2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

HHGNN: Heterogeneous Hypergraph Neural Network for Traffic Agents Trajectory Prediction in Grouping Scenarios

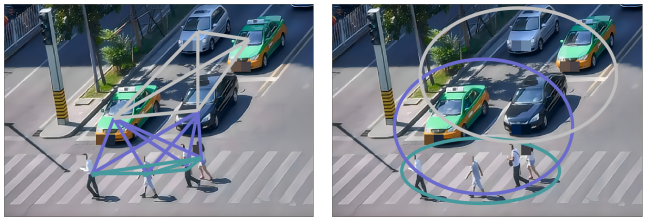
Hetian Guo[1], Yingzhi Peng[1], Zipei Fan[2,*], He Zhu[1], Xuan Song[1]

***Abstract*— In many intelligent transportation systems, predicting the future motion of heterogeneous traffic participants is a fundamental but challenging task due to various factors encompassing the agents' dynamic states, interactions with neighboring agents and surrounding traffic infrastructures, and their stochastic and multi-modal natural behavior tendencies. However, existing approaches have limitations as they either focus solely on static, pairwise interactions, ignoring interactions of varied granularity, or fail to tackle agents' heterogeneity. In this paper, instead of focusing solely on pairwise interactions, we propose a Heterogenous Hypergraph Graph Neural Network (HHGNN) based motion prediction model that leverages the nature of hypergraph to encode the groupwise interactions among traffic participants. Moreover, we propose the type-aware two-level hypergraph message passing module (TTHMS) with learnable hyperedge-type embeddings to model the intra-group and inter-group level interactions among heterogeneous traffic agents (e.g., vehicles, pedestrians, and cyclists). Besides, We integrate a scene context fusion layer in TTHMS to incorporate the scene context. Comparison and ablation experiments on the Waymo Open Motion Dataset (WOMD) demonstrate HHGNN's effectiveness within the motion prediction task.**

I. INTRODUCTION

Recently, a surging cohort of researchers has progressively focused on autonomous driving tasks, encompassing an expansive array of application scenarios within the industrial domain. Motion prediction is a core component in attaining safe and efficient autonomous driving systems. The public datasets, public benchmarks [1], [2], [3], [4], [5], and insightful works from researchers enable increasingly sophisticated solutions to the task. Autonomous driving vehicles (AVs) operate within complex scenarios encompassing scene context and intricate interactions between the AVs and other traffic participants. These complexities collectively present formidable challenges in the realm of motion prediction.

In real-world scenarios, drivers make traffic decisions based on various factors, including current vehicle state (speed, position, direction, etc.), surrounding agents' state and moving tendency, and the scene context (lanes, crossings, traffic lights, etc.). For example, consider a scenario where there is a three-lane highway with traffic flowing in the same direction. A vehicle positioned on the far left lane intends to merge into the middle lane. However, if, at that moment, a vehicle traveling side by side on the far right lane also attempts to merge into the middle lane, then the left lane vehicle would abort the lane change to avoid collision.



(a) Pairwise (b) Groupwise

Fig. 1. Fig 1(a) shows the pairwise connections (pedestrian-pedestrian, pedestrian-vehicles, vehicles-vehicles) by normal edges. Fig 1(b) shows the groupwise connections by hyperedges. In such a traffic scenario shown above, it is intuitive to leverage groupwise connections to represent the interactions among agents, and it diminishes the redundant connections. Overly dense connections result in extra computational costs and the potential freezing-robot problem.

Plentiful methods have been proposed to model the interaction between traffic participants. [6], [7], [8] encode the features of the road map and the agents into graph nodes. [9], [10] treat the traffic scenario as a spatial-temporal graph.

A large portion of current approaches restrict interaction to pairwise interactions, i.e., interactions between pairs of agents. These approaches ignore the fact that agents can naturally lead to groupings in various ways. These groupings can be temporary and spontaneous due to traffic congestion, traffic signal changes, merging lanes, or the natural flow of vehicles on the road. Fig 1(a) illustrates simply applying pairwise interaction is unsuitable for grouping scenarios.

Besides, the driving scenarios often involve heterogeneous traffic participants (e.g., vehicles, pedestrians, and cyclists), and their interactions are diverse. We imply that the interactions have different types, which must be explicitly preserved for comprehensive interaction representations.

To tackle the aforementioned problems, we present HHGNN. We use type-specific encoders to explicitly encode agents of varied categories to preserve heterogeneity, and a hyperedge construction module extracts interaction groups upon the agents. We show how hyperedges aid in modeling interaction in Fig 1. Based on the formulated hyperedges and encoded agent features, a novel hypergraph message passing module is leveraged, modeling the interaction of varied granularity and combining the scene context with an integrated scene context fusion layer.

