IEEE Transactions on Intelligent Transportation Systems, Volume 23, Number 7, July 2022, pp. 9554-9567
DOI:10.1109/TITS.2022.3146300

Multi-Agent Trajectory Prediction with Heterogeneous Edge-Enhanced Graph Attention Network

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

***Abstract*—Simultaneous trajectory prediction for multiple heterogeneous traffic participants is essential for safe and efficient operation of connected automated vehicles under complex driving situations. Two main challenges for this task are to handle the varying number of heterogeneous target agents and jointly consider multiple factors that would affect their future motions. This is because different kinds of agents have different motion patterns, and their behaviors are jointly affected by their individual dynamics, their interactions with surrounding agents, as well as the traffic infrastructures. A trajectory prediction method handling these challenges will benefit the downstream decision-making and planning modules of autonomous vehicles.**

To meet these challenges, we propose a three-channel framework together with a novel Heterogeneous Edge-enhanced graph ATtention network (HEAT). Our framework is able to deal with the heterogeneity of the target agents and traffic participants involved. Specifically, agents' dynamics are extracted from their historical states using type-specific encoders. The inter-agent interactions are represented with a directed edge-featured heterogeneous graph and processed by the designed HEAT network to extract interaction features. Besides, the map features are shared across all agents by introducing a selective gate-mechanism. And finally, the trajectories of multiple agents are predicted simultaneously. Validations using both urban and highway driving datasets show that the proposed model can realize simultaneous trajectory predictions for multiple agents under complex traffic situations, and achieve state-of-the-art performance with respect to prediction accuracy. The achieved final displacement error (FDE@3sec) is 0.66 meter under urban driving, demonstrating the feasibility and effectiveness of the proposed approach.

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

I. Introduction

INTELLIGENT transportation systems (ITS) leveraging connected autonomous vehicles [1] are expected to improve the safety, security, and efficiency of our daily transportation [2]–[5]. Among the technologies of ITS, accurate trajectory prediction of moving objects, e.g., pedestrians [6], [7] and vehicles [8], [9] that share the road with autonomous vehicles, is an important task in this field. With predicted trajectories of surrounding agents, autonomous vehicles can make informed 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, e.g., its own dynamics, its interaction with neighboring agents, and the road structure. Researchers in the field of autonomous driving have proposed many works for trajectory prediction and these methods fall into three categories: physics-based, maneuver-based, and interaction-aware methods [10]. Physics-based methods consider the object's individual dynamics to predict its motion ignoring possible maneuvers restricted by the road structure and neighboring agents' impacts [11]. Maneuver-based methods consider maneuver options and predict trajectory conditioned on maneuvers ignoring the impact of surrounding vehicles [12]. Interaction-aware methods have attracted more and more interests recently in that they: 1) naturally treat driving as an interactive activity; 2) show better performance compared to pure physics-based and maneuver-based methods; 3) can be extended to take physics and maneuvers into account [13]–[16]. Most existing interaction-aware methods represent the motion of all agents in a shared coordinate system, which is sensitive to translation and rotation, and only aim at predicting the trajectory of a single agent [13], [15]–[18]. However, autonomous vehicles should simultaneously predict future states of multiple surrounding agents, e.g., vehicles and pedestrians, to navigate in complex and highly dynamic urban driving scenarios.

This work focuses on simultaneously predicting future trajectories of multiple heterogeneous agents for both urban and highway driving by jointly considering agents' individual dynamics, their interactions, and the road structure. Agents' past states and a top view image of the interested area is assumed to be available leveraging the vehicle-to-vehicle and vehicle-to-infrastructure communications [19].

Since a moving agent's future motion is affected by many factors, including but not limited to its own dynamics, social interactions, and the road structure, ideally a trajectory predictor should consider as many of these associated factors as possible. Considering the availability of the datasets, we propose a three-channel framework for multi-agent trajectory prediction to handle these factors accordingly, enabling the modularized design and analysis. For agents' dynamics, we place each agent in its exclusive coordinate system to eliminate the impacts of coordinate shifting. This is because an agent's recorded states, no matter placed in which coordinate system, can be always converted to its own coordinate system without

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

Yang Xing is with the Centre for Autonomous and Cyber-Physical Systems at Cranfield Univeristy, UK, MK43 0AL. (e-mail: yang.x@cranfield.ac.uk)

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

affecting other agents. For inter-agent interaction, we represent the interaction among agents as an edge-featured heterogeneous graph and designs a novel heterogeneous edge-enhanced graph attention network to model the interaction among agents of different types. For the road structure, a pictorial map is shared across all agents with a gated map selector. Please see Fig. 1 for an overview of the proposed framework.

The main contributions of this work can be summarised as:

- A three-channel framework is proposed for multi-agent trajectory prediction. It jointly considers agents' individual dynamics, their interactions, and the road structure for trajectory prediction.
- A comprehensive and transformation-insensitive interaction representation is proposed based on the edge-featured heterogeneous graph, where the nodes and edges fall into different categories and contain corresponding attributes.
- A novel heterogeneous edge-enhanced graph attention network (HEAT) is designed to model the inter-agent interaction for multi-agent trajectory prediction.
- A gate-based map selector is proposed to allow sharing the map information across all target agents in a selective manner rather than store a local map for each agent or share the same map across all agents.

