Multi-Modal Interactive Agent Trajectory Prediction Using Heterogeneous Edge-Enhanced Graph Attention Network

Xiaoyu Mo, Zhiyu Huang, *Student Member, IEEE,* and Chen Lv, *Senior Member, IEEE,*

***Abstract*—Simultaneous predicting two highly interactive agents' future trajectories while considering the multi-modal nature of navigation is essential for the safe and efficient operation of connected automated vehicles under complex driving situations in the real world. The interaction prediction task is challenging, as the motions of traffic participants are affected by many factors, including their individual dynamics, their interactions with surrounding agents, and the road structure. In this work, we design a three-channel framework based on Heterogeneous Edge-enhanced graph ATtention network (HEAT) to understand the complex driving scene and simultaneously predict pairs of future trajectories of two highly interactive agents with assigned probabilities.**

***Index Terms*—Trajectory prediction, connected vehicles, graph neural networks, heterogeneous interactions.**

I. Introduction

ACCURATELY predicting future trajectories of two moving objects highly interacting with each other is an essential task in the field of self-driving. With predicted trajectories of surrounding agents, autonomous vehicles can make decisions in advance and avoid possible accidents. This increases the safety, efficiency, and comfort of autonomous driving. However, Trajectory prediction is challenging, especially in urban driving scenarios, since the motion of an agent is affected by many factors: its own dynamics, its interaction with neighboring agents, the road structure, and the multi-modality of motion.

Graph-based interaction representation attracts more and more interest in the field of trajectory prediction, and it gives rise to the application of graph neural networks on trajectory prediction [1]–[3]. SCALE-Net [1] constructs edge attributes with the relative measurements between two agents and employs Edge-enhanced Graph Convolutional Neural Network to summarize interactions considering edge attributes. TNT [3] adopts VectorNet [2] as an interaction feature extractor and further considers the multi-modality of driving by predicting multiple trajectories conditional on selected target points in the map. It shares the drawbacks of VectorNet. Authors of [4] represent inter-agent interactions as a directed edge-featured heterogeneous graph and designs a heterogeneous edge-enhanced graph attention network (HEAT) to model the interaction. HEAT is able to simultaneously predict multi-agent trajectories, which is suitable for the interaction prediction task, but it does not cover the multi-modality of motions.

We modify our recent work [4] for simultaneously predicting future trajectories of two highly interacting agents for both urban and highway driving by jointly considering agents' individual dynamics, their interactions, the road structure, and the multi-modality of motion. Agents' past states and a top view image of the interested area are assumed to be available leveraging the vehicle-to-vehicle and vehicle-to-infrastructure communications. The designed framework has three encode channels handling agents' individual dynamics, interactions, and road structure, and two decode channels producing pairs of joint trajectories and the mode probability assigned to each mode. For agents' dynamics, this work places each agent in its exclusive coordinate system to eliminate the impacts of coordinate shifting.

Xiaoyu Mo, Zhiyu Huang, and Chen Lv are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore. (e-mail: xiaoyu006@e.ntu.edu.sg, zhiyu001@e.ntu.edu.sg, and lyuchen@ntu.edu.sg)

II. Structure Overview

This section introduces the interaction prediction task and the high-level structure of the proposed method.

A. Input and Output

The task of the interaction prediction is to predict pairs of future trajectories of two highly interacting agents over the next 8 seconds given 1-second past tracks of agents and a corresponding local map. In our setting, at time a time t, the input $\mathbf{X}_t$ contains each agent's historical states and the map of the scene: $\mathbf{X}_t = [\mathcal{H}_t, \mathcal{M}_t]$, where $\mathcal{H}_t = \{h_t^1, h_t^2, \cdots, h_t^N\}$ contains the historical states of N agents at time t, $\mathcal{M}_t$ contains two local maps centered at the corresponding target agents $\{m_t^1, m_t^2\}$. Agent i's historical states at time t is represented by $h_t^i = [s_{t-T_h+1}^i, s_{t-T_h+2}^i, \cdots, s_t^i]$, with T_h as the traceback horizon. The state s_t^i contains agent i's position, velocity, and yaw angle at time t. Without loss of generality, we number the target agents as agent 1 and agent 2 respectively.

