for addressing multimodal trajectory prediction challenges, which involves multiple query vectors (random anchors) that participate in scene encoding and trajectory decoding. A review of recent studies indicates that Transformers perform well in extracting and fusing multi-source information. Both Autobot and mmTransformer utilize the original Transformer architecture, with mmTransformer additionally employing a specific region-based training strategy to stack Transformer blocks. DRTR enhances the anchor refinement process by employing the proposed relational reasoning network, which allows each modality to fully focus on the features critical to that modality, ensuring an efficient and reasonable refinement process. In the initial feature fusion stage, DRTR performs dynamic feature fusion, maintaining three groups of features: dynamic traffic flow, dynamic agents, and dynamic agent interaction features. This approach not only ensures the relevance of the features but also preserves their independence. The combined effect of these modules reduces the complexity of model training and results in superior performance on multimodal trajectory prediction tasks.

Figure 3 illustrates predictive instances from the Argoverse validation set in a bird's-eye view format. For simplicity, only the prediction results for a single agent are displayed. Figure 3a demonstrates the prediction scenario where the agent continues in a straight line. Despite being at an intersection, the agent's history of high speed makes a turning maneuver unlikely, hence all modalities predict continued straight-line movement. Figure 3b shows two images at a crossroads where the historical trajectory already exhibits clear turning characteristics, hence the model accurately predicts the turning intent and the trajectories quite precisely. Figure 3c features scenarios with numerous agents in front of the target

agent, involving significant interaction. Relying on a robust relational reasoning module, the model accurately captures the interactions among agents, resulting in effective predictions in such scenarios. For example, the graph below Figure 3c accurately predicts the agent's deceleration behavior. Figure 3d portrays a more complex intersection scenario. The historical trajectory does not indicate future directions. In the upper image, the target agent is on a lane designated for going straight or turning right, leading the model to provide multimodal predictions for both straight and right-turn movements. The lower image shows the target agent in a lane that allows for going straight, turning left, or turning right, thereby producing three distinct predictions: straight, left turn, and right turn, and the predicted trajectory is very accurate.



(a) **(b)** **(c)** **(d)**

Figure 3. Prediction instances from the bird's-eye view of Argoverse validation dataset. The blue dot represents the agent being discussed, the green dots represent surrounding agents, the green dashed lines represent historical trajectories, the yellow dashed lines represent the actual future trajectories, and the six red solid lines represent the trajectories predicted by the network across six different modalities.

4.4. Ablation experiment

In order to verify the performance of each module proposed in this model, an ablation experiment was performed. Here we define the feature encoding of the agent in the encoder as A, the feature encoding of the map as M, the feature fusion module from the agent to the lane line as A2M, dynamic traffic flow features to agents as M2A, the fusion of interaction features between agents as A2A, the fusion of trajectory proposals and target agent as A2Q, the fusion of trajectory proposals and dynamic traffic flow as M2Q, and the fusion of trajectory proposals and interaction features as I2Q:

- Non-Trajectory Query Network: in the feature fusion stage, first complete the feature fusion of A2M and M2A, then use the output of M2A as input to perform A2A feature fusion, and finally the decoder decodes the output of A2A to obtain the predicted trajectory;

- Non-Dynamic Scene Closed-Loop Modeling: A2M is removed and dynamic traffic flow is not output. M2A and M2Q in the trajectory query network directly use the output of map encoding in the encoder;
- Non-Feature Fusion Module: the trajectory query module directly uses the output of the encoder;
- Full Feature Network: the full feature network directly adds context features to the base features and normalizes them. It is specifically divided into a replacement feature fusion module, a replacement trajectory query module, and replace both;
- Full Elements: directly select all agents and lane lines in the scene;
- Complete Model proposed in this study

Table 2 shows the metrics of ablation in the 3-s prediction horizon using the Argoverse validation set. The following inferences were obtained based on these results:

Table 2. Metrics of the ablation model in the 3 s prediction horizon for the Argoverse validation set.