The remainder of this work is structured as follows: Sec. II introduces existing works most related to this work. Sec. III provides an overview of the proposed method. Sec. IV elaborates the proposed method and its key components. Sec. V validates the proposed method on real-world driving datasets collected from both urban and highway scenarios. Sec. VI concludes this work and outlines possible future improvements.

II. Related works

This section reviews interaction representation for interaction-aware trajectory prediction, various graph neural networks (GNNs) proposed for graph-based tasks, and how GNNs can be applied to trajectory prediction tasks. The distinction and advantage of the proposed interaction representation, graph neural network, trajectory prediction framework are illustrated following each group of related works.

A. Interaction Representation

Interaction-aware trajectory prediction methods have employed many ways to represent inter-agent interactions recently. Convolutional social pooling designs an occupancy grid, where each cell contains the feature of the agent that falls in it, to model the interaction among agents in the grid [13]. The grid representation is modified in [17] to observe only the eight agents that mostly affect the target vehicle's behavior. The grid representation is applicable to highway driving since the highway is almost straight and can be easily divided into a grid. But this is not the case for urban driving. Therefore, to model interaction beyond highway driving, multi-agent tensor fusion (MATF) models the interaction by aligning agents' individual features to a top-view image of the driving scene [18]. However, it still ignores the relationships among agents. More and more recent works represent interactions as a graph, where each node represents an agent and the edge represents the inter-agent relationship. Authors of [20] propose to represent the inter-vehicle interaction as a homogeneous directed graph for highway driving, where each vehicle is connected to its up to eight neighbors. GRIP also uses a homogeneous graph to model the interaction [14]. The drawback of this kind of method is that the homogeneous graph ignores the type of agents. On the other hand, ReCoG proposes to represent the interaction as a heterogeneous graph, where a node represents either an agent or a map and an agent is connected to other agents within a neighborhood [15]. ReCoG ignores the edge attributes between nodes. VectorNet [21] and TNT [22] both use a hierarchical heterogeneous graph to represent the interaction, where each object is represented by a sub-graph, and all the objects are then represented by a fully-connected graph. Nonetheless, these methods fail to consider the edge attributes between nodes. SCALE-Net considers edge attributes and proposes to represent the interaction with an edge-featured homogeneous graph, where the edge feature contains relative states between two connected agents [23]. It ignores the heterogeneity of traffic participants. Social-WaGDAT proposes to generate a dynamic pair of history and future graphs for each time step, yet the nodes are assumed to be homogeneous and with a fixed number [16]. EvolveGraph learns an interaction graph that considers the heterogeneity of nodes and edges' types and directions [24]. However, the edge attribute is not considered.

Representing inter-agent interaction as a graph is more natural than using an image or grid. However, most existing graph representations place all the agents on the same target-centered coordinate system, which is suitable for single-agent trajectory prediction but can hardly generalize to multi-agent situations because of the effects of coordinate translation and rotation. SCALE-Net places all agents in their own exclusive coordinates system for generalization and uses edge attributes to preserve spatial relationships among agents [23]. However, the graph representation in SCALE-Net is not comprehensive to cover the heterogeneity of agents and their relationships for trajectory prediction. In this work, we propose to represent the inter-agent interaction in exclusive coordinate systems as a directed heterogeneous edge-featured graph, where different agents are represented by different nodes and the edge between two agents is assigned with both attribute and type.

B. Graph Neural Networks

Neural networks have proven their powerful expression ability on tasks with well-structured data, e.g., image classification [25] with grid-like data and machine translation [26] with chain-like data. However, there are many interesting tasks with data represented in the form of graph [27]. More and more recent works are proposed to generalize neural networks to the graph domain. These works are either spectral [28]–[30] or non-spectral approaches [31]–[33]. Spectral methods, e.g., graph convolutional network (GCN) [30], depend on Laplacian eigenbasis of the graph, which is hard to calculate for a large graph, while non-spectral methods, e.g., graph attention network (GAT) [33], perform information aggregation only on