Following our recent work, we represent inter-agent interaction as a directed edge-featured heterogeneous graph for interaction prediction. Each node represents a traffic participant with a specific type and contains its feature extracted from a sequence recorded in its exclusive coordinate system [4]. An edge from node j to node i means that node i's behavior is influenced by node j, and the edge attribute is relative measurements of node j to node i, e.g., position, velocity,

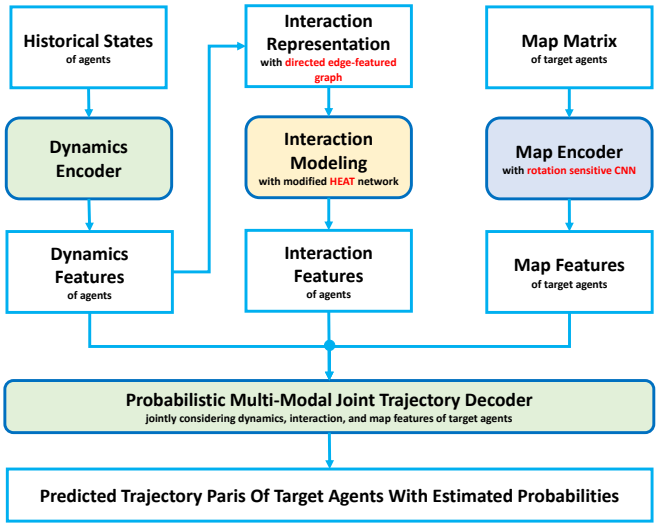

Fig. 1. High-level structure of the proposed multi-modal interaction prediction framework.

and yaw angle. The edge type is a concatenation of the type of node j and node i. We place all agents in their exclusive coordinate systems, which are non-sensitive to translation and rotation, and construct a directed heterogeneous graph representing these inter-agent relationships. An edge e_{ij} pointing from agent j to agent i is constructed if agent j is within a predefined neighborhood of agent i. Each valid edge e_{ij} is assigned with edge attribute and edge type. Then the edge set is: $E = \{e_{ij}\}_{(j\in\mathcal{N}_i)}, \quad i = 1, \cdots, n$, where i and j is the indexes of agents and $\mathcal{N}_i$ is the neighborhood of agent i. Self-loop e_{ii} is included in the edge set.

The output $\mathcal{F}_t$ is a set of K paired interaction trajectories, and $\mathcal{P}_t$ contains assigned probabilities: $\mathcal{F}_t = \{(f_t^1, f_t^2)_1, (f_t^1, f_t^2)_2, \cdots, (f_t^1, f_t^2)_K\}$, $\mathcal{P}_t = \{p_1, p_2, \cdots, p_K\}$, where $f_t^i = [(x_{t+1}^i, y_{t+1}^i), \cdots, (x_{t+T_f}^i, y_{t+T_f}^i)]$ is a sequence of the predicted 2D coordinates of agent i over a prediction horizon T_f, $(f_t^1, f_t^2)_k$ is the predicted trajectory pair of mode k, and p_k is the assigned probability of mode k.

B. High-Level Architecture

The proposed framework has three channels for dynamics, interactions, and map features, respectively. It then jointly considers these features to predict paired multi-modal future trajectories of interactive agents with estimated probabilities. The dynamics encoder summarises the dynamics features of individual agents from their individual historical state. The interaction encoder aggregates information from surrounding agents to model interaction among agents via an attention mechanism. The Map encoder applies a rotation-sensitive CNN on the pictorial local map to extract road structure. Within the probabilistic multi-modal joint trajectory decoder, the trajectory predictor produces pairs of trajectories of target agents and the mode probability estimator estimates probabilities of each trajectory pair. See Fig. 1 for an illustration of this three-channel framework.

III. Method

This section elaborates the proposed method for interaction prediction, consisting of four key components: dynamics encoder, HEAT-based heterogeneous interaction encoder, map encoder, and joint trajectory and mode probability decoders.

A. Dynamics Encoder

The dynamic encoder is shared across all agents to encode individual dynamics from their own historical states. Eq. 1 shows that the encoder is applied to historical tracks of all agents in parallel.

$$R_t = \{r_t^0, r_t^1, \cdots, r_t^N\} = \mathrm{GRU}_{\mathrm{hist}}(\mathrm{Emb}(\mathcal{H}_t)), \tag{1}$$

where $\mathrm{Emb}()$ is a linear transformation embedding the low-dimensional state into a high-dimensional vector space, $\mathrm{GRU}_{\mathrm{hist}}$ is a shared GRU applied to the embedded historical tracks of all vehicles, r_t^i is the dynamics feature of vehicle i at time t, which serves as the node feature in the interaction graph.