Encoder		**Feature Fusion**			**Trajectory Query**			**K = 6**		**K = 1**	
A	M	A2M	M2A	A2A	A2Q	M2Q	I2Q	FDE	ADE	FDE	ADE
✓	✓	✓	✓	✓				1.051	0.699	**2.919**	**1.327**
✓	✓				✓	✓	✓	1.089	0.71	3.083	1.397
✓	✓		✓	✓	✓	✓	✓	1.045	0.694	2.953	1.35
✓	✓	FF	FF	FF	✓	✓	✓	1.073	0.701	3.011	1.369
✓	✓	✓	✓	✓	✓	FF	FF	1.049	0.698	2.965	1.348
✓	✓	FF	FF	FF	✓	FF	FF	1.179	0.747	3.41	1.528
✓	✓	FE	FE	FE	✓	FE	FE	1.097	0.721	3.124	1.439
✓	✓	✓	✓	✓	✓	✓	✓	**1.042**	**0.691**	2.939	1.344

Note: FF represents Full Features, FE represents Full Elements.

Comparison between the Non-Trajectory Query Network and the Complete Model: The Non-Trajectory Query Network achieves the best performance under unimodal conditions, while the Complete Model has the best performance under multimodal scenarios. In the multimodal framework, the multimodal trajectory of the complete model is gradually refined by trajectory recommendations. These anchors are unaffected by variations in scene dynamics or inherent qualities of the anchors themselves. Each anchor represents a specific type of behavior, allowing for targeted decoding based on the outcome of feature fusion. This level of targeted behavior representation is absent in the Non-Trajectory Query Network. Under unimodal conditions, despite the Complete Model still operating under the guidance of a single anchor, it is challenging and impractical to expect one anchor to decode various trajectories in the real world. By removing the trajectory query network, the model decodes directly from the output of feature fusion without these constraints, making learning easier in unimodal. This approach also corroborates the design intent behind using multiple anchors, confirming their diversity where each anchor symbolizes a distinct behavioral pattern (and corresponding trajectory). Overall, the trajectory query module underscores the inherent uncertainties of future events and the diversity of actions, offering a distinct advantage in tackling multimodal trajectory predictions.

Comparison of Models with Non-Feature Fusion Network, Non-Dynamic Scene Closed-Loop Modeling, and Complete Model: Without the feature fusion module, the entire model lacks the capability to extract interaction information between agents and maps, and between agents themselves. Trajectory queries rely solely on independent agent and map information, resulting in significant performance degradation. Removing A2M leads to a performance improvement compared to Non-Feature Fusion Module, yet the map features do not integrate the agents' motion and location information, failing to reflect the relevance of agents to the map. This absence of integration does not support the output of dynamic traffic flow information, rendering the map data static. Consequently, when executing M2A and M2Q, the map features are static, and the model's feature fusion does not form a closed loop, leading to a decline in performance. Therefore, fusing context features of the scene is essential, and performing dynamic closed-loop modeling enhances the expression of feature interrelations, providing robust feature selection for subsequent trajectory queries.

Comparison of Full Feature Network, Non-Feature Fusion Network, and Non-Trajectory Query Network: The poorest performance was observed when both the feature fusion and trajectory query modules operated on Full Feature Network. This outcome is due to the presence of abundant extraneous information in complex urban road scenarios, while crucial information is particularly vital. This complexity increases the learning burden on the network, adversely affecting predictive performance. When the feature fusion and trajectory query network utilize feature selection networks individually, there is a notable improvement in performance, confirming that useful features are successfully screened in module. It is also evident that whether comparing the Non-Feature Fusion Network with Full Feature Fusion Network or comparing Non-Trajectory Query Network with Full Feature Trajectory Query Network, the latter shows enhanced performance in both cases. This improvement occurs because, although full feature selection introduces irrelevant features, the robust learning capabilities of neural networks still manage to express useful features effectively. Therefore, feature selection based on relational reasoning guided by the prior knowledge of distance and the hidden feature of feature correlation can learn more accurate features.