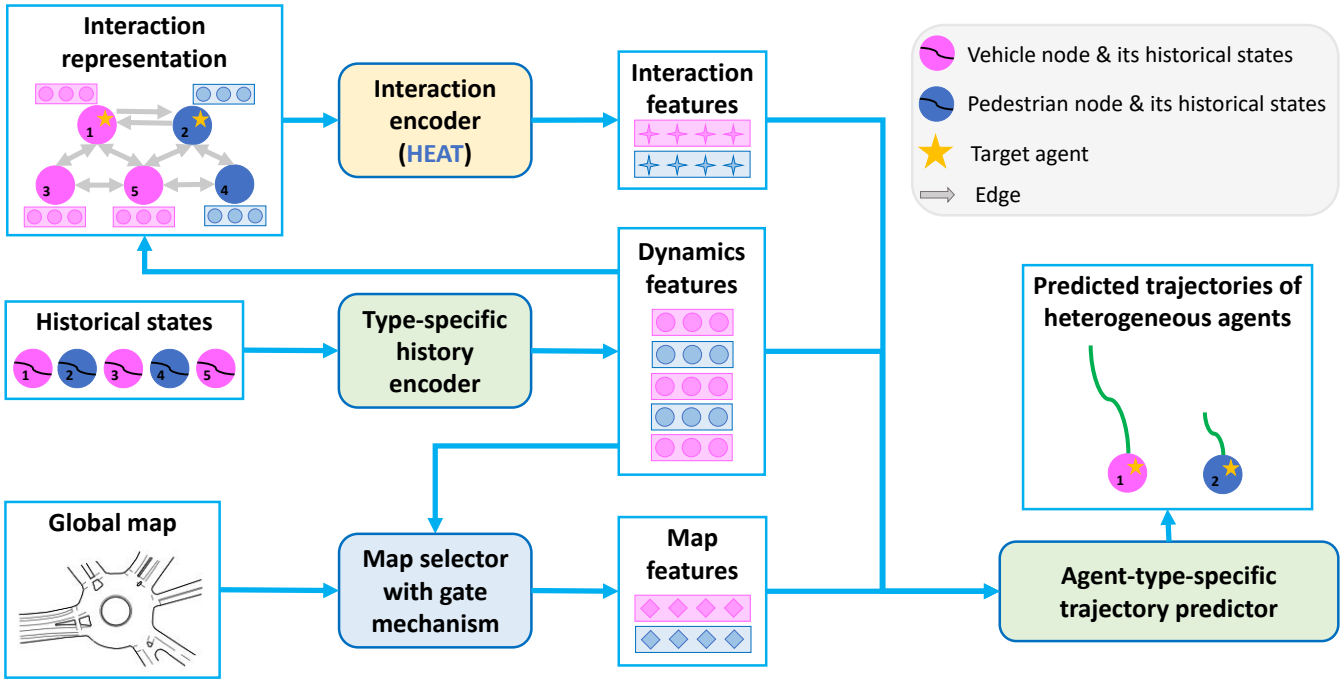


Fig. 1. **Proposed multi-agent trajectory prediction framework.** Historical states of agents are encoded with type-specific history encoders to get their individual dynamics features. Inter-agent interaction is represented by an edge-featured heterogeneous graph with each node contains the dynamics feature of its corresponding agent. Then the proposed HEAT is applied to the interaction graph to extract interaction features for all agents in parallel. Map feature for an agent is processed with a gate-mechanism by considering its dynamics on the map. Then features from these three channels are concatenated and fed to the agent-type-specific trajectory predictor to predict future trajectories of all target agents.

the local neighborhood, avoiding heavy calculation of Laplacian eigenbasis. However, GAT is designed for homogeneous graph [33]. Although it introduces an attention mechanism to aggregate features from neighboring nodes according to edge connections, the edge attribute is not considered. To address this issue, edge enhanced graph neural network (EGNN) considers continuous multi-dimensional edge feature by using each dimension to guide an individual attention operation [34]. Convolution with Edge-Node Switching graph neural network (CensNet) utilizes the line graph of the original undirected graph and designs convolution operations on both graphs to explore edge features [35]. NENN incorporates node-level and edge-level attentions in a hierarchical manner and learns the node and edge embeddings in the corresponding level [36]. EGAT extends GAT with edge embedding to handle continuous edge features of undirected homogeneous graphs [37]. Nonetheless, the heterogeneity of nodes and edges in a graph is ignored in the above-mentioned works. Heterogeneous graph attention network (HAN) proposes to handle heterogeneous nodes in a graph with a hierarchical attention mechanism, where the node-level attends over meta-path-based neighbors and the semantic-level attends over different meta-paths [38]. Heterogeneous Graph Transformer (HGT) proposes node- and edge-type dependent attention mechanism to handle both node and edge heterogeneity in a graph followed by heterogeneous message passing mechanism and target-specific aggregation for feature updating [39]. These GNNs can handle heterogeneity in a graph but ignore the edge features. For more information about graph neural networks, please refer to recent review articles [40], [41].

Most existing graph neural networks handle heterogeneity and edge features separately and cannot be directly used to model the interaction represented by a directed edge-featured heterogeneous graph. In this work, we extend GAT [33] to handle both heterogeneity and edge features for interaction modeling in multi-agent trajectory prediction.

C. Trajectory Prediction With GNNs

Graph-based interaction representation has attracted more and more interests in the field of trajectory prediction, which gives rise to the application of graph neural networks. Authors of [20] test two widely used GNNs (GCN [30] and GAT [33]) and their adaptions on the trajectory prediction task and find that adaptions of GNNs, which discern between the target and surrounding agents, outperforms the GNNs that treat them without distinction. They conceptually prove the effectiveness of graph-based interaction representation but the agents' dynamics are not considered in their work. GRIP proposes a graph convolutional model, which comprises convolutional and graph operation layers alternatively, to summarize the temporal and spatial features of interactive agents [14]. The convolutional layer is applied to the temporal dimension and the graph operation is applied to spatial relationships, and then an LSTM-based encoder-decoder is used for final prediction. GRIP can predict trajectories of multiple agents but it ignores edge attributes. SCALE-Net constructs edge attributes with

the relative measurements between two agents and employs Edge-enhanced Graph Convolutional Neural Network [34] to summarize interactions considering edge attributes [23]. One common issue of the above-mentioned models is that they ignore the heterogeneity of traffic participants. Social-WaGDAT designs Wasserstein Graph Double-Attention Network to learn the structure of the interaction graph dynamically and applies kinematic constraints on the predicted trajectory [16]. VectorNet applies GNN to a fully-connected hierarchical graph, where a sub-graph contains the feature of an object (either an agent or map component) represented by a sequence of vectors [21]. Heterogeneity is considered in the constructed graph, but the fully-connected graph ignores the spatial structure of interaction and the number of edges increases exponentially with the number of nodes. TNT adopts VectorNet 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 [22]. It shares the drawbacks of VectorNet. ReCoG constructs a heterogeneous graph to represent agent-agent and agent-infrastructure relationships, where the infrastructure (a top-view map) is a node in the graph, and applies GAT and GCN to extract interaction features [15]. However, edge attribute and type are ignored in ReCoG.