The main contributions of this work can be summarised:

1) We propose the TTHMS module to model the inter-group and intra-group interactions.

*Corresponding author
[1]The authors are with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, P.R. China (e-mail: hetianguoml@gmail.com; yingzhi.peng@outlook.com; zhuye140@gmail.com; songx@sustech.edu.cn).
[2]The author is with the School of Artificial Intelligence, Jilin University, Changchun 130012, P.R. China (e-mail: fanzipei@jlu.edu.cn)

2) We include learnable hyperedge-type embeddings into attention calculation for handling heterogeneity of interactions (e.g., vehicle-vehicle-pedestrian, pedestrian-pedestrian-vehicle).
3) We integrate a scene context fusion layer within the message-passing module above to incorporate the scene context information as a complementary information source, resulting in more scene-compliant predicted trajectories.

II. Related Works

A. Motion Prediction for Autonomous Driving.

Autonomous driving systems are now booming with applications in many fields. Autonomous driving encompasses three interdependent components: perception, motion prediction, and motion planning. Autonomous vehicles use sensors to obtain information about the state of the surrounding traffic participants and the scene context. Based on the scene context, the driving states of ego agents, and other agents, the future states of the surrounding vehicles can be predicted. The predicted states, the state of the surrounding traffic participants, and the environmental information are upstream features for path planning. A well-performing trajectory prediction model is crucial for the motion prediction module, being an intermediate component, to obtain a safe and efficient autonomous driving system.

In addition to the application on pedestrians and vehicles, recent works have also extended the task to vessel motion prediction, which fuels the motion planning model of vessels [11], and TrajAir [12] constructed a motion prediction dataset for air-crafts and proposed a baseline of air-craft motion prediction. Motion prediction demonstrates its importance in different application scenarios.

Existing trajectory prediction models mainly contain three critical components: the interaction modeling module, which models interactions between different traffic participants (e.g., cars, pedestrians, and bicyclists), and it is further discussed in the section II-C; and the scene context information extraction&fusion module which encodes scene context features. There are many ways to incorporate scene context information, further discussed in section II-B. The prediction module outputs the future trajectories given the interaction representations and encoded scene context features. Existing works explore various prediction strategies to derive multi-modal future motion. Early works [13], [14], [15], [16] directly regress a set of trajectories. Other works like [17], [18], [19] utilize Gaussian Mixture Models (GMMs) to parameterize multi-modal predictions of the agent's future positions, generating compact distribution. HOME [20] and GOHOME [21] first predict a heatmap and obtain trajectories by sampling. Goal-based methods [22], [6], [23], [8], [24] first predict several potential goal points and then regress the complete trajectory for each goal.

B. Scene Context Encoding.

Incorporating scene context is essential for motion prediction. For instance, the relation of two adjacent lanes traveling in opposite directions close to each other at an intersection is critical for predicting U-turns.

Two common approaches are utilized for scene context encoding. First, the rasterization approach encodes the road map data points as an image from a "birds-eye" view and uses Convolutional Neural Networks (CNNs) to encode the image as in Social Lstm [15], Trajereon++ [25], etc. Trajectron++ [25] utilizes a grid-based map encoding module aiming to capture the environment's spatial layout and static features. It takes as input a grid map representation of the scene, where each cell represents a particular region of the environment. However, the rasterization approach has significant drawbacks: the constrained field of view due to CNN's limited receptive field and inefficiency in representing continuous physical states [18]. Moreover, map rasterization may lose helpful information, such as road graph topology characteristics. Besides, the vectorization approach, proposed by VectorNet [7], extracts all the geographic entities (e.g., lanes, traffic lights) as polylines and better captures the HD map's structural features. Recent works with the vectorization approach treat the polylines as nodes on graph [16], [26] or process the highly related polylines in the per-agent coordinate system [18], [7], [6]. This work leverages the vectorization method and extracts a dynamic local map representation for each traffic participant to incorporate road map information.

C. Interaction Modelling.

Individual agents adjust their paths by implicitly reasoning about neighboring agents. These neighbors, in turn, are influenced by others in their immediate surroundings and could alter behavior over time. Recent studies have proposed various methods to model the interactions between agents.