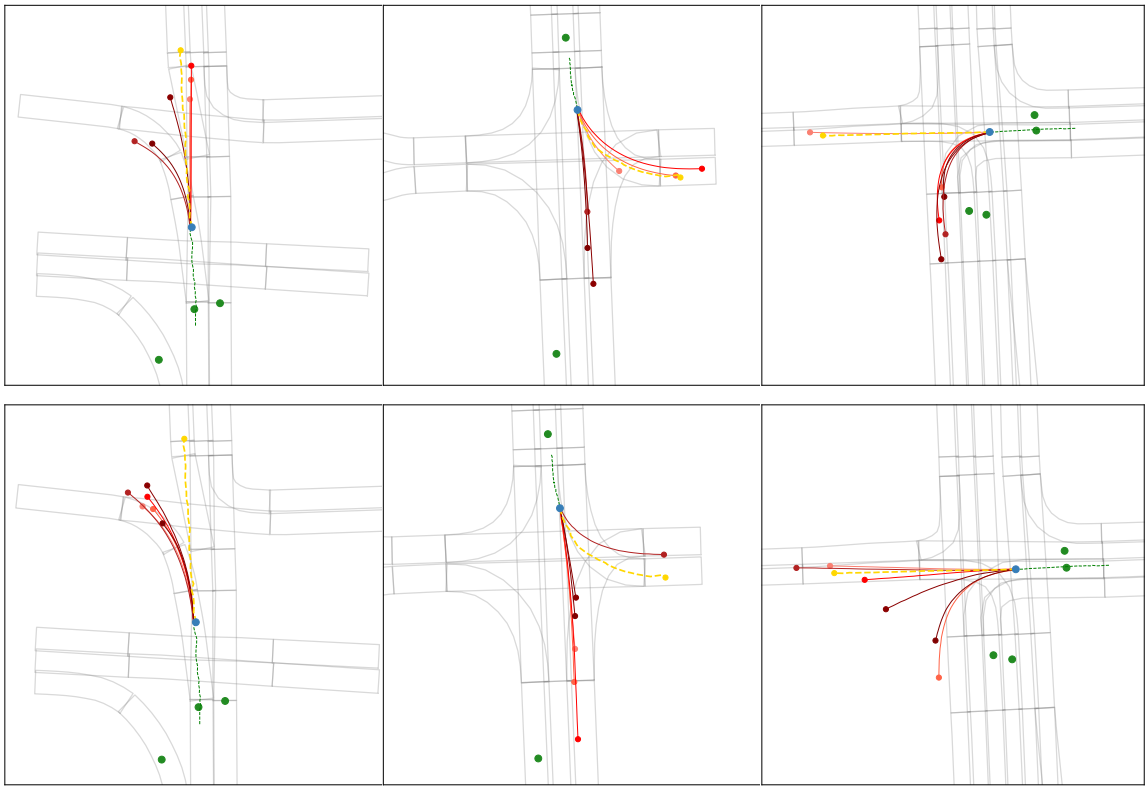
Comparison of Full Element Network, Full Feature Network, and Complete Modal: The Full Element Network, which explicitly incorporates elements that do not impact the target vehicle within the scene, such as vehicles far from the target agent, shows a performance decline. This highlights the necessity for neural networks to eliminate irrelevant features. Despite introducing additional elements, the Full Element Network shows improved performance over the Full Feature Network. This is because, with the robust feature filtering capability of the feature selection network based on relational reasoning, it manages to discard ineffective information. This improvement indirectly validates the powerful feature filtering capability of the feature selection network based on relational reasoning, underscoring its effectiveness in enhancing overall model performance by focusing on relevant features.

Figure 4 visualizes several instances of the Argoverse validation set to qualitatively analyze the prediction results of the full model and the trajectory-removed query network.

As illustrated in Figure 4, intersections present complex traffic scenarios where the future trajectories of agents are highly uncertain due to potential behaviors such as going straight or turning. The absence of a trajectory query network and an overly simplistic decoder, leads to inaccurate or even implausible trajectory

predictions. In contrast, our complete model, guided by trajectory suggestions and refined by scene features, can accurately predict various behaviors and trajectories, including precise correct predictions.

In the scenario depicted in Figure 4a, the historical trajectory indicates a tendency to turn; hence, the model without a trajectory query network predicts turning for all six modalities. However, at such an intersection, there is a significant possibility that the agent might continue straight. Accordingly, the complete model correctly identifies this possibility. The scenario in Figure 4b is similar to a; although the agent does indeed turn, the model without the trajectory query network fails to adequately consider the constraints of lane lines (traffic rules), resulting in an unreasonable prediction of the turn. Meanwhile, the complete model accurately predicts the lane to be taken and the final position. In Figure 4c, the historical trajectory does not clearly indicate whether the agent will go straight or turn. Both models offer various suggestions, but it is evident that the turning suggestions provided by the model without a trajectory query network are highly unreasonable, even veering out of the lane. In contrast, the complete model not only provides more accurate predictions for going straight, but its suggested turning trajectories are also reasonable.



