1. Introduction

Due to the complexity of traffic scenarios, the motion of agents is influenced not only by road geometry and traffic rules but also by surrounding agents, making trajectory prediction for autonomous vehicles exceptionally challenging. Accurate and timely trajectory prediction is crucial for downstream decision-making and traffic planning, thus significantly enhancing the safety of autonomous vehicles. Recently, scholars [1, 2, 3] adopted complex network architectures to integrate heterogeneous information in different formats, including agent historical trajectories, inter-agent interactions, and agent-map interactions. To further explore the features of heterogeneous information, scholars [4] have proposed three different stages of integrating heterogeneous information. Additionally, to account for future uncertainties, recent studies [5, 6, 7] have shifted towards multimodal trajectory prediction. To facilitate the generation of multimodal trajectories, methods based on multiple candidate agents as anchors [8, 9] and learnable query methods [10] can generate various possible prediction results.

The movement pattern of a single vehicle is typically influenced by nearby vehicles and its surrounding environmental information. Social psychologists have pointed out that individuals often imitate or follow other members of a group [11], using them as a reference for their behavior, which leads to the frequent occurrence of the herd effect in vehicle movement patterns [12]. A common example is a group of multiple vehicles, where the movement of the vehicle group is influenced by each member, each of which has similar movement patterns and close destinations [13], while the group of vehicles collectively gathers surrounding information. Unlike individual behavior, such a group can maintain a relatively stable formation, which is significantly helpful for predicting future trajectories [14].

To account for group relationships, some researchers [14] have employed multi-scale hypergraph neural networks among groups, integrating interaction intensity and types to capture the interactions between multiple agents. Other researchers [15] have utilized motion coherence [16] to cluster target groups with similar movement trajectories and transmit their hidden states through shared group information in the social pooling layer. However, these methods tend to be homogeneous, disregarding the diversity and individual differences within the group that affect group behavior. Furthermore, they primarily focus on pedestrian groups, overlooking the exploration of the trajectories of the vehicle groups. Current methods generally have the following issues: (1) Focusing only on individual agent feature information, repeatedly encoding and fusing the features of a single agent to achieve interaction between agents. However, this interaction is essentially independent. Although these methods can implicitly capture relationships between individuals, they are insufficient for the explicit representation of group behavior. Learning only individual-level interactions within the group, without encoding their group affiliations and future paths, can lead to decreased prediction accuracy and increased network parameters. (2) Inadequate fusion of heterogeneous information. Single attention fusion mechanisms are ineffective for addressing variable map scenes and complex interaction relationships, making accurate and timely decision-making more difficult. Existing work struggles to bridge the semantic gap from heterogeneous information, thus failing to extract meaningful information. To address these issues, we propose a trajectory prediction model for autonomous vehicles based on a grouped spatial-temporal encoder (GSTEP) building upon existing methods, with the following contributions:

- We propose a grouped spatial-temporal encoder to simultaneously encode group information and spatial-temporal features, capturing deep commonalities among the group of vehicles. By dividing agents into different groups, it captures interactions both within and between groups, thereby achieving precise trajectory prediction.
- We propose an internal and external synergistic perception fusion module (**IESP**) for more comprehensive information fusion. The internal and external synergistic perception mechanism considers both intrinsic and extrinsic factors, better capturing the interaction between agents and the environment, resulting in a more complete fusion mechanism and significantly improving model performance. By fully considering the relationship between agents and the environment, it accurately predicts future vehicle trajectories, avoiding potential collisions and better handling various uncertainties and sudden situations.
- We evaluated the proposed method on the public motion prediction datasets Argoverse 1 [17] and Argoverse 2 [18]. Experimental results show that despite its simplified design, **GSTEP** outperforms other state-of-the-art methods and baseline methods. More importantly, **GSTEP** achieves efficient multi-agent trajectory prediction with fewer learnable parameters without sacrificing performance, making it highly suitable for practical application deployment. We emphasize that **GSTEP**, as a robust model, exhibits exceptional scalability. Its streamlined architecture facilitates the direct integration of the latest advancements in the motion prediction field, thereby offering opportunities for further performance enhancement.

2. Related Work

In the field of autonomous driving trajectory prediction, current research primarily focuses on how to effectively ex-

tract and integrate information from the environment and historical trajectories to achieve accurate future trajectory prediction. Two notable techniques among these methods that have garnered widespread attention are context encoding and the attention mechanism.

2.1. Context Encoding

Driving context is typically divided into two categories: the historical trajectories of surrounding agents and static map information. Trajectories, represented as time series data, are commonly encoded utilizing temporal neural networks. For map features, early studies commonly represented them as multi-channel bird's-eye images, dividing different semantic elements into different channels and then using convolutional neural networks (CNNs) for feature fusion. Dai et al. [19] used two sets of long short-term memory (LSTM) networks to predict the trajectory of the target agents. One set modeled the trajectories of surrounding agents, while the other modeled the interactions between surrounding agents. Nikhil et al. [20] argued that compared to CNNs, recurrent neural networks (RNNs) are more advantageous in trajectory prediction because trajectories have strong spatial-temporal continuity. They employed a sequence-to-sequence structure, using historical trajectories as input. Convolutional layers were stacked on fully connected layers to maintain temporal continuity, and future trajectories were generated via fully connected layers. However, rasterization methods inevitably result in information loss and a limited field of view. To address these issues, researchers have proposed vector-based methods [21, 22], which are increasingly being adopted in studies. While some methods employing RNNs and CNNs have achieved significant success in extracting Euclidean spatial data features, the vehicle trajectory prediction scenario encompasses numerous non-Euclidean interaction relationships, which are effectively addressed by graph neural networks (GNNs). Li et al. [21] proposed a trajectory prediction model based on graph convolutional networks (GCNs), considering each agent at each sampling time point as a node and accounting for interaction factors. Subsequently, Li et al. [22] proposed an improved model, using fixed and dynamic graph networks to enhance generalization capability in complex scenarios. Benz [23] first applied high-definition maps to trajectory prediction and performed map topology based on lane information associated with agents to predict their future trajectories. However, it did not consider the interaction relationships between agents. Since the introduction of the Argoverse dataset [17], researchers have employed GNNs to extract interaction features among agents and between agents and maps, thereby improving the accuracy of trajectory prediction. Gao et al. [24] modeled agents and vector maps within the scene as nodes and introduced VectorNet, a GNN-based approach for trajectory prediction. Liang et al. [25] used CNNs to extract features of agents and GCNs to extract lane features from vector maps, then combined these features for trajectory prediction. However, these encoding methods overlook the group commonality among agents, treating each agent as an individual traffic participant. The latest research on group commonality has made good progress in studies of pedestrians [15, 26, 27], but due to the different dynamic constraints and spatial-temporal characteristics of the subject, it is still difficult to apply in vehicle studies.