In particular, Social LSTM [15] and Social GAN [14] aggregate features from neighboring agents. Social pooling schema shares the latent pedestrian representation with neighbors by constructing a shared hidden state tensor. VectorNet [7] and Dense-TNT [6] use a hierarchical heterogeneous graph to represent the interaction, where a sub-graph and a fully connected graph represent each object and then describe all the objects. Interaction is modeled by message passing on the global graph. Edge attribute is ignored in both. [27] places moving agents in exclusive per-agent coordinates as nodes in the Graph Neural Network (GNN) and leverages edge-enhanced masked attention to cooperate with the edge attributes. [9], [10], [28], [29] derive spatial-temporal graph from trajectory data to capture the correlation in both time and space domains. [29], [10] utilize spatial-temporal graph convolution to aggregate information. [30] proposes a two-phase aggregation network that subsequently aggregates messages from a specific category of the target agent and all other categories.

Hypertron [31] leverages a spatial-temporal hypergraph neural network to capture group-wise interaction. DynGroupnet [32] learns a dynamic multi-scale hypergraph whose node represents the agent and hyperedge represents the interacting

group and leverages the dynamic multi-scale hypergraph to learn agent and interaction representations across time.

III. Methodology

A. Problem Formulation

Motion prediction aims to predict the target agent's future states based on its historical states, the historical states of neighboring agents, and the scene context.

Let $S^- \in \mathbb{R}^{N_a \times T_p \times S_a}$, $S^+ \in \mathbb{R}^{N_a \times T_f \times S_a}$ be the past states and future states of agents respectively, where the N_a is the total number of agents, T_p, T_f are past time steps and future time steps, and S_a is the number of state information. The i th agent's state at time step t is represented as a $1-d$ vector $s_t^i \in \mathbb{R}^{S_a}$, including the location, velocity, yaw angle, etc. $M_{in} \in \mathbb{R}^{N_m \times n \times S_m}$ denotes the input scene context, where N_m is the number of map polylines, n is the number of map points in each polyline and S_m is the number of attributes(e.g., position, unit direction, and type) of each map point. The input $\mathbf{X}$ contains each agent's historical states and the scene context.

$$\mathbf{X} = \left[S^-, M_{in}\right]$$

where $S = \{s^1, s^2, ..., s^{N_a}\}$ is the historical states of total N_a agents. Assume our model predicts K trajectories in total. We predict probability p_t, and a set of parameters $(\mu_x, \mu_y, \sigma_x, \sigma_y, \rho)_t$ for the GMM with K components, representing the distribution of predicted trajectories of each future time step t.

B. HHGNN

In this paper, we propose a heterogeneous hypergraph-based motion prediction model, HHGNN. An illustration of model architecture is shown in Figure 2. Groupnet [33] utilizes the nature of hypergraph to capture the muti-scale interaction among moving agents and has achieved great performance in multiple pedestrian motion datasets. Inspired by both Groupnet and Hypertron [31], we propose HHGNN. This model aims to capture the interactions of varied granularity among agents while preserving the agent's heterogeneity of interactions. Groupnet ignores the context information and does not explicitly support heterogeneous driving scenarios. Hypertron utilizes a rasterization encoding method for road map representation, which could be more carefully designed. Our work, HHGNN, is designed to handle complex driving scenarios with heterogeneous agents and incorporate rich scene context to further fuel motion prediction.

C. Encoding Process

To cope with the heterogeneous agents, this study suggests the utilization of a shared history encoder for a particular type of traffic participant. For example, when traffic participants can be categorized into three groups (Vehicle, Pedestrian, or Cyclist), the history encoder will comprise three type-specific encoders. These history encoders are employed to process the historical states of individual participants, extracting their dynamic features. This work leverages Long Short-Term Memory (LSTM) [34] to efficiently capture temporal dependencies along the state sequence.

Assuming there are $|C|$ agent categories, the category i th agent is c, and its past state information $S^i \in \mathbb{R}^{T_p \times S_a}$. $LSTM^c$ take S^i as its input:

$$V_{in}^i = LSTM^k(S^i)$$

where $V_{in}^i \in \mathbb{R}^{T_p \times D}$, D is the feature dimension.

Given the extracted feature $V_{in} \in \mathbb{R}^{N_a \times T_p \times D}$, we fuse V_{in} along the time dimension by simply applying an MLP.

$$V_f = MLP(V_{in})$$

where $V_f \in \mathbb{R}^{N_a \times D}$, which are the final extracted agent representations.

Road map $M_{in} \in \mathbb{R}^{N_m \times n \times S_m}$ is first encoded by a PointNet-like [35] polyline encoder as:

$$M_f = MaxPooling(\mathscr{F}_m(M_{in}))$$

where $M_f \in \mathbb{R}^{N_m \times D}$ and $\mathscr{F}_m$ is an MLP. MaxPooling summarises scene context features by downsampling.

D. Hypergraph Construction

Unlike other data formats, such as literature citation or social network data, trajectory data does not have natural hyper-edges. Therefore, hyperedges must be manually derived first from the trajectory data to construct hypergraphs. Hypergraph construction includes two subsequent processes: constructing the affinity matrix of the agents and deriving hyperedges based on the affinity matrix.