(a) **(b)** **(c)**

Figure 4. Compare Complete Module and Non-Trajectory Query under predicted instances in the bird's eye view of the Argoverse validation dataset. The blue dot represents the agent being discussed, the green dots represent surrounding agents, the green dashed lines represent historical trajectories, the yellow dashed lines represent the actual future trajectories, and the six red solid lines represent the trajectories predicted by the network across six different modalities. The three pictures above are the complete model, and the three pictures below are the Non-Trajectory Query Module.

5. Conclusion

In this paper, we introduce DRTR, a novel trajectory prediction model comprising an encoder, feature fusion network, multimodal trajectory query network, and decoder. This model addresses the challenges of inadequate scene modeling, inefficient feature utilization, and difficulties in multimodal trajectory prediction. In the feature fusion network, we dynamically close-loop fuse scene features and output dynamic traffic flow, dynamic agent, and agent interaction information, ensuring the relevance and independence of each feature. The multimodal trajectory query network starts with randomly initialized anchors and refines step-by-step based on the output from the feature fusion network, ensuring rational and accurate multimodal trajectory predictions. Both modules employ the feature selection network based on relational reasoning, which uses distance as prior knowledge and feature correlation as a latent feature for relational reasoning, and selects useful features based on these inferences, allowing the entire model to effectively utilize valid features and enhance its expressiveness. On the Argoverse 1 dataset, this model outperforms other trajectory prediction models. Ablation studies validate the effectiveness of the proposed modules.

Despite the promising performance of the proposed method, there are limitations. The Argoverse 1 dataset contains few examples of highly interactive or extremely complex scenes, which does not fully validate the model's effectiveness in highly complex scenarios, and the model's inference speed is also not fast. In the future, we plan to extend the network to more complex interactive systems and improve the model to enhance inference speed. DRTR does not explicitly consider scene consistency; ensuring the mutual compatibility of predicted multi-agent trajectories remains an important research direction for future work. Moreover, the iterative query-based multimodal refinement introduces additional computation, and future work will focus on improving inference speed while preserving multimodal quality.

Data availability statement

The data supporting the findings of this study are openly available in the Argoverse repository. The dataset can be accessed directly through the official website (https://www.argoverse.org/).

Declaration of generative AI and AI-assisted technologies

During the preparation of this work, the author(s) used GPT-4 solely to polish the language and improve the overall readability of the manuscript. After using these tools, the authors thoroughly reviewed and edited the content as necessary and take full responsibility for the final text and the accuracy of the publication.

Acknowledgments

This work was supported by the Science and Technology Development Project of Jilin Province under Grant No. 20260601002RC, and by the 2025 Graduate Innovation Fund of Jilin University under Grant No. 2025CX153.

Authors' contribution

Conceptualization, resources, writing—review and editing and supervision, Hongyu Hu; methodology, software, formal analysis, writing—original draft, Linwei Song; validation, visualization, software, Zhengyi Li; conceptualization, methodology, project administration, writing—review and editing, Rui Zhao; data curation, visualization, software, Zhonghua Xiong; visualization, writing—review and editing, Zhiwen Wei. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

Hongyu Hu holds the position of Associate Editor for *Artificial Intelligence and Autonomous Systems* and has not peer reviewed or made any editorial decisions for this paper.

References

[1] Zhao R, Yuan Q, Li J, Wang Z, Li Y, *et al.* VLM-Driver: human-like autonomous driving decision-making via vision language model. *IEEE Trans. Veh. Technol.* 2025, 75(5):7327–7342.

[2] Chen G, Gao Z, Hu H, Du J, Wang H. Multi-maneuver vertical parking trajectory planning and tracking control in narrow environments. *Automot. Innovation* 2024, 7(2):300–311.