Group encoding is an imperative part of trajectory prediction, as it captures contextual features and spatial information from the agents' dynamic group perception representations. One widely employed method for group encoding is the social grouping approach, which characterizes the behavioral differences between agents within a group and those belonging to other groups. Xu et al. [28] proposed a graph network model, which used a stacked graph convolution module to extract the global spatial-temporal features of vehicle historical trajectory data, and encoded the obtained graph features based on the seq2seq network to realize the trajectory prediction of road vehicles at different times in the future. Zhao et al.[29] proposed a graph-based information-sharing network (GIS-Net), which could encode the historical trajectory of vehicles and share the information with the surrounding vehicles. However, none of these methods distinguish the different groups in a suitable way, nor do they have three diverse interactions. In fact, these methods are similar to most pedestrian trajectory prediction methods in that only intra-group shared information exists[30]. Therefore, we propose a grouped spatial-temporal encoder to study the commonality of the vehicle group. By using appropriate grouping methods and pooling techniques, vehicles are assigned to suitable groups while considering vehicle differences, and the relationships between groups are aggregated. At the same time, the interaction relationships at the individual and group levels are learned, enabling the model to capture deep commonality.

2.2. Attention Mechanism

The attention mechanism has been widely applied in sequential tasks. Its advantage lies in its ability to enhance the significance of crucial data components. Recently, Tim et al. [31] proposed a graph-structured recurrent model for generating dynamic future trajectories. This model represents the scene as a directed spatial-temporal graph, intending to closely integrate with the planning system of autonomous vehicles. The Transformer [32] has achieved significant success in natural language processing and computer vision, largely due to the attention mechanism, which examines the entire context and focuses attention on important parts of the input data. Zhou et al. [2] proposed the hierarchical vector transformer, which de-

composes the trajectory prediction problem into local context extraction and global interaction modeling to achieve fast and accurate multi-agent motion prediction. Zhang et al. [33] proposed a heterogeneous polyline transformer with relative pose encoding, enabling asynchronous token updates during online inference and sharing context among agents. Additionally, many methods [34, 35] also adopt attention mechanisms to process agent historical sequences or establish agent-agent and agent-map interaction models, achieving significant success in trajectory prediction. However, trajectory prediction tasks typically include data across both temporal and spatial dimensions. Applying attention across multiple dimensions [2, 7] may be more suitable for trajectory prediction tasks, as it enhances semantic consistency and reduces computational complexity. In our work, the key component IESP module also utilizes this approach. By employing an attention mechanism akin to the Transformer structure, we integrate features across various dimensions of instances and update features through internal and external synergistic perception, thereby achieving excellent performance while reducing computational complexity.

3. Methodology

3.1. Problem Definition

The goal of trajectory prediction tasks is to generate possible future trajectories for target agents based on the observed motion history of moving objects and their surrounding information. Specifically, in a driving scenario with N_a moving agents, we use $\mathcal{M}$ to represent the map information and $X = \{\mathrm{x}_0, ..., \mathrm{x}_{N_a}\}$ to represent the observed trajectories of all agents. Each $\mathrm{x}_i = \{x_{i,-H+1}, ..., x_{i,0}\}$ represents the historical trajectory of the i-th agent across the past H time steps. Typically, a multi-agent trajectory predictor needs to generate possible future trajectories for all N_a agents in the scene, which can be represented as $Y = \{\mathrm{y}_0, ..., \mathrm{y}_{N_a}\}$. For each agent i, K possible future trajectories and their corresponding probability scores need to be predicted to capture the multimodal distribution. The multimodal trajectories are represented as $y_i = \{y_i^1, \ldots, y_i^K\}$, where each $y_i^k = \{y_{i,1}^k, ..., y_{i,T}^k\}$ represents the predicted trajectory of the i-th agent over T time steps in the k-th modal. The list of probability scores corresponding to each modal can be represented as $\alpha_i = \{\alpha_i^1, ..., \alpha_i^K\}$. Therefore, the multimodal trajectory prediction for agent i can be considered an estimation of a mixture distribution:

$$P(\mathrm{y}_i|X, \mathcal{M}) = \sum_{k=1}^{K} \alpha_i^k P(\mathrm{y}_i^k|X, \mathcal{M}) \tag{1}$$

3.2. Framework Overview

An overview of the proposed GSTEP method is presented in Fig.1. First, we use a vectorized scene representation to convert both map information and agent information into vector representations. Specifically, for each semantic instance (such as trajectories and lane lines), we construct multiple local encoding modules to decouple inherent features and relative information among instances. Subsequently, the grouped spatial-temporal encoder extracts features from agents and maps, calculates the relative poses of agents in pairs, and encodes them using a multilayer perceptron (MLP) to obtain relative pose embedding (RPE). The group encoding module is used to capture deep group features. First, it estimates grouping information through a group assignment network, then generates intra-group and inter-group graphs by masking irrelevant nodes and using group pooling strategies to capture deep interactions of agents with collective perception. The weights of inter-group, intra-group, and direct interactions are shared and fused, then concatenated with map information to form instance tokens. Then the instance tokens and RPE are sent to the proposed IESP module. With its compact and comprehensive characteristics, the model can use dual internal and external attention to cooperatively update and fuse features. Finally, the updated group features are fed into a motion decoder, consistent with the baseline, to obtain the predicted multimodal trajectory results.

3.3. Grouped Spatial-Temporal Encoder

After obtaining the relative poses, map information, and historical trajectories of the agents, we use relative pose encoding, map encoding, and group encoding methods to convert them into feature vectors. The first two encoders capture the spatial-temporal information of the agents, while the group encoder captures the group commonality of the agents.

3.3.1 Relative Pose Encoder

Based on the use of vectorized features to represent agents, we incorporate their relative poses to supplement the positional relationship information. Specifically, the average motion state of agent i in the global coordinate system over a certain period can be represented by its position $p_i \in \mathbb{R}^2$ and direction vector $v_i \in \mathbb{R}^2$. According to [36], we describe the relative pose between agent i and agent j using three quantities: heading difference as $\alpha_{i\to j}$, relative azimuth as $\beta_{i\to j}$, and Euclidean distance as $\parallel d_{i\to j} \parallel$. To enhance numerical stability, the angles are represented using their sine and cosine values due to their periodicity.

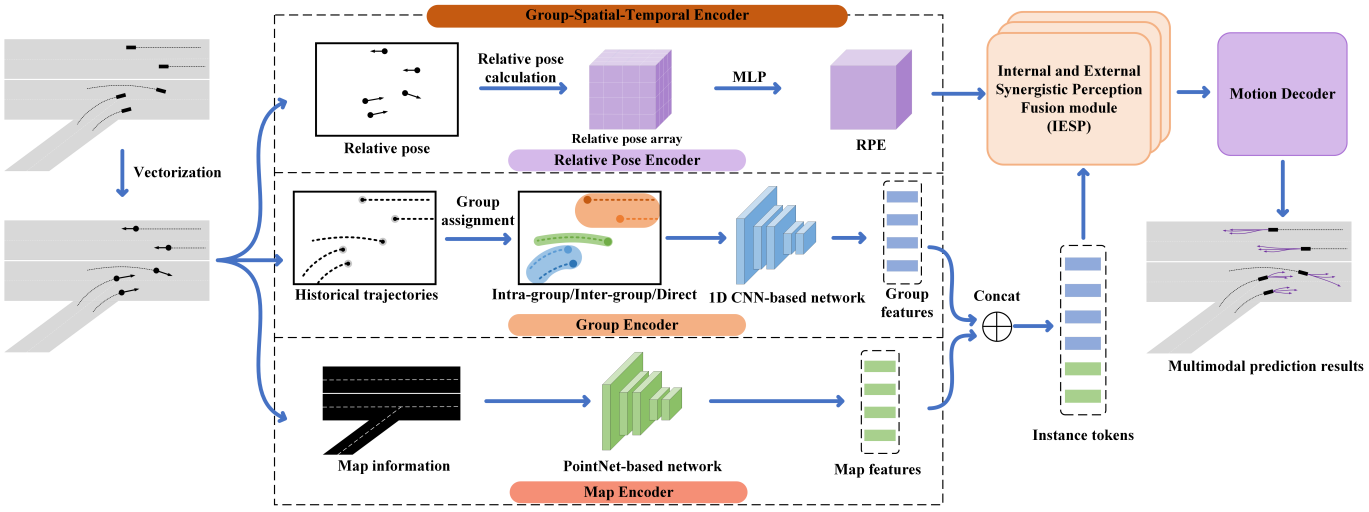

Figure 1: An overview of the proposed approach. The local features of the semantic instances are processed by the grouped spatial-temporal encoder, and the encoding results are input to the motion decoder after passing through the internal and external synergistic perception fusion module, which ultimately generates multimodal trajectory prediction results.