1) Affinity Matrix Construction: In recent works, two major ways exist to construct the affinity matrix of the interacting agents. First, the weight-by-distance method either sets up edges to K nearest agents, where K is a pre-defined hyperparameter, or it sets up edges to agents within a pre-defined radius R. However, the weight-by-distance approach has at least two disadvantages: the assumption that the closer the agent is, the more likely it strongly influences the ego-agent, and agents' interactions from a long distance are ignored. This work leverages the other way, the attention-based method, to reflect the correlations of agents. Specifically, let $q_i = v_i \in \mathbb{R}^D$, $q_j = v_j \in \mathbb{R}^D$ the scaled dot-product attention between i th agent and j th agent is

$$\alpha(i,j) = \frac{q_i^T q_j}{\sqrt{D}}$$

The (i,j) th entry of affinity matrix $\mathbb{A} \in \mathbb{R}^{N_a \times N_a}$ is

$$\mathbb{A}(i,j) = \alpha(i,j)$$

2) Hyperedge Forming: With the derived affinity matrix $\mathbb{A}$, this work leverages a hyperedge forming schema from Groupnet [33].

Given affinity matrix $\mathbb{A}$, we pre-define a set of increasing group sizes $G = \left[g^1, g^2, g^3, ...\right]$. For each agent, the goal is to find a highly related subset of agents at scale g, constructing N_a hyperedges in total. When deriving g scale

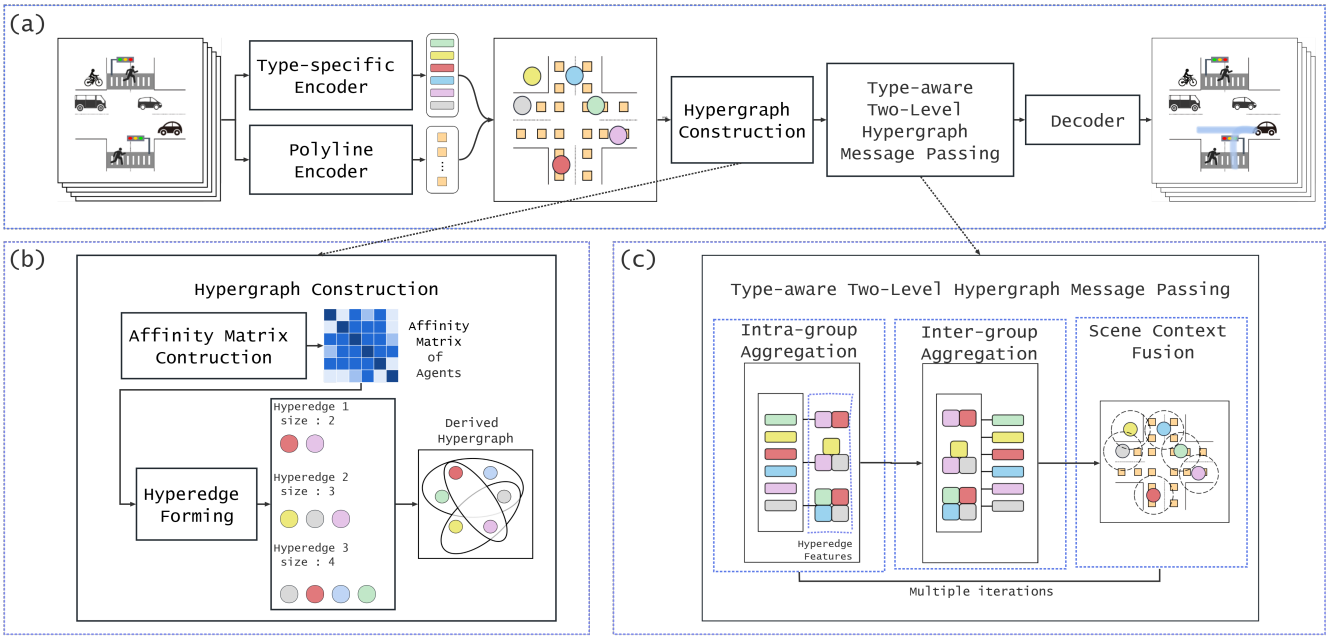


Fig. 2. The architecture of our proposed HHGNN. (a) indicates the overall workflow of the proposed model HHGNN. The implementation of type-specific encoder and decoder are presented in section III-C, section III-F, respectively (b) indicates the Hypergraph construction module, which derives an interaction hypergraph from the encoded agents' features. The inputs to this module are embeddings of both the target agent and other agents. Deriving details are illustrated in section III-D. (c) indicates the TTHMS module. This module includes three phases: intra-group aggregation, inter-group aggregation, and scene context fusion. The output of this module includes interaction features and scene context features, which will be taken as input to the subsequent decoder layer as in (a). Detailed implementation of this module is displayed in section III-E.