[3] Hu H, Qiao X, Wei Z, Peng D, Tian C, *et al.* Driver interaction intent orediction with dynamic scene semantic fusion for intelligent cockpit. *IEEE Trans. Ind. Inf.* 2026, 22(4):3550–3561.

[4] Deo N, Rangesh A, Trivedi MM. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *IEEE Trans. Intell. Veh.* 2018, 3(2):129–140.

[5] Zhou Z, Wang J, Li Y, Huang Y. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Vancouver, Canada, June 18–22, 2023, pp. 17863–17873.

[6] Lan Z, Jiang Y, Mu Y, Chen C, Li SE. Towards efficient scene representation learning for motion prediction. In *Proceedings of The Twelfth International Conference on Learning Representations*, Vienna, Austria, May 7–11, 2024.

[7] Varadarajan B, Hefny A, Srivastava A, Refaat KS, Nayakanti N, *et al.* MultiPath++: efficient information fusion and trajectory aggregation for behavior prediction. In *Proceedings of 2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA, May 23–27, 2022, pp. 7814–7821.

[8] Zhao H, Gao J, Lan T, Sun C, Sapp B, *et al.* TNT: target-driven trajectory prediction. In *Proceedings of Conference on Robot Learning*, Virtual Conference, November 16–18, 2021, pp. 895–904.

[9] Gao J, Sun C, Zhao H, Shen Y, Anguelov D, *et al.* VectorNet: encoding HD maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 14–19, 2020, pp. 11525–11533.

[10] Jia X, Wu P, Chen L, Liu Y, Li H, *et al.* HDGT: heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *IEEE Trans. Pattern Anal. Mach.* 2023, 3(2):13860–13875.

[11] Liang M, Yang B, Hu R, Chen Y, Liao R, *et al.* Learning lane graph representations for motion forecasting. In *Proceedings of Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, Glasgow, UK, August 23–28, 2020, pp. 541–556.

[12] Quan R, Zhu L, Wu Y, Yang Y. Holistic LSTM for pedestrian trajectory prediction. *IEEE Trans. Image Process.* 2021, 30:3229–3239.

[13] Sadeghian A, Kosaraju V, Sadeghian A, Hirose N, Rezatofighi H, *et al.* SoPhie: an attentive GAN for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 15–20, 2019, pp. 1349–1358.

[14] Lee N, Choi W, Vernaza P, Choy CB, Torr PHS, *et al.* DESIRE: distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, July 21–26, 2017, pp. 336–345.

[15] Hong J, Sapp B, Philbin J. Rules of the road: predicting driving behavior with a convolutional model of semantic interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 15–20, 2019, pp. 8454–8462.

[16] Cui H, Radosavljevic V, Chou FC, Lin TH, Nguyen T, *et al.* Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *Proceedings of 2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 20–24, 2019, pp. 2090–2096.

[17] Phan-Minh T, Grigore EC, Boulton FA, Beijbom O, Wolff EM. CoverNet: multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, December 8–14, 2019, pp. 15398–15408.

[18] Chai Y, Sapp B, Bansal M, Anguelov D. MultiPath: multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv* 2019, arXiv:1910.05449.

[19] Fang L, Jiang Q, Shi J, Zhou B. TPNet: trajectory proposal network for motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 14–19, 2020, pp. 6797–6806.

[20] Pan J, Sun H, Xu K, Jiang Y, Xiao X, *et al.* Lane-Attention: predicting vehicles' moving trajectories by learning their attention over lanes. In *Proceedings of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, October 24–30, 2020, pp. 7949–7956.

[21] Ma Y, Zhu X, Zhang S, Yang R, Wang W, *et al.* TrafficPredict: trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, USA, January 27–February 1, 2019, pp. 6120–6127.

[22] Da F, Zhang Y. Path-aware graph attention for HD maps in motion prediction. In *Proceedings of 2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA, May 23–27, 2022, pp. 6430–6436.

[23] Mohamed A, Qian K, Elhoseiny M, Claudel C. Social-STGCNN: a social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, USA, June 14–19, 2020, pp. 14424–14432.

[24] Zeng W, Liang M, Liao R, Urtasun R. LaneRCNN: distributed representations for graph-centric motion forecasting. In *Proceedings of 2021 IEEE/RSJ International Conference on Intelligent Robots*

and Systems (IROS), Prague, Czech Republic, September 27–October 1, 2021, pp. 532–539.