B. HEAT-Based Interaction Encoder

*1) **Heterogeneous Interaction Modeling With HEAT**:* To comprehensively model the inter-agent interaction among heterogeneous agents, this work represents the interaction as a directed edge-featured graph. It adopts the Heterogeneous Edge-enhanced graph ATtention network (HEAT) proposed in a recent work [4] to extract interaction features from the graph representation.

Agents' dynamics features R_t are put into their corresponding node in the graph. Then the proposed HEAT is applied to the graph to model the interaction features for all agents simultaneously.

$$G_t = \{g_t^1, g_t^2, \cdots, g_t^N\} = \mathrm{HEAT}_{\mathrm{enc}}(R_t, E_t), \tag{2}$$

where E_t is the edge set containing edge indexes, edge attributes, and edge types, g_t^i is the interaction feature of agent i at time t, and G_t contains interaction features of all agents.

*2) **HEAT Layer**:* The interaction representation described in Sec. II-A should be treated with a graph neural network that can handle the heterogeneity of nodes, directed edges, and continuous edge attributes. We slightly modify the heterogeneous edge-featured graph attention network (HEAT) proposed in [4] for interaction modeling. HEAT can be constructed by stacking HEAT layers. A HEAT layer updates node features by aggregating information from neighborhoods. It first transforms node and edge features accordingly, then aggregates node features via edge-enhanced masked attention (or optional multi-head attention) mechanism. Please note that the HEAT layer here is slightly different from the HEAT layer in [4].

The input to the HEAT layer contains a set of node features: $\mathbf{h} = \{\vec{h}_i | i = 1, \ldots, N\}$, where $\vec{h}_i \in \mathbb{R}^{F_h}$ is the feature vector of node i; A set of edge attributes: $\mathbf{e}^{attr} = \{e_{ij}^{attr} | i = 1, \ldots, n, j = 1, \ldots, N\}$, where $e_{ij}^{attr} \in \mathbb{R}^{F_e^{attr}}$ is the attribute of the edge pointing from node j to node i; In addition, a set of edge types: $\mathbf{e}^{type} = \{e_{ij}^{type} | i = 1, \ldots, n, j = 1, \ldots, N\}$, where $e_{ij}^{type} \in \mathbb{R}^{F_e^{type}}$ is the type of the edge pointing from

node j to node i. It produces a new set of node features: $\mathbf{h}' = \{\vec{h}'_i | i = 1, \ldots, N\}, \vec{h}'_i \in \mathbb{R}^{F'_h}$.

To handle different node features, we apply a shared node feature transformation, parametrized by a matrix $\mathbf{M}$, on raw node features before sending them to the HEAT layer. $\vec{h}_i = \mathbf{M} \cdot \vec{h}_i$, where we reuse $\vec{h}_i$ to represent the projected feature of node i.

Rather than considering edge attributes or edge types as edge features as in existing works, we jointly consider the edge features and types by introducing the edge attribute transformation: $\mathbf{e}_\phi^{attr} = \mathbf{M}_\phi \cdot \mathbf{e}^{attr}$, where $\mathbf{e}_\phi^{attr}$ is the projected edge attributes, and the edge type transformation: $\mathbf{e}_\chi^{type} = \mathbf{M}_\chi \cdot \mathbf{e}^{type}$, where $\mathbf{e}_\chi^{type}$ is the projected edge types.

For an edge pointing from node j to node i, its edge feature: $e_{ij} = [e_{\phi ij}^{attr} \| e_{\chi ij}^{type}]$, is a concatenation of its transformed edge attribute and type. For node i, a concatenated feature vector $e_{ij}^+ = [e_{ij} \| \vec{h}_j]$ represents the feature of node j from node i's point of view considering the edge attribute and type. e_{ij}^+ is then sent to a shared attention mechanism [5], which is a single-layer feed-forward neural network, $\vec{\mathbf{a}}$, applying LeakyReLU non-linearity and softmax normalization. The attention coefficient α_{ij} indicates the importance of the node j to node i jointly considering node and edge features. Inspired by GAT layer [6], which performs masked attention attending over the neighborhood of node i to utilize the structural information of the graph while casting away the edges' feature and type, the HEAT layer performs edge-enhanced masked attention in Eq. 3 to fully consider the graph attributes.

$$\alpha_{ij} = \frac{\exp\left(\text{LeakeyReLU}\left(\vec{\mathbf{a}}^T [\vec{h}_i \| e_{ij}^+]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakeyReLU}\left(\vec{\mathbf{a}}^T [\vec{h}_i \| e_{ik}^+]\right)\right)}, \tag{3}$$

where $\mathcal{N}_i$ is the neighborhood of node i in the graph. The attention coefficients are then used to update feature of node i with a linear combination over its neighborhood.