hyperedge associated with i th agent, we add $v_i \in \{V_f\}$ into the hyperedge first and successively select other agents with maximum affinity scores until g agents are selected. Each agent will be assigned a hyperedge at group size g. Therefore, N_a hyperedges are formed for each scale as follows:

$$E^g = \left[e_1^g, ...e_{N_a}^g\right]$$

The derived hypergraph with group size g can be represented as $\mathbb{G} = \left[V_f, E^g\right]$. The hyperedges are represented by an incidence matrix $H^s \in \mathbb{R}^{|\mathbf{V}|\times|\mathbf{Eg}|}$ where $H^s(i,j) = 1$, if the i th node lies within j th hyperedge. We refer the readers to Groupnet [33] for more details on deriving hyperedges.

E. Type-Aware Two-Level Message Passing

In this work, the interaction modeling leverages a Type-Aware Two-Level Hypergraph Message Passing module (TTHMS) inspired by [36], whose components are the intra-group aggregation layer, inter-group aggregation layer, and scene context fusion layer. Type-Aware indicates that we explicitly preserve the interacting agents' type information by including learnable hyperedge-type embeddings. Two-level stands for intra-group level aggregation and inter-group level aggregation. We stack multiple layers of the TTHMS module. For the l th layer, the agent features are $V^l \in \mathbb{R}^{N_a \times D}$, where $V^0 = V_f$.

1) Intra-group Aggregation Layer: Aiming to derive a representation for the interacting group, we represent the hyperedge e_i with two elements: $\mathbf{e_i^{attr}}$, which is a collective embedding obtained by pooling among all the agent embedding associated with the hyperedge e_i, representing the interaction group attribute, $\mathbf{e_i^{type}}$, which is a learnable hyperedge-type embedding, representing the interaction group type.

Specifically, for the i th hyperedge e_i, the $\mathbf{e_i^{attr}}$ is calculated by aggregating the nodes within e_i:

$$e_i^{attr} = \mathscr{F}_a\left(\sum_{v_j \in e_i} v_j^l\right) \in \mathbb{R}^D$$

where v_j^l is the extracted agent's feature in lth layer, $\mathscr{F}_a$ is an aggregation function. $\mathscr{F}$ can be average pooling, max pooling, or simply an MLP.

Assuming within hyperedge e_i of group size g, the agents categories are $\{c_i\}_{i=1}^g$, and there are C_d distinct agent categories. We first encode the category variables through one-hot encoding. Then, we obtain the $\psi(e_i)$ by calculating the sum of one-hot encoded variables $\{c_i\}_{i=1}^g$, corresponding to the categories of $\{v_i^l\}_{i=1}^g$:

$$\psi(e_i) = \sum_{v_j^l \in e_i} c_j \in \mathbb{R}^{C_d}$$

Assuming a total of T pre-defined interaction types, the shape of the learnable type embedding matrix C is $T \times D$. The specific learnable hyperedge-type embedding e_i^{type} for hyperedge e_i is derived as follows:

$$e_i^{type} = softmax(\mathscr{F}_t(\left[\psi(e_i), e_i^{attr}\right])) \times C \in \mathbb{R}^D$$

where $[\ :\ ]$ denotes vector concatenation, and $\mathscr{F}_t$ is simply implemented by an MLP.

2) Inter-group Aggregation Layer: Given the learned hyperedge representation $e = [e^{attr}, e^{type}] \in \mathbb{R}^{2D}$, the agents' embedding is updated regarding the associated hyperedges. We add a pre-activation residual connection. Specifically, the embedding of i th agent v_i^l is updated as :

$$v_i^l = \mathscr{F}_v \left(\sum_{e_j \in E_i} w_{ij} \cdot W[v_i^l : e_j] + v_i^l \right)$$

$$w_{ij} = \frac{exp(LeakyReLU(\mathscr{F}_w([v_i^l : e_j])}{\sum_{e_k \in E_i} exp(LeakyReLU(\mathscr{F}_w([v_i^l : e_k])}$$

where $\mathscr{F}_v$ and $\mathscr{F}_w$ are trainable MLPs, W is an learnable linear transformation $W \in \mathbb{R}^{D \times 3D}$, E_i is a subset of hyperedges which include ith agent, and $[\ :\ ]$ denotes concatenation.