[25] Jia X, Sun L, Tomizuka M, Zhan W. Ide-Net: interactive driving event and pattern extraction from human data. *IEEE Robot. Autom. Lett.* 2021, 6(2):3065–3072.

[26] Li LL, Yang B, Liang M, Zeng W, Ren M, *et al.* End-to-end contextual perception and prediction with interaction transformer. In *Proceedings of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, October 24–30, 2020, pp. 5784–5791.

[27] Mercat J, Gilles T, El Zoghby N, Sandou G, Beauvois D, *et al.* Multi-head attention for multi-modal joint vehicle motion forecasting. In *Proceedings of 2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 31–August 31, 2020, pp. 9638–9644.

[28] Gupta A, Johnson J, Li F, Savarese S, Alahi A. Social GAN: socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, June 18–22, 2018, pp. 2255–2264.

[29] Salzmann T, Ivanovic B, Chakravarty P, Pavone M. Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, Glasgow, UK, August 23–28, 2020, pp. 683–700.

[30] Yuan Y, Weng X, Ou Y, Kitani KM. AgentFormer: agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, Canada, October 11–17, 2021, pp. 9813–9823.

[31] Messaoud K, Yahiaoui I, Verroust-Blondet A, Nashashibi F. Relational recurrent neural networks for vehicle trajectory prediction. In *Proceedings of 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, October 27–30, 2019, pp. 1813–1818.

[32] Azadani MN, Boukerche A. STAG: a novel interaction-aware path prediction method based on Spatio-Temporal Attention Graphs for connected automated vehicles. *Ad Hoc Networks* 2023, 138:103021.

[33] Liao H, Li Z, Shen H, Zeng W, Liao D, *et al.* Bat: behavior-aware human-like trajectory prediction for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vancouver, Canada, February 20–27, 2024, pp. 10332–10340.

[34] Liao H, Li Y, Li Z, Wang C, Cui Z, *et al.* A cognitive-based trajectory prediction approach for autonomous driving. *IEEE Trans. Intell. Veh.* 2024, 9(4):4632–4643.

[35] Xu C, Li M, Ni Z, Zhang Y, Chen S. GroupNet: multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, June 19–24, 2022, pp. 6498–6507.

[36] Xu Q, Mao W, Gong J, Xu C, Chen S, *et al.* Joint-Relation transformer for multi-person motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Paris, France, October 2–6, 2023, pp. 9816–9826.

[37] Hu H, Wang Q, Zhang Z, Li Z, Gao Z. Holistic transformer: a joint neural network for trajectory prediction and decision-making of autonomous vehicles. *Pattern Recognit.* 2023, 141:109592.

[38] Hu H, Xiong Z, Li Z, Sun T, Ran R. Semi-supervised Risk assessment research for intelligent vehicles inspired by collective biological risk-avoidance behaviors. *J. Bionic Eng.* 2026, 23(1):225–238.

[39] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, *et al.* Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, Long Beach, USA, December 4–9, 2017, pp. 5998–6008.

[40] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, *et al.* End-to-end object detection with transformers. In *Proceedings of European Conference on Computer Vision*, Glasgow, UK, August 23–28, 2020, pp. 213–229.

[41] Chang MF, Lambert J, Sangkloy P, Singh J, Bak S, *et al.* Argoverse: 3D tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, June 15–20, 2019, pp. 8748–8757.

[42] Liu Y, Zhang J, Fang L, Jiang Q, Zhou B. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Virtual Conference, June 19–25, 2021, pp. 7577–7586.

[43] Gu J, Sun C, Zhao H. DenseTNT: end-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Montreal, Canada, October 11–17, 2021, pp. 15303–15312.

[44] Gilles T, Sabatini S, Tsishkou D, Stanciulescu B, Moutarde F. HOME: heatmap output for future motion estimation. In *Proceedings of 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, USA, September 19–22, 2021, pp. 500–507.

[45] Girgis R, Golemo F, Codevilla F, Weiss M, D'Souza JA, *et al.* Latent variable sequential set transformers for joint multi-agent motion prediction. *arXiv* 2021, arXiv:2104.00563.