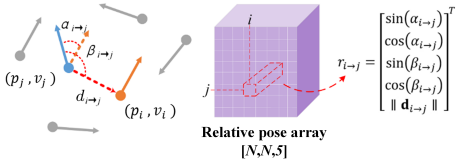

Figure 2: Schematic diagram of relative pose calculation. A typical scenario is shown on the left. The relative position between instances i and j can be described by the heading difference $\alpha_{i\to j}$, the relative azimuth $\beta_{i\to j}$ and the position distance $\|d_{i\to j}\|$. All relative positions are calculated and represented as a 3D array.

We define the heading difference $\alpha_{i\to j}$ as:

$$\begin{aligned}\sin(\alpha_{i\to j}) &= \frac{v_i \times v_j}{\|v_i\|\|v_j\|},\\ \cos(\alpha_{i\to j}) &= \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|}\end{aligned} \tag{2}$$

We define the relative azimuth $\beta_{i\to j}$ (the angle between displacement vector $d_{i\to j} = p_i - p_j$ and direction vector v_j) as:

$$\begin{aligned}\sin(\beta_{i\to j}) &= \frac{d_{i\to j} \times v_j}{\| d_{i\to j} \|\| v_j \|},\\ \cos(\beta_{i\to j}) &= \frac{d_{i\to j} \cdot v_j}{\| d_{i\to j} \|\| v_j \|}\end{aligned} \tag{3}$$

For simplicity, we omit the additional positional encoding process for the distance values used in [36], making the relative spatial-temporal information a 5-dimensional vector $r_{i\to j} = [\sin(\alpha_{i\to j}), \cos(\alpha_{i\to j}), \sin(\beta_{i\to j}), \cos(\beta_{i\to j}), \|d_{i\to j}\|]$. Therefore, given a scene containing $N = N_a + N_m$ semantic elements, where N_a is the number of participants and N_m is the number of map elements. The resulting relative pose information is an array with the shape of $[N, N, 5]$, where $r_{i\to j}$ is located at the j-th row and i-th column. Note that both temporal and spatial computations are involved in this encoding process, i.e. the process incorporates spatial-temporal information. An illustration of relative pose calculation is shown in Fig. 2.

3.3.2 Group Encoder

As shown in Fig.3, in the group encoding module, we first estimate grouping information using a group assignment module, then generate intra-group/inter-group graphs by masking irrelevant nodes and performing agent group pooling to capture socially aware interactions. Next, the graphs are fed into the target encoding module, the obtained inter-group/intra-group/direct interaction weights are shared, and inter-group features are unpooled and input into the group fusion module along with intra-group and direct interaction features. Finally, the output from the group fusion module is transmitted to the subsequent perception module to facilitate the comprehension of additional diverse information.

Group Assignment Module. In trajectory prediction, each node needs to retain its original identity index information and describe the dynamic attributes of group behavior in the scene, so that the original identity information can be restored in subsequent steps. We introduce both Euclidean distance and velocity direction into the grouping rules to enable the model to consider the

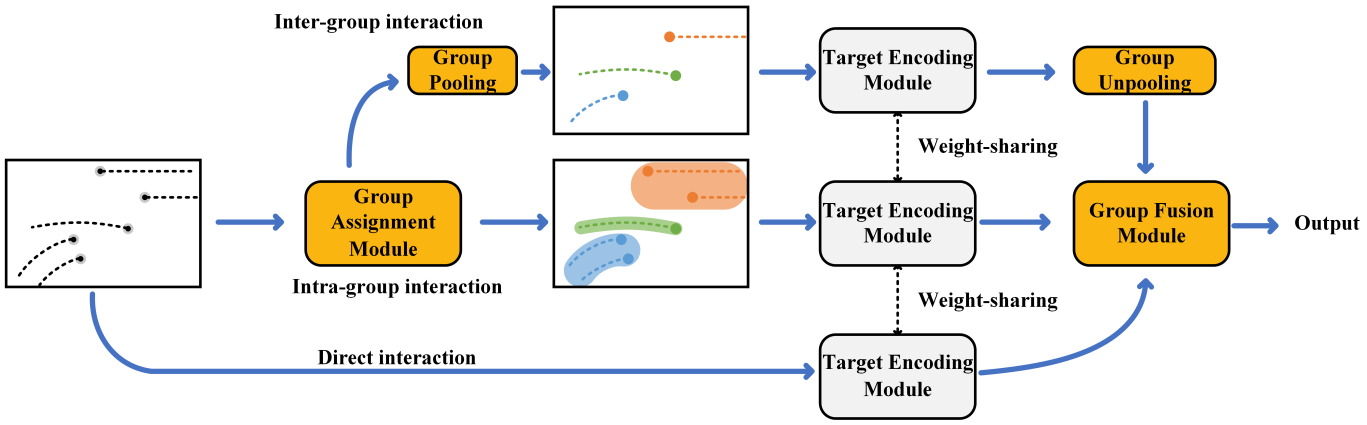

Figure 3: Schematic diagram of group encoding. Group commonality is captured through a triad of inter-group/intra-group/direct interaction.

effects of both speed and distance on grouping. Our research found that a group assignment module is needed to estimate the grouping information of the agents. It is known that distance information, speed difference information, and directional angle information between agents can all play a significant role in grouping. Therefore, we use each agent's historical trajectory, Euclidean distance, and speed to calculate the feature similarity between all pairs of agents. Based on the obtained similarity, all potential group members belonging to the same group can be selected. The pairwise distance matrix D and a set of group member indices P are defined as:

$$\begin{aligned} D_{ij} =& \parallel F_\varphi(X_i V_i) - F_\varphi(X_j V_j) \parallel \quad \text{for } i,j \in [1,2,...,N], \\ P =& \{pair(i,j) \mid i,j \in [1,2,...,N], i \neq j, D_{ij} \leq \pi\} \end{aligned} \tag{4}$$

Where F_φ is a learnable convolutional layer used to learn deep group features. X and V represent the agent's historical position information and velocity direction information, respectively. π is a learnable threshold parameter. $pair(\cdot,\cdot)$ represents agent pairs that may be in the same group.