Existing trajectory prediction methods are proposed for specific interaction representations and they can hardly be applied to new representations. In this work, we propose our three-channel framework for simultaneous multi-agent trajectory prediction along with our interaction representation and HEAT network.

III. Structure Overview

This section introduces the high-level structure of the proposed framework for heterogeneous multi-agent trajectory prediction and its key components, namely, agent-type-specific history encoder, HEAT-based heterogeneous interaction encoder, adaptive map selector, and agent-type-specific trajectory predictor. The proposed framework has three channels for dynamics, interaction, and map features, respectively, then jointly considers these features to predict future trajectories of heterogeneous agents. See Fig. 1 for an illustration of the three-channel framework.

Input and output. The task (see Fig. 2 for an illustration) of this work is to simultaneously predict multi-agent trajectories of a group of heterogeneous interactive agents considering their inter-agent interactions and the scene context (shown in the left of Fig. 2). At a time t, the input $\mathbf{X}_t$ contains each agent's historical states and the map of the scene (shown in left of Fig. 2).

$$\mathbf{X}_t = [\mathcal{H}_t, \mathcal{M}], \tag{1}$$

where $\mathcal{H}_t = \{h_t^1, h_t^2, \cdots, h_t^n\}$ contains the historical states of n agents at time t, $\mathcal{M}$ is the scene context. 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, for instance, can be agent i's position and velocity at t. The number of observed agents n is variable from case to case. The map $\mathcal{M}$ is to be shared by all the agents. The output contains predicted trajectories of $m \leq n$ heterogeneous agents (shown in right of Fig. 2):

$$\mathcal{F}_t = \{f_t^1, f_t^2, \cdots, f_t^m\}, \tag{2}$$

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, $\mathcal{F}_t$ is the set of predicted trajectories of m agents. Please note that the number of target agents m is not necessary to be equal to n and can vary from case to case. This is a typical situation in which an autonomous vehicle would need to predict trajectories of the agents it is interacting with (the target agents) considering other non-target agents' information.

Agent-type-specific history encoder. To handle the heterogeneity of traffic participants, we propose to share a history encoder over a specific type of traffic participants. In this work, we assume that there are two types of traffic participants (i.e. vehicle and pedestrian/bicyclist), such that there will be two type-specific history encoders, one for each (see Fig. 1). History encoders are applied to individual agents' historical states to extract their dynamics features. The dynamics features are also used in the interaction channel as node features.

HEAT-based heterogeneous interaction encoder. In this work, we represent the interaction among heterogeneous traffic participants with a directed edge-featured heterogeneous graph (see the middle of Fig. 2 for an illustration and Sub.Sec. IV-B for details) and propose a novel heterogeneous edge-enhanced graph attention network (HEAT) to extract interaction features (see Fig. 1). Nodes in the graph contain dynamics features of corresponding agents out from their history encoders.

Adaptive map selector. We design a CNN to extract road feature from a bird's eye view map of the driving scene and selectively share the map feature across all target agents according to their current positions, velocities, and yaw angles, by introducing a gate-mechanism (see Fig. 1).

Agent-type-specific trajectory predictor. Similar to the history encoder, a trajectory predictor is shared over a specific type of target agents. The target agents in this work also fall into two categories (i.e. vehicle and pedestrian/bicyclist). To simultaneously predict trajectories, the predictor jointly considers the target vehicles' dynamic features extracted from the history encoder, their interaction features obtained from the interaction encoder, and their corresponding map features received from the map selector. Please note that the input features of the trajectory predictors are hidden features (represented by high-dimensional vectors) from neural networks. See Fig. 1.

IV. Method

This section first provides the architecture of the proposed multi-agent trajectory prediction framework (IV-A), then elaborates on the proposed interaction representation (IV-B), the proposed heterogeneous edge-enhanced graph attention network: HEAT (IV-C), and gated map selector (IV-D) for the framework.

A. Heterogeneous Multi-Agent Trajectory Prediction Scheme

The framework shown in Fig. 1 is proposed for multi-agent trajectory prediction, where there are two types of agents,