The feature of node i is updated by calculating a weighted sum of edge-integrated node features over its neighborhood, followed by a sigmoid function.

$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}_h [e_{\phi ij}^{attr} \| \vec{h}_{\kappa i}]\right). \tag{4}$$

Edge types are not included in the edge-integrated feature since it is discrete and already considered in the previous attention mechanism. Similar to GAT [6], HEAT allows running several independent attention mechanisms to stabilize the self-attention mechanism. Then the features out from each attention head are concatenated as the updated feature.

C. The Rotation-Sensitive Map Encoder

Road structure formulates traffic participants' motions in the scene so that it should be considered in the trajectory prediction, especially for urban situations. In this work, we consider a pictorial bird's eye view (BEV) map because it is easy to get and contains enough road information. Since the target agents in this task are fixed to two, we use two map images centered at these two target agents, respectively. A shared CNN is designed to extract the map features from the images. The designed CNN casts off the max-pooling layer to be rotation-sensitive since a map contains directional information, and a rotated map may mislead an agent's navigation.

Eq. 5 shows the map encoder applied to the local map at time t.

$$m_t = \text{FC}_2(\text{Map}_{\text{enc}}(m_t)), \tag{5}$$

where Map_{enc} is the designed map encoder applied to the image of map. FC_2 is a fully connected layer applied to the encoded raw map feature. And m_t is the feature vector of the local map.

D. Probabilistic Multi-Modal Joint Trajectory Decoder

The probabilistic multi-modal joint trajectory decoder consists of two parts, the trajectory predictor and the mode probability estimator. For a target agent, its features from three channels are concatenated as one final encoding and sent to the shared trajectory predictor for the following multi-modal trajectory prediction.

$$\{(f_t^i)_1, (f_t^i)_2, \cdots, (f_t^i)_K\} = \text{FC}_4(\text{GRU}_{\text{dec}}([g_t^i \| d_t^i \| m_t^i])), \tag{6}$$

where $[g_t^i \| d_t^i \| m_t^i]$ is the concatenation of g_t^i, d_t^i, and m_t^i, GRU_{dec} is the trajectory decoder for prediction, and FC_4 is the fully connected layer, mapping decoded features to proper outputs. The mode probability estimator takes as input the concatenation of both target agents' final encoding and outputs a probabilities assigned to each mode.

$$\{p_1, p_2, \cdots, p_K\} = \text{FC}_5([g_t^1 \| d_t^1 \| m_t^1] \| [g_t^2 \| d_t^2 \| m_t^2]), \tag{7}$$

where p_k is the estimated probability of mode k.

IV. Training And Testing

In this section, we provide detail for training the proposed prediction method, including the loss function, data representation, data pre-processing, and implementation.

A. MTP Loss For Joint Prediction

We modify the Multiple-Trajectory Prediction (MTP) loss proposed in [7] for joint multi-modal trajectory prediction. The modified MTP loss takes as input pairs of predicted trajectories and the paired ground truth trajectories and outputs the sum of the classification loss and regression loss. Unlike the original MTP, our modified MTP selects the trajectory pair with the smallest average L2 distance to the ground truth pair as the best mode and calculates the cross-entropy between the estimated mode probabilities and the one-hot representation of the best mode. The regression loss the smoothed L1 loss between the best mode trajectory pair and the ground truth trajectory pair. For more details about the MTP loss, please refer to [7].