Specifically, we calculate the mutual distance between each pair of agents to obtain the distance matrix D_{matrix} and the calculated velocity similarity matrix V_{matrix}. The feature similarity matrix is defined as: $F_{matrix} = D_{matrix} \cdot (1 - V_{matrix})$, where V_{matrix} is calculated based on the cosine similarity between the velocities of the agents. We provide the cosine similarity formula:

$$cos\vartheta = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{5}$$

Where A and B represent the velocity vectors of different agents, respectively. According to the cosine similarity formula, both the magnitude and direction of the velocities determine the cosine similarity. In other words, the more similar the velocities of the agents, the larger the result value, closer to 1. From the matrix representing the Euclidean distance between agent pairs, it can be inferred that the closer the agents are, the smaller the result value, closer to 0. To measure consistency, we use$(1 - V_{matrix})$ as a determining parameter. After obtaining the feature similarity matrix F_{matrix}, a group is formed based on the lower value of the two-agent pair. Then, the group index set can be constructed based on the relationships between the members, where k is the k-th group, and is the union of each pair (i,j). There are no overlapping members between each group. The indices of all members in group G are defined as:

$$\begin{aligned} G = \{G_k \mid G_k = \bigcup_{(i,j)\in P} \{i,j\}, \\ G_a \cap G_b = \emptyset \text{ for } a \neq b\} \end{aligned} \tag{6}$$

In extreme cases, when all agents have similar speeds and distances, they are usually grouped together. In fact, in such an extreme grouping situation the intra-group interaction and direct interaction will still work, and the rest of the model will function normally and ultimately be able to effectively make predictions.

Group Pooling and Group Unpooling. To achieve an efficient and accurate agent trajectory prediction model, we use GNNs for modeling. In our model, each graph node must retain its identity index information to ensure that no unnecessary nodes are introduced or redundancies are removed. Zhu et al. [37] discuss the impact of aggressive driving behavior on traffic flow and surrounding vehicles, pointing out that vehicles with high speeds and frequent lane changes affect the vehicles around them, causing them to increase speeds as well. According to our observations, there must be certain factors within a group that generate different weights, enabling more reasonable aggregation. Just like in a team, there must be one or more leaders who contribute significantly to the team, and

through the behavior of these leaders, the behavior of the entire group can be better reflected [38]. Therefore, we infer that the weights of these determining factors should be emphasized during group aggregation. As shown in Fig.4, the stronger the determining factors of different agents, the greater their weights, and the more they influence the group's actions.

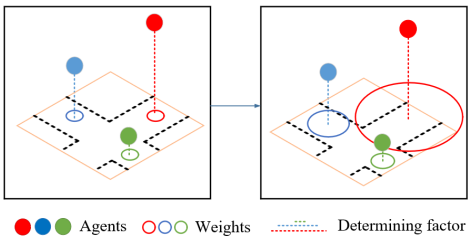

Figure 4: Schematic representation of pooling types (left: average of pooling, right: attention pooling according to determinants). Different colors represent different agents, the weight is indicated by the radius of the circle below the agents, and the determinant factor is indicated by the dotted line below the agents.

To achieve this, we propose group pooling and unpooling methods for agent graphs. Based on the group behavior properties of collecting surrounding information and sharing behavior patterns among group members, we first group the agent nodes and aggregate the features of the corresponding nodes into one node based on the attention weights. Then, the group features processed by the target encoding module are stacked for the subsequent group fusion module. Through group pooling, the most representative features of each agent node are selected, with the speed of the group members determining their influence. If a member has a higher speed, its influence on the group is greater, and the assigned weight is correspondingly larger. With this feature, we can model the re-grouped graph structure, which has fewer nodes than before. The clustered trajectory feature Z is defined as:

$$Z = \{Z_k \mid k \in [1, 2, \ldots, K]\}, \quad Z_k = W_k \sum_{i \in G_k} X_i \tag{7}$$

Where K is the total number of groups, and W is the attention weight. Next, unpooling operations are needed to reconstruct the grouped graph structure back to its original size. This allows each agent trajectory to be predicted using the deep feature information of the fused output. Considering that the convolution process on zero vector nodes cannot display group attributes [39], we choose to copy the group features and assign them to all relevant group member nodes to ensure they have the same group behavior information. The unpooling of vehicle groups can be expressed as:

$$\begin{aligned} \overline{X} &= \left\{\overline{X}_n \mid n \in [1, 2, \ldots, N]\right\}, \\ \overline{X}_n &= Z_k \quad where \quad n \in G_k \end{aligned} \tag{8}$$

Our initial motivation for replicating group features and assigning them to all group members is to capture the collective characteristics and patterns of the group as a whole. This is done to simplify the representation of the group and to improve the model's ability to understand and process group-related information.

Target Encoding Module. To maintain the simplicity of GSTEP, we use a 1D CNN-based network [25] as our target encoding module to process historical trajectories. The target encoding module encodes the intra-group/inter-group/direct interaction features and outputs them to the group fusion module. The target encoding module is a combination of 1D CNN and Feature Pyramid Network (FPN). The 1D CNN processes the trajectories, extracts multi-scale features and improves the efficiency of parallel computation, and then the feature pyramid network fuses the multi-scale features for output.

Group Fusion Module. We incorporate interactions between agents as a form of group-level interaction into the constructed target encoding module. By providing three different types of graph structure data (intra-group/inter-group/direct) to the same encoding module, rich features are extracted. The vehicle graph is defined as $G_{veh} = (V_{veh}, E_{veh})$, consisting of a set of agent nodes $V_{veh} = \{X_n \mid n \in [1, 2, \ldots, N]\}$ and the corresponding edges $E_{veh} = \{e_{i,j} \mid i, j \in [1, 2, \ldots, N]$. The intra-group interaction graph is defined as $G_{member} = (V_{veh}, E_{member})$, consisting of a set of agent nodes V_{veh} and the pairwise interaction edges E_{member} of the group members. $E_{member} = \{e_{i,j} \mid i, j \in [1, 2, \ldots, N], \{i, j\} \subset G_k, k \in [1, 2, \ldots, N]\}$. Through this graph representation, agent nodes can learn the norms of internal collision avoidance and following among group members while maintaining their own formation and direction. Interactions between groups are equally important for learning norms between groups. We define the inter-group interaction graph as $G_{group} = (V_{group}, E_{group})$. The nodes represent the features of each group as $V_{group} = \{\overline{X}_k | k \in [1, 2, \ldots, K]\}$, and the edges represent the interactions between groups as $E_{group} = \{\overline{e}_{p,q} \mid p, q \in [1, 2, \ldots, K]\}$.

Weights are shared to reduce the number of parameters. Subsequently, the output features are aggregated, and the group fusion features generated by the group fusion module F_ψ are denoted as $\hat{Y}$, represented as:

$$\begin{aligned} \hat{Y} = F_\psi(& \underbrace{F_\theta(X, G_{veh})}_{\text{Direct interaction}}, \\ & \underbrace{F_\theta(X, G_{member})}_{\text{Intra-group interaction}}, \\ & \underbrace{F_\theta(X, G_{group})}_{\text{Inter-group interaction}}) \end{aligned} \tag{9}$$

Where F_ψ and F_θ are learnable parameters, randomly initialized at the beginning of the experiment.

3.3.3 Map Encoder

To maintain the simplicity of GSTEP, we use a PointNet-based encoder [24] as our map encoding module to extract static map features.