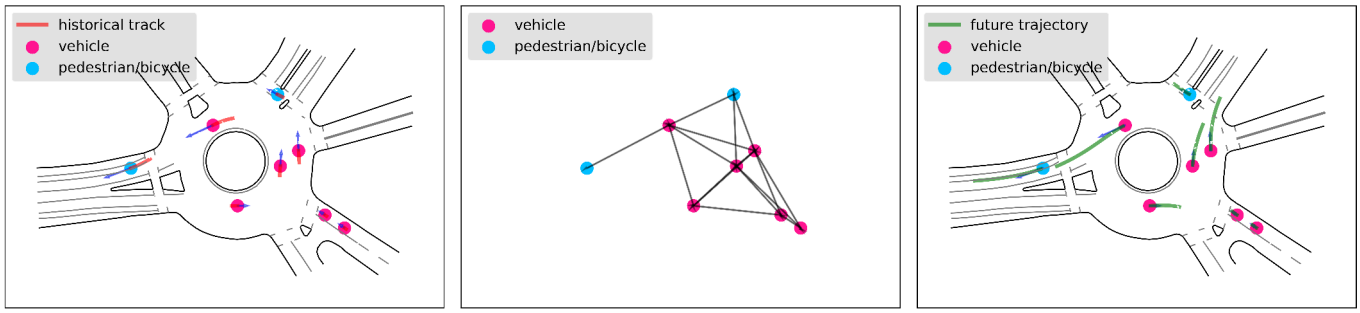


Fig. 2. **Input, graph, and output.** *Left*, the one-second historical tracks of multiple agents of different types navigating in a roundabout scene. *Middle*, The structure of the constructed directed heterogeneous graph with neighboring connections. Self-loop is masked out for clarity. *Right*, the three-second future trajectories of multiple heterogeneous agents in the scene. The pink and blue dots show the current positions of vehicle and pedestrian/bicyclist agents, respectively. The red solid lines in the left figure are the historical trajectories of the agents over the last one second. The green solid lines in the right figure are the corresponding future trajectories in three seconds. This figure is sampled from a roundabout scenario named *DR_USA_Roundabout_FT* in the INTERACTION dataset.

leveraging both historical states of agents and the infrastructure information. To handle the heterogeneity of agents, we design specific encoders (IV-A1) and decoders (IV-A4) for each type of agent. Considered agents are placed in their own exclusive coordinate system and their interactions are represented by a directed edge-featured heterogeneous graph (IV-B). A novel heterogeneous edge-enhanced graph attention network is proposed to extract interaction features from the constructed graph (IV-A2). To utilize the road structure and share it across all considered agents, we propose an adaptive map selector (IV-A3).

1) Agent-Type-Specific History Encoder: For an agent of type $\kappa, \kappa \in \{vehicle, pedestrian/bicyclist\}$, its historical states h_t^i is represented by a temporal sequence that can be passed to a type-specific encoder to extract its dynamics feature. RNNs, e.g., Long short-term memory (LSTM) and gated recurrent unit (GRU), are widely used for sequence modeling in machine translation [42], [43] and trajectory prediction [13], [15]. We adopt GRUs as history encoders in this work (Eq. 3) because of its effectiveness and simplicity:

$$r_t^i = \mathrm{GRU}_{\mathrm{hist}}^{\kappa}(h_t^i), \tag{3}$$

where $\mathrm{GRU}_{\mathrm{hist}}^{\kappa}$ is the historical encoder of agent type κ implemented using GRU and r_t^i is the dynamics feature of vehicle i at time t. The output of this module is the dynamics features of all the agents:

$$R_t = \{r_t^1, r_t^2, \cdots, r_t^n\}, \tag{4}$$

where the dynamics features R_t also serve as the node features in the graph-based interaction representation.

2) Heterogeneous Interaction Modeling With HEAT: To comprehensively model the inter-agent interaction among heterogeneous agents, we represent the interaction as a directed edge-featured heterogeneous graph and propose a novel Heterogeneous Edge-enhanced graph ATtention network (HEAT) to extract interaction features from the graph representation. Details of the interaction representation and the proposed HEAT can be found in Sub.Sec. IV-B and Sub.Sec. IV-C, respectively.

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.

$$G_t = \{g_t^0, g_t^1, \cdots, g_t^n\} = \mathrm{HEAT}_{\mathrm{enc}}(R_t, E_t), \tag{5}$$

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.

3) Map Selection With Gate Mechanism: Road structure shapes the motion of agents navigating within an urban scene, so it is necessary to take into consideration the road structure for trajectory prediction. Previous single trajectory prediction methods use a fixed-size local map centered at the target vehicle's current position. But for simultaneous multi-agent trajectory prediction, this map representation has at least two drawbacks: 1) It needs to save multiple maps for multiple agents; 2) The fixed-size map can be either too large for a slow agent or too small for a fast agent. To handle the above-mentioned drawbacks, we propose an adaptive map selection method that allows sharing a global map across all the agents according to their current positions, velocities, and yaw angles:

$$m_t^i = \mathrm{Selector}_{\mathrm{map}}\left(\mathcal{M}, (x_t^i, y_t^i, v_{x_t}^i, v_{y_t}^i, \phi_t^i)\right), \tag{6}$$

where $\mathcal{M}$ is the global map and $(x_t^i, y_t^i, v_{x_t}^i, v_{y_t}^i, \phi_t^i)$ is the current position, velocity, and yaw angle of agent i in the map. The selector allows the method to selectively consider the global map and focus on the most relevant areas. The selected map feature of agent i is conditioned on its states.

4) Agent-Type-Specific Future Decoder: For an agent of type $\kappa, \kappa \in \{vehicle, pedestrian/bicyclist\}$, its future trajectory is predicted using an agent-type-specific future trajectory decoder by jointly considering its individual dynamics r_t^i, its interaction with other agents g_t^i, and the map feature regarding its current states m_t^i.

$$f_t^i = \mathrm{LSTM}_{\mathrm{fut}}^{\kappa}([r_t^i \| g_t^i \| m_t^i]), \tag{7}$$

where $\mathrm{LSTM}_{\mathrm{fut}}^{\kappa}$ is the future decoder shared across agents of type κ, $[r_t^i \| g_t^i \| m_t^i]$ is the concatenation of features, and f_t^i is the predicted future trajectory of agent i.