B. Processed Data

The raw data is processed to train the proposed method. Missing values in the raw data are interpolated or calculated with available values. A processed data is stored with:

- *Historical states*: The historical states within a 1-second traceback horizon of all agents. The historical states of agent i (position, velocity, and orientation) are stored in its own exclusive coordinate system, with the origin fixed at its current position and the horizontal axis pointing to its current direction.
- Edge indexes: The graph connectivity represented a set of directed edges. A directed edge from node j to node i means that agent j is within a distance of 30 meters to agent i and affects the behavior of agent i.
- Edge attributes: The attributes of all edges. The attribute of an edge from agent j to agent i contains agent j's relative states to that of agent i.
- Edge types: The types of all edges. The type of an edge from agent j to agent i contains a concatenation of the types of agent j and agent i.
- Target masks: The mask of the agents to be predicted. If agent i's future trajectory is to be predicted, its mask is set to 1. Else it is set to 0.
- Local maps: A top view image represents the local maps centered at the target agents. The local map of a target agent is rotated according to its direction. The local map of a vehicle covers a square with 70 meters side length. The local map of a pedestrian covers a square with 35 meters side length. The local map of a cyclist covers a square with 50 meters side length. The local maps are all stored as arrays of shape (3, 140, 140).
- Ground truth future trajectories: The recorded future trajectories of both target agents over the prediction horizon.

After processing, we got $204,178$ data pieces for training and $43,483$ for validation.

C. Implementation Details

The proposed method is implemented with PyTorch. The hyper-parameters of the winner solution are given below. The $\mathrm{Emb}()$ in Eq. 1 is a linear layer mapping a 5-dimensional vector into a 24-dimensional vector. The GRU_{hist} in Eq. 1 is a one-layer GRU network with a hidden size equals 48. A one-layer HEAT network with eight-head attention is used for interaction modeling. The node feature transformation is a linear mapping from 48 to 48. The edge attribute transformation is a linear mapping from 5 to 32. The edge type transformation is a linear mapping from 8 to 32. $\mathrm{LeakyReLU}$ is used as the nonlinear activation. The updated node feature size is 96. For the rotation-sensitive map encoder, Map_{enc} in Eq. 5 is a three-layer convolutional block with $[\#\_filters, kernel\_size, stride] = [8, 8, 4]$ for the first layer, $[16, 6, 4]$ for the second layer, and $[32, 4, 2]$ for the third layer. FC_2 in Eq. 5 is a one-layer fully-connected network with hidden sizes of 128. The structure is [[Conv, LeakyReLU, BatchNorm], [Conv, LeakyReLU, BatchNorm], [Conv, LeakyReLU, BatchNorm], FC_2]. The trajectory decoder $\mathrm{GRU}_{\mathrm{dec}}$ in Eq. 6 is a two-layer GRU network with hidden size equals to 256. It outputs 4 pairs of XY-coordinates for multi-modal prediction. The mode prediction is implemented with a one-layer fully-connected layer with hidden size of 128. It outputs estimated probabilities of four-modal trajectories.

D. Testing

This work is done for the Waymo open dataset challenge. Tab. I shows the performance on the test set in terms of multiple metrics. Note that the results are averaged over different measurement times (3 seconds, 5 seconds, and 8 seconds) across all object types (vehicle, pedestrian, cyclist). Details of these metrics used for the challenge can be found in [8] and the challenge website [9].

TABLE I
PREDICTION PERFORMANCE OF PROPOSED METHOD ON THE WAYMO OPEN DATASET CHALLENGE

	Metrics				
Dataset	***minADE***	***minFDE***	***Miss Rate***	***Overlap Rate***	***mAP***
Testing	1.4197	3.2595	0.7224	0.2838	0.0844

V. CONCLUSION

In this work, we implemented a framework for multi-modal interaction prediction. The framework has three channels covering the target agents' dynamics, interaction among agents, and road structure. It outputs pairs of trajectories of two highly interacting agents along with probabilities assigned to each mode. One possible future work is to predict joint trajectories of more than two highly interacting agents.

REFERENCES

[1] H. Jeon, D. Kum, and J. Choi, "Scale-net: Scalable vehicle trajectory prediction network under random number of interacting vehicles via edge-enhanced graph convolutional neural network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 2020.

[2] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[3] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid *et al.*, "Tnt: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.

[4] X. Mo, Y. Xing, and C. Lv, "Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction," *arXiv preprint arXiv:2106.07161*, 2021.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[7] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.

[8] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," *arXiv preprint arXiv:2104.10133*, 2021.

[9] "Waymo open dataset interaction prediction challenge," https://waymo.com/open/challenges/2021/interaction-prediction/.