Map encoder encodes map data by representing road elements as polylines, where each polyline consists of a sequence of vectorized segments. These segments are modeled as local subgraphs, capturing intra-polyline spatial relationships. A global graph is then constructed by connecting local subgraphs, allowing for the modeling of inter-polyline interactions, such as intersections and lane connections. The encoding process leverages a hierarchical graph neural network that first aggregates node features within each local subgraph and subsequently aggregates across the global graph, enabling efficient extraction of both local and global map features.

In general, we let all potential features have D channels. Therefore, the generated agent and map tokens have shapes $[N_a,\ D]$ and $[N_m,\ D]$. Additionally, relative poses are further encoded by an MLP to obtain RPE with shape $[N,\ N,\ D]$.

3.4. Internal and External Synergistic Perception Fusion Module

Once the instance tokens and corresponding RPE are obtained, we use the proposed IESP module to collaboratively update the instance tokens both internally and externally. Fig.5 shows the overall structure of the proposed IESP module, which consists of multiple stacked IESP layers, similar to a standard Transformer [32]. Essentially, the driving scene can be considered a complete digital graph with self-loops, where the features centered on the input instances are the nodes, and the RPE describes the related edge information. During the update process, node features are only influenced by edges related to the target node. At the microscopic level, the tokens of the i-th and j-th instances are denoted as f_i and f_j, respectively. The RPE vector associated with the edge from f_i to f_j is designated as $r^{'}_{i\to j}$ to contain all the information to be propagated from node i to node j. Therefore, a simple MLP can be used to encode these features $(f_i, f_j, r'_{i\to j})$ and obtain the i-th contextual feature vector of node j :

$$c_{i\to j} = \phi\big(f_i \oplus f_j \oplus r^{'}_{i\to j}\big) \tag{10}$$

Where $\oplus$ denotes the concatenation operator, and $\phi : \mathbb{R}^{3D} \to \mathbb{R}^{D}$ represents the MLP, consisting of linear layers, layer normalization, and ReLU activation. Then cross-attention is performed on the target node and its context:

$$f'_j = \mathrm{MHA}(\text{Query: } f_j, \text{Key: } C_j, \text{Value: } C_j) \tag{11}$$

$\mathrm{MHA}(\cdot,\cdot,\cdot)$ is the standard multi-head attention function, and $C_j = \{c_{i\to j}\}_{i\in\{1,\dots,N\}}$ is the set of contextual feature vectors for node j. Note that C_j also includes $c_{i\to j}$, indicating that each node has a self-loop. Similar to the standard Transformer, a point-wise feed-forward layer is integrated after the attention mechanism. Additionally, in each layer, we update the RPE in two ways simultaneously: first, through self-updating with extrinsic attention $r^{'}_{i\to j}$, and second, by re-encoding the contextual feature vector with another MLP to update $r^{'}_{i\to j}$.

It should be noted that RPE corresponds to the instance tokens. As the instance tokens undergo a series of updates, the RPE needs to be updated both intrinsically and extrinsically to match the latest tokens. This allows RPE to not only focus on the state and behavior of the vehicle itself but also on changes in the surrounding environment, such as other vehicles and traffic maps. Therefore, we further enhance RPE through external attention to feature learning. Specifically, the RPE implicitly learns the connections between samples through two memory units (M_k and M_v). External attention[40] is characterized by simplicity and low complexity compared to the standard self-attention, and is suitable for calculating attention at different locations on the same sample. This characteristic makes it suitable for use in efficient models. The structure of external attention is shown in Fig.6.

First, given the input instance tokens as $F \in \mathbb{R}^{N\times D}$, we unfold and replicate them N times along different dimensions to establish source arrays and target arrays, both shaped $[N,N,D]$. By concatenating the source array, target array, and corresponding RPE, we obtain a tuple array and then apply ϕ to get the contextual feature array $C \in \mathbb{R}^{N\times N\times D}$. Note that the j-th row of C is exactly C_j , representing the set of contextual features centered on token j. Thus, C is used as the key and value, while the expanded $F \in \mathbb{R}^{N\times 1\times D}$ is used as the query. Then, the standard multi-head attention module transfers information from the contextual features to the instance tokens. Other components of the IESP layer are also vectorized, but we will not delve into the details due to their simplicity. It is noteworthy that our proposed IESP layer is similar to recent "query-centric" methods [7, 41], but we have incorporated attention and RPE updates, making the design more comprehensive and detailed.

3.5. Motion Decoder

After the internal and external synergetic perception modules, the model collects instance tokens to feed into the multimodal motion decoder, which generates predictions for all agents. We predict a total of k future trajectories, and for each mode a simple MLP is applied, which has a trajectory regression header and a classification header, which is followed by a softmax function to compute the

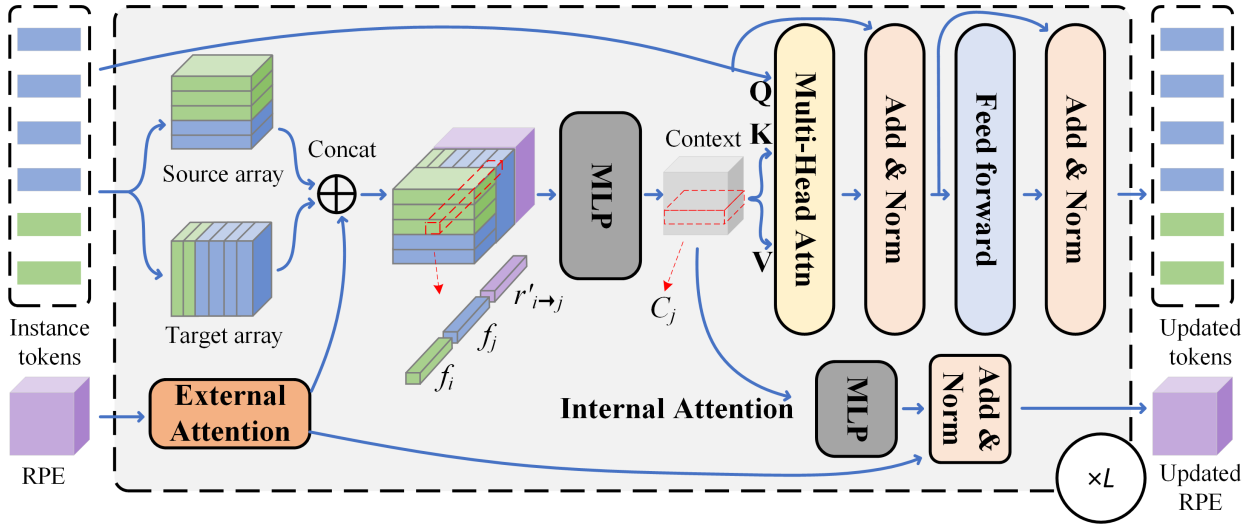

Figure 5: Illustration of the proposed IESP module with L layers. RPE and tokens are updated at each IESP layer.

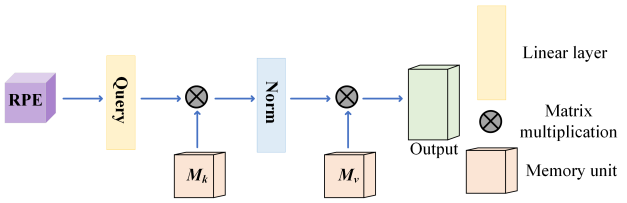

Figure 6: Schematic diagram of the structure of external attention. RPE will learn implicit features through two memory units.

corresponding probability scores.