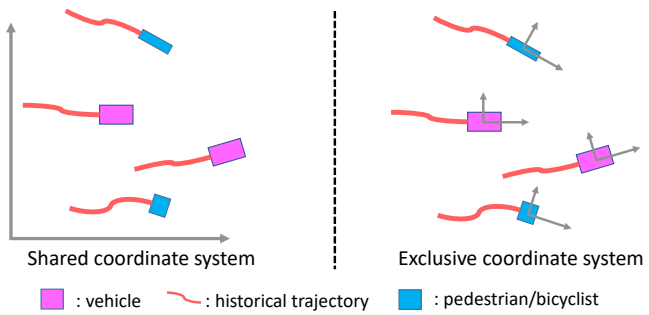


Fig. 3. **Shared and exclusive coordinate systems.** *Left*, Heterogeneous agents in a shared coordinate system. *Right*, Heterogeneous agents with exclusive coordinate systems. An agent's exclusive coordinate system is with its origin fixed at the vehicle's current position and its horizontal axis pointing to the vehicle's moving direction.

B. Interaction Representation With Directed Edge-featured Heterogeneous Graph.

In this work, we place all agents in their own exclusive coordinate systems and represent their interaction as a directed edge-featured heterogeneous graph.

1) Exclusive Coordinate System: Most existing interaction-aware trajectory prediction methods use either a shared coordinate system for all agents or an exclusive coordinate system for each to represent trajectories. A shared coordinate system preserves the spatial relationship among agents, however, it is sensitive to translation and rotation. The input to the model becomes totally different when a new shared coordinate system is applied. But in the case of the exclusive coordinate system, agents' states are represented locally and independent of other agents. The localized exclusive coordinate system standardises the states of agents but omits spatial relationships among agents, which should be reserved with edge features to take advantage of the exclusive coordinate system.

2) Graph Represented Interaction: In this work, we propose to represent inter-agent interaction as a directed edge-featured heterogeneous graph for multi-agent trajectory 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. 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, and yaw angle. The edge type is a concatenation of the type of node j and node i. For details of the edge construction in this work, please refer to Sub.Sec. V-A, and the code will be released soon.

Definition 1 (Directed Edge-Featured Heterogeneous Graph)**.** *A directed edge-featured heterogeneous graph can be represented by $\mathbb{G} = (V, E)$, where $V = \{v_1, \cdots, v_n\}$ is the set of n nodes, and $E \subset V \times V$ is the set of directed edges. Each node contains its node feature and belongs to a specific type. Each directed edge contains an edge attribute and falls into a specific edge type.*

Compared to previous works that represent interaction as a homogeneous graph [20], edge-featured homogeneous graph [23], or heterogeneous graph without edge attributes [24], the proposed representation is more comprehensive. It covers the heterogeneity of traffic participants with heterogeneous nodes; preserves their individuality with exclusive coordinate systems; considers the difference of the mutual influence between two agents with directed edges; maintains the spatial relationship of all the agents using edge attributes.

Transformation-insensitive interaction graph. To model the interaction among agents of different types, we construct a directed heterogeneous graph to represent 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:

$$E = \{e_{ij}\}_{(j \in \mathcal{N}_i)}, \quad i = 1, \cdots, n, \tag{8}$$

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. An example of the constructed graph is shown in middle of Fig. 2

C. HEAT Layer

The above-mentioned interaction representation should be treated with a graph neural network that can handle the heterogeneity of nodes, directed edges, and continuous edge attributes. However, as shown in related works, existing GNNs do not handle this in one fell swoop. In this work, we design a heterogeneous edge-featured graph attention network (HEAT), an extension of GAT, for the proposed comprehensive interaction representation. 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.

1) Input and Output: The input to the HEAT layer contains a set of node features: $\mathbf{h} = \{\vec{h}_i | i \in [1, n]\}$, where $\vec{h}_i \in \mathbb{R}^{F_h}$ is the feature vector of node i; and a set of edge attributes: $\mathbf{e}^{attr} = \{e_{ij}^{attr} | i, j \in [1, n]\}$, where $e_{ij}^{attr} \in \mathbb{R}^{F_e^{attr}}$ is the attribute of the edge pointing from node j to node i. A set of edge types is represented as $\mathbf{e}^{type} = \{e_{ij}^{type} | i, j \in [1, n]\}$, where $e_{ij}^{type} \in \mathbb{R}^{F_e^{type}}$ is the type of the edge pointing from node j to node i. The output of the HEAT layer is a new set of node features: $\mathbf{h}' = \{\vec{h}_i' | i \in [1, n]\}, \vec{h}_i' \in \mathbb{R}^{F_h'}$.

2) Heterogeneous Transformation: Different kinds of nodes in a heterogeneous graph have different feature spaces and should be projected to a shared feature space. We adopt the node-type-specific transformation matrix $\mathbf{M}_{\kappa i} (\kappa \in \{vehicle, pedestrian/bicyclist\})$ introduced in [38] to handle two kinds of nodes in this work. The transformation can be expressed as $\vec{h}_i = \mathbf{M}_{\kappa i} \cdot \vec{h}_i$. Please note that we will simply re-use symbols including $\vec{h}_i, \mathbf{e}^{attr}, \mathbf{e}^{type}$ to indicate the transformed feature in the attention (IV-C3) and aggregation (IV-C4) parts.