3) Scene Context Fusion Layer: To incorporate the scene context information, we propose an iterative approach to collect scene context information dynamically. We utilize multi-head attention [37] to fuse the updated agent features and scene context features as follows:

$$M^{l+1} = MHA(q = V^l, k = \kappa(M_f), v = \kappa(M_f)) \in \mathbb{R}^{N_a \times D}$$

where V_l is the agent features after l th aggregation, κ denotes the sub-sampling process to select k closest encoded map polylines, k is a pre-defined hyperparameter. M^l is dynamically updated across layers, resulting in more fine-grained scene context features. Then, we fuse the scene context features, updated agent features, and original agent features as follows :

$$V^{l+1} = \mathscr{F}_f \left([V^{l+1} : V_f : M^{l+1}] \right) \in \mathbb{R}^{N_a \times D}$$

, where $\mathscr{F}_f$ is an trainable MLP, and $[\ :\ ]$ denotes vector concatenation. With the agent features updated, V^{l+1} are input to the next layer. On the last layer, v^o and M^o are input to the decoder.

F. Decoding Process

Due to the stochastic, multi-modal nature of agents, HHGNN predicts distributions of future trajectories parameterized as Gaussian Mixture Model (GMM) instead of directly regressing the trajectory, as is done in other works [18], [33], [17], [32], [38], [39]. The final output parameters for GMM is a fusion of multiple modes. By incorporating multiple models' complementary information, we can enjoy the benefits of a higher-capacity model with lower statistical variance [18].

Given final agent features V^0 and scene context features M^o, We predict probability p_k, and a set of parameters $(\mu_x^t, \mu_y^t; \sigma_x^t, \sigma_y^t; \rho^t)_k$ of each Gaussian component k at each future time step $t \in \{1, 2, ..., T_f\}$ as follows:

$$\theta_{1:T_f} = MLP([V^{l+1} : M^l])$$

where $\theta_t \in \mathbb{R}^{K \times 6}$, consisting of K Gaussian components $\mathscr{N}_{1:K}(\mu_x, \sigma_x; \mu_y, \sigma_y; \rho)$, K is the number of predicted trajectories, $[\ :\ ]$ denotes vector concatenation.

The trajectories distributions of agent's location (x, y) ,at future time step t, are parameterized as:

$$P_t(x, y) = \sum_{k=1}^{K} p_k \cdot \mathscr{N}_k(x - \mu_x^t, \sigma_x^t; y - \mu_y^t, \sigma_y^t; \rho^t)_k$$

where $P_t(x, y)$ is the probability of the agent being at (x, y). We derive the final predicted trajectories by taking the geometric centers $(\mu_x, \mu_y)_t$ of the GMM component at each time step t.

G. Training Strategy

For each target agent, we predict K possible trajectories, each consisting of T_f coordinates $\{(x_k^t, y_k^t)\}_{t=1}^{T_f}$ for $k = 1, ..., K$. The likelihood $\mathscr{L}$ of the prediction is as follows:

$$\mathscr{L} = \sum_{k=1}^{K} p_k \prod_{t=0}^{T_f} \cdot \mathscr{N}_k(x - \mu_x, \sigma_x; y - \mu_y, \sigma_y; \rho)$$

We adopt negative log-likelihood loss as an objective function for optimization. Among K predicted trajectories, we select the trajectory based on the $L1$ norm between the endpoint of each predicted trajectory and the ground truth trajectory. Only the corresponding Gaussian component is selected for optimization to ensure prediction multi-modality.

IV. Real-world Dataset Validation

A. Dataset and Metrics

We train and evaluate our model on the WOMD [5], which includes 103,354 scenes every 20 seconds long at 10Hz. It contains trajectories of different types of traffic agents, i.e., vehicles, bicycles, and pedestrians, and all scenes have at least one vehicle, 20% of scenes have at least four pedestrians, and 16% of scenes have at least one cyclist. The state of a traffic agent at a time step includes its position, velocity, and yaw angle. The dataset also includes the state of transportation facilities, i.e., traffic lights and road map information. The state of a traffic light contains its position, light state, and countdown number. The state of a road contains its position and type (i.e., freeway, broken single white, and stop sign).

We follow the official evaluation metrics of the WOMD benchmark [5] and focus on several important ones most considered in recent works [40], [41], [42]. The minimum average displacement error (minADE) is the minimum average Euclidean distance between the K-predicted trajectories and the ground-truth trajectory. In contrast, the minimum final displacement error (minFDE) focuses on the difference at the last time steps. The Miss Rate is a metric used to evaluate the accuracy of object predictions. It measures the proportion of predictions that fail to meet specific proximity criteria to the ground truth trajectory. If, for one prediction, the displacement vector at a given time doesn't fall within defined lateral and longitudinal thresholds, it's counted as a miss. mAP involves categorizing the actual trajectories of agents into distinct buckets (e.g., Going straight, Left turn). Subsequently, predicted trajectories are ranked based on their confidence scores. If a prediction falls short of predefined

TABLE I

COMPARISON OF HHGNN AND BASELINES ON THE WOMD.