It is worth noting that in the regression header, we chose to use a continuous parameterized representation, introducing Bernstein basis polynomials (Bezier curves) for better convergence.

In practice, a simple MLP is used to perform the mapping of the regression head and we can calculate the coordinates of the predicted trajectory positions.

4. Experiments

4.0.1 Quantitative Analysis

4.1. Experimental Setup

Datasets. We evaluated the proposed method on the Argoverse 1 and Argoverse 2 motion forecasting datasets. The Argoverse 1 contains 205942, 39472, and 78143 sequences for training, validation, and testing, respectively. Each sequence is sampled at $10Hz$, and the task is to predict future 3-second motion trajectories based on 2 seconds of historical observations (i.e., H=20, T=30). The Argoverse 2 contains 200000, 25000, and 25000 sequences for training, validation, and testing. The sequences are also sampled at $10Hz$, and the task is to predict future 6-second motion trajectories based on 5 seconds of historical observations (i.e., H=50, T=60). Both datasets provide high-definition maps.

Evaluation Metrics. We use standard metrics commonly used in multimodal trajectory prediction, including minimum average displacement error (minADE_k), minimum final displacement error (minFDE_k), miss rate (MR_k), and brier-minFDE_k. All these metrics evaluate the best predicted trajectory of a single target agent among k hypotheses against the ground truth. The minADE_k is the average Euclidean distance between the predicted trajectory and the ground truth, while minFDE_k only considers the error at the predicted endpoint. MR_k is the percentage of sequences with a prediction error greater than 2 meters under the minFDE_k metric. Brier-minFDE_k adds an additional Brier score $(1-p)^2$ to minFDE_k, where p represents the probability of the best predicted trajectory. For detailed definitions, refer to [17].

Implementation Details. The overall loss function is consistent with the baseline and consists of a regression loss function as well as a classification loss function, respectively. The regression task handles multimodal predictions according to the winner-take-all rule. For each agent, the best prediction among the k predictions is based on the one with minimum final displacement error. The maximum marginal loss is used in the classification task to make a distinction. We set all potential vector dimensions to $D = 128$ and the number of fusion layers to $L = 4$. For the multimodal decoder, we set the number of modes to $k = 6$, following the common setups. GSTEP was trained end-to-end on a single Nvidia RTX 3090 GPU server with

Table 1: Comparison of experimental results for different models on the test split on the Argoverse 1 motion forecasting dataset. (The best result is in **bold** while the second best result is underlined .)

Method	minADE_k	minFDE_k	MR_k	b-minFDE_k	Param
AutoBot[42]	0.89	1.41	-	-	1.5M
LaneGCN[25]	0.87	1.36	0.162	2.05	3.7M
mmTrans[43]	0.84	1.34	0.154	2.03	2.6M
D-TNT[44]	0.88	1.28	0.126	1.98	**1.1M**
THOMAS[45]	0.94	1.44	0.100	1.97	-
TPCN[46]	0.82	1.24	0.133	1.93	-
SceneTrans[47]	0.80	1.23	1.126	1.89	15.3M
HiVT[2]	0.77	1.17	0.127	1.84	2.5M
GANet[48]	0.81	1.16	0.118	1.79	-
HPNet[49]	0.76	1.10	0.107	1.74	-
LTP[5]	0.83	1.30	0.155	1.86	**1.1M**
ADAPT[49]	0.80	1.18	-	1.82	1.4M
SIMPL[41]	0.79	1.18	0.123	1.81	1.8M
GSTEP(ours)	**0.65**	**0.97**	**0.084**	**1.60**	1.9M

Table 2: Comparison of experimental results for different models on the test split on the Argoverse 2 motion forecasting dataset.

Method	minADE_k	minFDE_k	MR_k	b-minFDE_k	Param
HDGT[50]	0.84	1.60	0.214	2.24	12.1M
GoReal[36]	0.76	1.48	0.220	2.01	-
QCNet[7]	**0.65**	**1.29**	**0.160**	**1.91**	7.3M
SIMPL[41]	0.72	1.43	0.192	2.05	**1.9M**
GSTEP(ours)	0.67	1.35	0.180	1.95	2.0M

a batch size of 8 for 30 epochs. The Adam optimizer was used with an initial learning rate of 1e-3, which gradually decreased to 1e-4 after 20 epochs.

4.2. Experimental Analysis

As shown in Table 1, GSTEP is compared with state-of-the-art models on the Argoverse 1 motion dataset, and GSTEP achieves highly competitive results among all listed methods. We select models such as LaneGCN which is a graph convolutional networks (GCNs) variant, and GANet which is a generative adversarial networks (GANs) variant. These models are chosen as they represent common and well-established approaches in the field of deep learning for trajectory prediction. Notably, SceneTransformer [47] has a larger model size but performs worse, indicating a higher demand for data and a lack of generality. HiVT [2] explicitly considers relative pose during feature fusion, while GSTEP has a simpler design and better performance. The LTP [5] model is smaller and has fewer parameters, making it difficult to handle slightly complex interactions, resulting in relatively poorer performance. ADAPT [49] uses dynamic weight learning similar to our group encoding concept, but its experimental results are slightly inferior to GSTEP. Although GSTEP does not have the fewest parameters, it achieves the best performance within the limited parameters. The experimental results on the Argoverse 2 motion dataset are shown in Table 2. Compared to the baseline SIMPL [41], GSTEP only adds 0.1M parameters but achieves significant improvements in both ADE and FDE evaluation metrics. The evaluation results of inference latency are depicted in Fig.7. All experiments are conducted using the original PyTorch implementation on the same GPU. In Fig.7, with the benefit of the small number of parameters and concise model design, the inference latency of GSTEP is superior to HiVT[2] and LaneGCN[25]. With the same acceleration, GSTEP achieves high-speed inference capability, which is expected to be followed by further real-world applications. Overall, GSTEP achieves the desired results with fewer parameters by effectively extracting deep common features of agent groups and employing an attention fusion mechanism, successfully generating multiple agent trajectories that follow specific scene constraints in complex scenarios. GSTEP is designed in a modular and hierarchical manner. A key aspect that promotes its potential scalability is its relatively small number of parameters. A smaller number of parameters means that the model requires less computational resources and memory during training and inference.

Table 3: Ablation studies for each module. Both group encoder (GE) and the IESP module enhance model performance.

GE	IESP	minADE_k	minFDE_k	MR_k	b-minFDE_k	Param
		0.793	1.179	0.123	1.809	**1.8M**
√		0.688	1.017	0.095	1.624	1.9M
	√	0.695	1.055	0.102	1.688	**1.8M**
√	√	**0.654**	**0.973**	**0.084**	**1.602**	1.9M

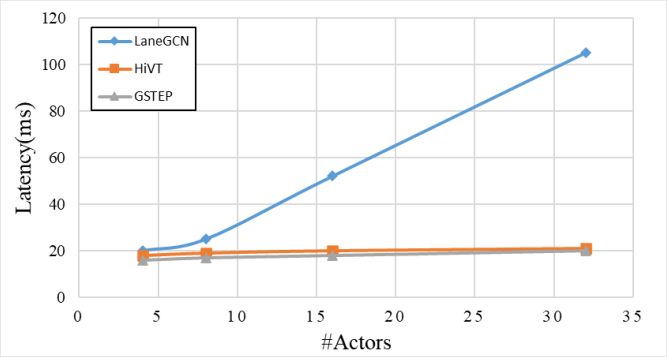

Figure 7: Evaluation results of the average inference latency of different methods on the Argoverse 1 dataset in relation to the number of target agents.