Existing works either consider edge attributes or edge types as edge features, whereas this is not the case for multi-agent trajectory prediction. We argue that, for trajectory prediction,

edge feature and type are two different attributes. The edge features are usually some measurements in a continuous space, such as the distance between two nodes. However, the edge type is always a discrete indicator. Thus, we separately consider the edge features and types by introducing the edge attribute transformation: $\mathbf{e}^{attr} = \mathbf{M}_\phi \cdot \mathbf{e}^{attr}$, where $\mathbf{M}_\phi$ is the edge attribute transformation matrix, and the edge type transformation is: $\mathbf{e}^{type} = \mathbf{M}_\chi \cdot \mathbf{e}^{type}$, where $\mathbf{M}_\chi$ is the edge type transformation matrix.

3) Edge-Enhanced Masked Attention: For an edge pointing from node j to node i, its edge feature: $e_{ij} = [e_{ij}^{attr} \| e_{ij}^{type}]$, 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 [26], which is a single-layer feed-forward neural network, $\vec{\mathbf{a}}$, followed by 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. GAT layer performs masked attention, which attends over the neighborhood of node i only, to utilize the structural information of the graph while casting away the edges' feature and type [33]. In this work, an edge-enhanced masked attention in Eq. 9 is performed to fully consider the graph attributes:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakeyReLU}\left(\vec{\mathbf{a}}^T[\vec{h}_i \| e_{ij}^{+}]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\text{LeakeyReLU}\left(\vec{\mathbf{a}}^T[\vec{h}_i \| e_{ik}^{+}]\right)\right)}, \tag{9}$$

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

4) Node Feature Aggregation: 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:

$$\vec{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{W}_h[e_{ij}^{attr} \| \vec{h}_j]\right). \tag{10}$$

Edge types are not included in the edge-integrated feature since it is discrete and already considered in the previous attention mechanism (Eq. 9). Similar to GAT [33], HEAT allows running of several independent attention mechanisms to stabilize the self-attention mechanism.

The elements of the proposed HEAT layer are listed in Tab. I for convenience.

TABLE I
NOTATIONS OF HEAT LAYER

$\vec{h}_i$	Feature vector of node i
$\mathbf{h}$	The set of node features in a graph
e_{ij}^{attr}	attribute of edge from j to i
$\mathbf{e}^{attr}$	The set of edge attributes in a graph
e_{ij}^{type}	type of edge from j to i
$\mathbf{e}^{type}$	The set of edge types in a graph
e_{ij}	concatenation of projected attribute and type
e_{ij}^{+}	concatenation of e_{ij} and $\vec{h}_j$
α_{ij}	node j's attention coefficient for node i
$\mathcal{N}_i$	Neighborhood of node i
$\|$	concatenation
$\vec{\mathbf{a}}^T$	Attention mechanism
σ	Sigmoid function
$\mathbf{h}'$	The set of updated node features in a graph

D. Gated Map Selection

The road structure highly affects the motions of traffic participants, so trajectory prediction cannot ignore this information. Single-agent trajectory prediction methods, such as ReCoG [15], use a fixed-size local map centered at the target vehicle's current position. However, a fixed-size local map ignores the dynamics of agents. A small map is enough to predict a slow agent, while a larger map is needed for a fast-moving agent. Multi-agent prediction methods, such as MATF [18], share the same map feature across all target vehicles ignoring the fact that different target agents are affected by different parts of the map. To enable selective map sharing, we propose to apply the gate mechanism on the CNN-extracted map feature for map selection, which has been widely used in sequence modeling [44]–[46]. For example, LSTM has three gates (input gate, forget gate, and output gate) to manage the information flows along the sequence data [46]. The input gate is designed to select what information of the current step to be added to the memory of the network. In this work we design the selection gate z_t^i to select map feature for an agent i according to its current state in the map:

$$z_t^i = \sigma(W_z[\vec{\mathcal{M}} \| s_t^i] + b_z), \tag{11}$$

where $\vec{\mathcal{M}}$ is the map $\mathcal{M}$'s feature vector extracted using a CNN, s_t^i is agent i's current state in the map's coordinates system, W_z is a projection weight matrix, b_z is a bias, σ is a Sigmoid function. Thus, z_t^i is a vector with each element contains a number between 0 and 1. Then the map feature of agent i at time t is selected with this gate:

$$m_t^i = z_t^i \circ \vec{\mathcal{M}}, \tag{12}$$

where $\circ$ is element-wise production and m_t^i is the selected map feature.

V. REAL-WORLD DATASET VALIDATION

This section first compares the proposed trajectory prediction method with state-of-the-art models on the recently published INTERACTION dataset [47] for urban driving, then on the NGSIM US-101 [48] dataset for highway driving. The INTERACTION dataset is provided by the Mechanical Systems Control (MSC) Lab of University of California, Berkeley, the Centre for Robotics of MINES ParisTech, and the Institute for Measurement and Control Technology (MRT) of FZI Research Center for Information Technology and Karlsruhe Institute of Technology. The NGSIM dataset is provided by the U.S. Federal Highway Administration.