Waymo Open Motion Prediction (k=6, t=8s)				
Methods	minADE ↓	minFDE ↓	MR ↓	mAP ↑
MotionCNN	1.1867	2.5312	0.2455	0.1682
DenseTNT	1.8061	2.5605	0.1864	0.2589
Waymo Lstm	1.7398	4.2971	0.4915	0.0985
HHGNN (Ours)	**0.9437**	**1.9527**	**0.1726**	**0.2930**

TABLE II

PER-CLASS PERFORMANCE OF MOTION PREDICTION ON THE WOMD.

Waymo Open Motion Prediction (k=6, t=8s)				
Category	minADE ↓	minFDE ↓	MR ↓	mAP ↑
Vehicle	1.2247	2.4386	0.2393	0.3220
Pedestrian	0.5816	1.1435	0.1027	0.3029
Cyclist	1.0245	2.2760	0.1758	0.2541
Avg	0.9437	1.9527	0.1726	0.2930

thresholds, it's treated as a false positive; otherwise, it's considered a true positive. Only the prediction with the highest confidence score attains true positive average precision, calculated for each bucket by generating Precision-Recall curves. Ultimately, the mean average precision (mAP) is determined by averaging the bucket-wise Average Precision.

B. Comparison Experiments

HHGNN is compared with the following models on the WOMD, as shown in Table I. Distance measures are in terms of meters.

MotionCNN [43]: MotionCNN is a simple but effective multimodal motion prediction model based on Convolutional Neural Networks.

DenseTNT [6]: DenseTNT is a goal-based motion prediction model. It extracts features of the HD map using a sparse encoding method. Then it uses a dense goal encoder to generate the probability distribution of the goals and regresses the complete trajectories.

Waymo LSTM Baseline: Waymo LSTM Baseline is an official baseline implemented by feeding the agent's history states into a vanilla LSTM network.

C. Per-class Performance

Table II shows the per-class performance of HHGNN on the WOMD. Distance measures are in terms of meters. The dataset has three classes of traffic participants: Pedestrian, Cyclist, and Vehicle.

TABLE III

COMPARISON OF HHGNN AND ABLATIVE IMPLEMENTATIONS ON THE WOMD.

Waymo Open Motion Prediction (k=6, t=8s)					
Interaction	Map	minADE ↓	minFDE ↓	MR ↓	mAP ↑
✗	✗	1.4428	2.9541	0.4126	0.1276
✓	✗	1.1081	2.1715	0.3126	0.2495
✗	✓	1.0397	2.0845	0.2438	0.2641
✓	✓	0.9437	1.9527	0.1726	0.2930

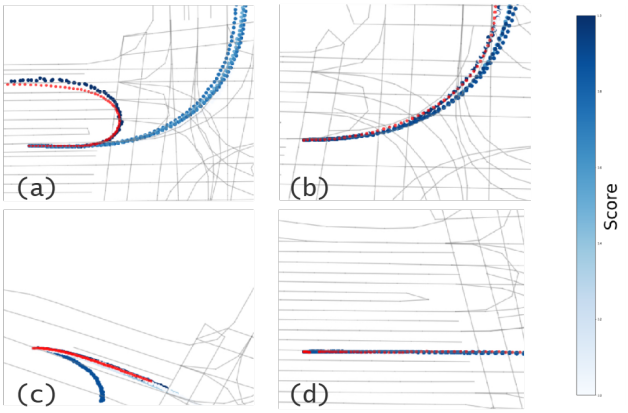


Fig. 3. The red color indicates the ground truth future trajectory, and the blue color indicates the predictions from our HHGNN model. The deeper the color, the higher the probability of the trajectory.

D. Ablative Study

This work conducts an ablative study on the WOMD to show the effectiveness of the TTHMS module and scene context fusion layer. Table III shows the performance evaluation of the variants. Interaction means the usage of the TTHMS module. Map refers to the usage of the scene context fusion layer. The proposed HHGNN outperforms all the variants, which supports the intuition that the interactions among agents are essential, and the scene context serves as a complementary information source that fuels accurate prediction.