Table 4: Ablation studies of the IESP module design on the Argoverse 1 test split. D and L represent the potential vector dimension and the number of IESP layers, respectively.

Model	D	L	minADE_k	minFDE_k	MR_k
M1	64	2	0.70	1.08	0.105
M2	128	4	0.65	0.97	0.084
M3	128	6	0.64	0.96	**0.080**
M4	256	4	**0.62**	**0.95**	0.085

Table 5: Ablation studies with group assignment modules.

Method	minADE_k	minFDE_k	MR_k	b-minFDE_k
-	0.793	1.179	0.123	1.809
$0.5d$+$0.5c$	0.721	1.082	0.104	1.681
d	0.695	1.035	0.097	1.641
c	0.714	1.090	0.122	1.670
d*c	**0.688**	**1.017**	**0.095**	**1.624**

Table 6: Ablation studies for group pooling.

Method	minADE_k	minFDE_k	MR_k	b-minFDE_k
Attention	**0.688**	**1.006**	**0.092**	**1.616**
Average	**0.688**	1.017	0.095	1.624

4.2.1 Qualitative Analysis

The qualitative results on the Argoverse 1 dataset are shown in Fig.8. GSTEP can simultaneously predict realistic, reasonable, and accurate multimodal future trajectories for multiple agents in the complex scene. This excellent performance is due to GSTEP's ability to delineate appropriate groups within the traffic flow. As shown in the second row of Fig.8, by grouping similar agents, our GSTEP can capture the relationships and interactions between group members. By encoding group features and leveraging both internal and external synergistic perception, the proposed GSTEP method improves prediction accuracy and ensures a clear estimation of the overall direction of group members' actions. This manifestation of the model's decision-making ability and effectiveness demonstrates the advantages of our GSTEP in vehicle trajectory prediction tasks. Through reasonable decision-making and accurate predictions, our GSTEP can better simulate and understand vehicle behavior, providing useful information and guidance for practical applications.

4.2.2 Ablation Study

To test the effectiveness of the proposed approach, we conducted ablation studies on each of the proposed innovations, as shown in Table 3. It is noteworthy that both group encoder (GE) and the IESP module enhance model performance. The overall model performance improved after adding group encoding, which forms intra-group and inter-group interactions, demonstrating that group encoding can indeed capture deep commonalities among agents. We found that the experimental results further improved to the best level when the IESP module was added, indicating that the IESP module can better fuse and update data through dual attention mechanisms.

As shown in the Table 4, we found that GSTEP achieves better performance in all metrics (M1→M4) as the potential vector dimension and the number of IESP layers increased. However, at the cost of a large number of parameter increases, the prediction accuracy was only slightly improved, which was not desirable in real-time applications (M2→M3,M2→M4). We therefore chose M2 as the parameter configuration for our experiments.

Additionally, to further verify the effectiveness of group encoding, ablation studies were also conducted on the

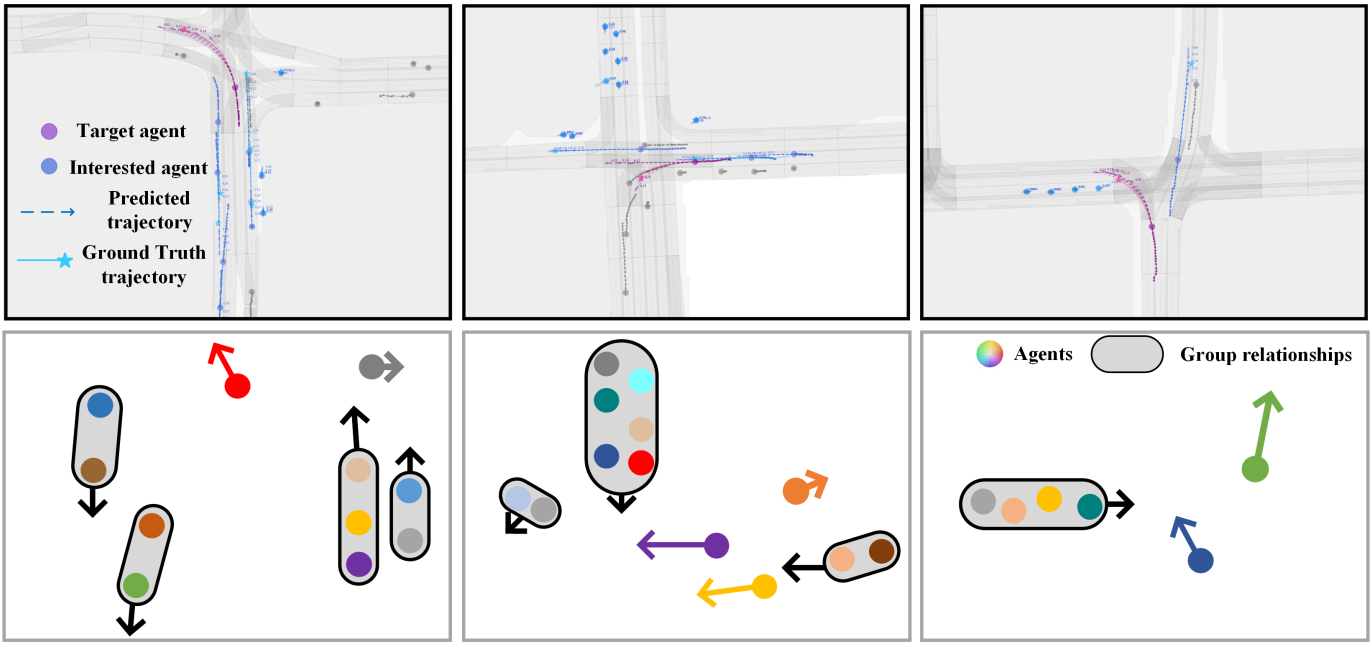

Figure 8: Visualization of qualitative results and corresponding group relationships for the Argoverse 1. (First row) The target agents are shown in red, while other agents of interest are shown in blue. Ground truth endpoints are denoted by asterisks and predicted trajectories are denoted by dashed lines. For conciseness, we omit the motion results for the ignored agent (gray). (Second row) Different agents are represented by different colors, arrows refer to the direction of movement, and group relationships are circled in gray.

grouping module and group pooling. As shown in Table 5, experiments were conducted on the impact of different relationships between the distance matrix D_{matrix} and the velocity similarity matrix V_{matrix} in the feature similarity matrix F_{matrix} on model performance, where d and c represent D_{matrix} and (1-V_{matrix}), respectively. The experiments revealed that the combined case of d*c achieved the best performance. Comparing this with experiments using only d or c, it can be concluded that using a single determining parameter alone cannot achieve appropriate grouping results and thus cannot reach the desired experimental outcomes. These experiments demonstrate that distance is an important factor for group assignment, but the direction of velocity is also indispensable.