E. Qualitative Results

Fig. 3 shows our best model's predictions on WOMD under four driving scenarios. In (a), the agent makes a left U-turn; In (b), the agent turns left; In (c), the agent goes straight right; In (d), the agent goes straight;

V. CONCLUSION

In this work, we have proposed a new approach to effectively represent the complex interactions between heterogeneous agents. Instead of focusing on pairwise interactions, we propose a heterogeneous hypergraph-based framework to represent group interactions. We leverage learnable type embeddings to tackle groups' heterogeneity. Besides, we integrate the scene context fusion layer to utilize the scene context information. We did an ablative study to verify the effectiveness of the proposed modules. Experiments on the WOMD demonstrate the effectiveness of the model.

We plan to investigate how to derive hyperedge-type embeddings that better represent the group's heterogeneity. Additionally, we will explore the combination of regular graphs and hypergraphs for interaction modeling.

ACKNOWLEDGMENT

This work was partially supported by the grants of the National Key Research and Development Project (2021YFB1714400) of China and Guangdong Provincial Key Laboratory (2020B121201001).

REFERENCES

[1] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2702–2719, 2020.

[2] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8740–8749.

[3] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[4] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[5] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9710–9719.

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

[7] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 522–11 530.

[8] L. Zhang, P. Li, J. Chen, and S. Shen, "Trajectory prediction with graph-based dual-scale context fusion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 374–11 381.

[9] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[10] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8994–9003.

[11] S. Park, M. Cap, J. Alonso-Mora, C. Ratti, and D. Rus, "Social trajectory planning for urban autonomous surface vessels," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 452–465, 2021.

[12] J. Patrikar, B. Moon, J. Oh, and S. Scherer, "Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2525–2531.

[13] Z. Huang, R. Li, K. Shin, and K. Driggs-Campbell, "Learning sparse interaction graphs of partially detected pedestrians for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1198–1205, 2022.

[14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 2255–2264.

[15] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 961–971.

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

[17] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 86–99. [Online]. Available: https://proceedings.mlr.press/v100/chai20a.html

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

[19] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transformer with global intention localization and local movement refinement," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6531–6543, 2022.

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

[21] ——, "Gohome: Graph-oriented heatmap output for future motion estimation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9107–9114.

[22] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "Tnt: Target-driven trajectory prediction," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 895–904. [Online]. Available: https://proceedings.mlr.press/v155/zhao21b.html

[23] C. Wang, Y. Wang, M. Xu, and D. J. Crandall, "Stepwise goal-driven networks for trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2716–2723, 2022.

[24] J. Lian, F. Yu, L. Li, and Y. Zhou, "Causal temporal–spatial pedestrian trajectory prediction with goal point estimation and contextual interaction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 499–24 509, 2022.

[25] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 683–700.

[26] W. Zeng, M. Liang, R. Liao, and R. Urtasun, "Lanercnn: Distributed representations for graph-centric motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 532–539.

[27] X. Mo, Y. Xing, and C. Lv, "Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction," 2021.

[28] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019, pp. 2375–2384.

[29] Y. Liu, L. Yao, B. Li, X. Wang, and C. Sammut, "Social graph transformer networks for pedestrian trajectory prediction in complex social scenarios," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1339–1349.

[30] S. Liu, X. Chen, Z. Wu, L. Deng, H. Su, and K. Zheng, "Hega: Heterogeneous graph aggregation network for trajectory prediction in high-density traffic," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1319–1328.

[31] Y. Tian, X. Huang, R. Niu, H. Yu, P. Wang, and X. Sun, "Hypertron: Explicit social-temporal hypergraph framework for multi-agent forecasting," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, L. D. Raedt, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2022, pp. 1356–1362, main Track. [Online]. Available: https://doi.org/10.24963/ijcai.2022/189

[32] C. Xu, Y. Wei, B. Tang, S. Yin, Y. Zhang, and S. Chen, "Dynamic-group-aware networks for multi-agent trajectory prediction with relational reasoning," 2022.

[33] C. Xu, M. Li, Z. Ni, Y. Zhang, and S. Chen, "Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6498–6507.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[36] D. Arya, D. K. Gupta, S. Rudinac, and M. Worring, "Adaptive neural message passing for inductive learning on hypergraphs," 2021.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[38] L. Fang, Q. Jiang, J. Shi, and B. Zhou, "Tpnet: Trajectory proposal network for motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 6797–6806.

[39] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7577–7586.

[40] X. Huang, G. Rosman, I. Gilitschenski, A. Jasour, S. G. McGill, J. J. Leonard, and B. C. Williams, "Hyper: Learned hybrid trajectory prediction via factored inference and adaptive sampling," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2906–2912.

[41] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "Trafficbots: Towards world models for autonomous driving simulation and motion prediction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1522–1529.

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

[43] S. Konev, K. Brodt, and A. Sanakoyeu, "Motioncnn: a strong baseline for motion prediction in autonomous driving," 2022.