In group pooling, determining factors were chosen to reflect the weight of agents in group pooling. To verify the effectiveness of these determining factors, a comparison was made with average pooling, which has no weight differences. As shown in Table 6, it was found that using attention for pooling improves group performance compared to average pooling. Overall, the experiments demonstrate the effectiveness of determining factors.

5. Conclusion

In this paper, we present a trajectory prediction model based on a grouped spatial-temporal encoder. To explore the deep commonalities among traffic participants, the GSTEP model groups and pools the features of all traffic participants, capturing and integrating group commonalities to obtain deep features. To maintain synchronized fusion and updating of RPE with relevant features, we propose a compact and efficient internal-external collaborative perception fusion module, achieving comprehensive and efficient global feature fusion, thus improving model accuracy. Experimental results on the large-scale public dataset Argoverse 1&2 show that the GSTEP achieves competitive results in terms of model size and accuracy. In the future, we will further investigate the prediction challenges of interactions between pedestrian groups and vehicle groups.

References

[1] Feng, Chen and Zhou, Hangning and Lin, Huadong and Zhang, Zhigang and Xu, Ziyao and Zhang, Chi and Zhou, Boyu and Shen, Shaojie . Macformer: Map-agent coupled transformer for real-time and robust trajectory prediction. IEEE Robotics and Automation Letters, 2023

[2] Zhou Z, Ye L, Wang J, Wu K, Lu K. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 8823–8833

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

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

[5] Wang J, Ye T, Gu Z, Chen J. Ltp: Lane-based trajectory prediction for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 17134–17142

[6] Wang X, Su T, Da F, Yang X. Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 21995–22003

[7] Zhou Z, Wang J, Li Y H, Huang Y K. Query-centric trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 17863–17873

[8] Zhang L, Su P H, Hoang J, Haynes G C, Marchetti-Bowick M. Map-adaptive goal-based trajectory prediction. In: Conference on Robot Learning. 2021, 1371–1383

[9] Zhao H, Gao J, Lan T, Sun C, Sapp B, Varadarajan B, Shen Y, Shen Y, Chai Y, Schmid C, others . Tnt: Target-driven trajectory prediction. In: Conference on Robot Learning. 2021, 895–904

[10] 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. 2021, 7577–7586

[11] Sieben A, Schumann J, Seyfried A. Collective phenomena in crowds—where pedestrian dynamics need social psychology. PLoS one, 2017, 12(6): e0177328

[12] Li G Q. Research on the model of road crossing based on pedestrian psychology. In: 2021 33rd Chinese Control and Decision Conference. 2021, 6864–6868

[13] Moussaïd M, Perozo N, Garnier S, Helbing D, Theraulaz G. The walking behaviour of pedestrian social groups and its impact on crowd dynamics. PloS one, 2010, 5(4): e10047

[14] Zhou B, Wang X, Tang X. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2012, 2871–2878

[15] Bisagno N, Zhang B, Conci N. Group lstm: Group trajectory prediction in crowded scenarios. In: Proceedings of the European conference on computer vision (ECCV) workshops. 2018

[16] Zhou B, Tang X, Wang X. Coherent filtering: Detecting coherent motions from crowd clutters. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12. 2012, 857–871

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

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

[19] Dai S, Li L, Li Z. Modeling vehicle interactions via modified lstm models for trajectory prediction. Ieee Access, 2019, 7: 38287–38296

[20] Nikhil N, Tran Morris B. Convolutional neural network for trajectory prediction. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018, 0–0

[21] Li X, Ying X, Chuah M C. Grip: Graph-based interaction-aware trajectory prediction. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). 2019, 3960–3966

[22] Li X, Ying X, Chuah M C. Grip++: Enhanced graph-based interaction-aware trajectory prediction for autonomous driving. arXiv preprint arXiv:1907.07792, 2019

[23] Ziegler J, Bender P, Schreiber M, Lategahn H, Strauss T, Stiller C, Dang T, Franke U, Appenrodt N, Keller C G, others . Making bertha drive—an autonomous journey on a historic route. IEEE Intelligent transportation systems magazine, 2014, 6: 8–20

[24] Gao J, Sun C, Zhao H, Shen Y, Anguelov D, Li C, Schmid C. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, 11525–11533

[25] Liang M, Yang B, Hu R, Chen Y, Liao R, Feng S, Urtasun R. 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. 2020, 541–556

[26] 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. 2022, 6498–6507

[27] Bae I, Park J H, Jeon H G. Learning pedestrian group representations for multi-modal trajectory prediction. In: European Conference on Computer Vision. 2022, 270–289

[28] Xu D, Shang X, Liu Y, Peng H, Li H. Group vehicle trajectory prediction with global spatio-temporal graph. IEEE Transactions on Intelligent Vehicles, 2022, 8(2): 1219–1229

[29] Zhao Z, Fang H, Jin Z, Qiu Q. Gisnet: Graph-based information sharing network for vehicle trajectory prediction. In: 2020 International Joint Conference on Neural Networks (IJCNN). 2020, 1–7

[30] Alahi A, Goel K, Ramanathan V, Robicquet A, Fei-Fei L, Savarese S. Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, 961–971

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

[32] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I. Attention is all you need. Advances in neural information processing systems, 2017, 30(1): 261–272

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

[34] Yuan Y, Weng X, Ou Y, Kitani K M. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021, 9813–9823

[35] Seff A, Cera B, Chen D, Ng M, Zhou A, Nayakanti N, Refaat K S, Al-Rfou R, Sapp B. Motionlm: Multi-agent motion forecasting as language modeling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023, 8579–8590

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

[37] Zhu H, Chen G, Lin H, Zhou Y. The impact of aggressive driving behaviors on multi-lane highway traffic flow. International Journal of Modern Physics C, 2018, 29: 1850056

[38] Zhang P, Cheng H, Huang D, Yang L, Lo S, Ju X. Experimental study on crowd following behavior under the effect of a leader. Journal of Statistical Mechanics: Theory and Experiment, 2021, 2021: 103402

[39] Gao H, Ji S. Graph u-nets. In: international conference on machine learning. 2019, 2083–2092

[40] Guo M H, Liu Z N, Mu T J, Hu S M. Beyond self-attention: External attention using two linear layers for visual tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(5): 5436–5447

[41] Zhang L, Li P, Liu S, Shen S. Simpl: A simple and efficient multi-agent motion prediction baseline for autonomous driving. IEEE Robotics and Automation Letters, 2024, 9(4): 3767–3774

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

[43] Huang Z, Mo X, Lv C. Multi-modal motion prediction with transformer-based neural network for autonomous driving. In: 2022 International Conference on Robotics and Automation (ICRA). 2022, 2605–2611

[44] 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. 2021, 15303–15312

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

[46] Ye M, Cao T, Chen Q. Tpcn: Temporal point cloud networks for motion forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, 11318–11327

[47] Ngiam J, Caine B, Vasudevan V, Zhang Z, Chiang H T L, Ling J, Roelofs R, Bewley A, Liu C, Venugopal A, others . Scene transformer: A unified multi-task model for behavior prediction and planning. arXiv preprint arXiv:2106.08417, 2021, 2

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

[49] Tang X, Kan M, Shan S, Ji Z, Bai J, Chen X. Hpnet: Dynamic trajectory forecasting with historical prediction attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 15261–15270

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