A. Validation On Heterogeneous Dataset

The proposed heterogeneous multi-agent trajectory prediction method is trained and validated on the INTERACTION [47]. The full name of the dataset is INTERnational,

Adversarial and Cooperative moTION Dataset. It contains naturalistic trajectories of different traffic participants, e.g., vehicles and pedestrians, in highly interactive urban scenarios world-wide. The recorded scenarios fall into three categories: roundabout, intersection, and merging.

1) Heterogeneous Dataset: INTERACTION dataset provides states of agents at each timestamp along with a high definition (HD) map. The state of a vehicle at a timestamp includes its position, velocity, yaw angle, and shape, while the state of a pedestrian/bicyclist includes only its position, velocity, and yaw angle. Since this work aims at simultaneously predicting trajectories of multiple heterogeneous agents and proposes to represent inter-agent interaction as a heterogeneous directed-edge-featured graph, the raw dataset is processed accordingly. Please see Fig. 4 for an illustration of the processed data and the appendix for a detailed description. The processed dataset is split to train and validation set following the split suggested by the authors of the INTERACTION dataset. The processed dataset contains $425,192$ data pieces for training and $104,627$ for validation.

2) Comparison With State-of-the-art Methods: In this work, we evaluate prediction performance using average displacement error (ADE) and final displacement error (FDE) in meters adopted by previous works [15], [22]. The proposed method is compared with the following methods on the INTERACTION dataset.

- DESIRE: DESIRE predicts multi-modal trajectories by jointly considering motion history, static scene context and inter-agent interactions. It first generates diverse hypothetical future prediction samples using a conditional variational auto-encoder, then ranks and refines the samples in an inverse optimal control framework with regression [49].
- MultiPath: MultiPath handles driving uncertainty by hierarchically model intent and control uncertainties. It first produces a set of K anchor trajectories as intents, then predicts future trajectories conditioned on anchors, where the uncertainty is modeled as a Gaussian distribution given an intent [50].
- TNT: TNT utilizes VectorNet [21] to encode the target agent's interaction with surrounding agents and the environment. It first predicts an agent's target states within a prediction horizon, then generates trajectories for each target. Finally a set of predictions is selected according to estimated likelihoods for multi-modality [22].
- ReCoG: ReCoG represents vehicle-vehicle and vehicle-infrastructure interactions as a heterogeneous graph and applies state-of-the-art GNNs for interaction encoding. It predicts a single trajectory for a single target vehicle [15].

Tab. II compares the proposed three-channel model HEAT-I-R with existing methods. It shows that: 1) The proposed HEAT-I-R outperforms DESIRE [49] and MultiPath [50], even though these two methods predict multi-modal (MM) trajectories for a single agent and the ADE and FDE are reported with the minimum values among the multiple predictions [22]; 2) HEAT-I-R matches the performance of TNT [22] and ReCoG [17]. Please note that TNT [22] predicts six-modal trajectories for a single agent and reports the minimum ADE and FDE over all predictions and ReCoG [15], the winner solution of the INTERPRET Challenge (NeurIPS 2020) [51], predicts a single trajectory for a single target. Compared to TNT [22] and ReCoG [15], the proposed HEAT-I-R is able to predict trajectories of multiple agents (MA) simultaneously. The inference time satisfies the real-time requirements. It uses 0.06 seconds to predict trajectories of a batch of 128 data pieces including hundreds of target agents.

TABLE II
COMPARISON WITH STATE-OF-THE-ART METHODS ON THE INTERACTION DATASET

Methods	MM	ADE@3sec (m)	FDE@3sec (m)
DESIRE [49]	✓	0.32 ($\mathbf{min}_6$)	0.88 ($\mathbf{min}_6$)
MultiPath [50]	✓	0.30 ($\mathbf{min}_6$)	0.99 ($\mathbf{min}_6$)
TNT [22]	✓	0.21 ($\mathbf{min}_6$)	0.67 ($\mathbf{min}_6$)
ReCoG [15]		**0.19**	**0.65**
HEAT-I-R (Ours)		**0.19**	0.66

TABLE III
ABLATIVE COMPARISON ON THE INTERACTION DATASET'S DR_USA_ROUNDABOUT_FT SCENARIO

Methods	ADE@8sec (m)	FDE@8sec (m)
R	3.99	11.64
GAT	3.98	11.59
GAT-R	3.5	10.62
HEAT	3.10	8.83
HEAT-R	3.09	8.84
HEAT-I-R	**2.97**	**8.56**

3) Ablative Study: In this work, we conduct ablative studies on the INTERACTION dataset's DR_USA_Roundabout_FT scenario for a longer prediction horizon (eight seconds). The eight-second horizon is selected to challenge our method, since a longer-term prediction is intrinsically more difficult than a short-term one [52]. The following settings are trained and validated on the same dataset.

- R: One-channel model considering the target vehicle's RNN-encoded dynamics feature only for future trajectory prediction.
- GAT: One-channel model predicting the future trajectory of the target vehicle considering its graph-modeled interaction feature extracted using GAT.
- GAT-R: Two-channel model jointly considering the GAT-extracted interaction and RNN-encoded dynamics features for trajectory prediction.
- HEAT: One-channel model predicting the future trajectory of the target vehicle considering its graph-modeled interaction feature extracted using HEAT.
- HEAT-R: Two-channel model jointly considering the HEAT-extracted interaction and RNN-encoded dynamics features for trajectory prediction.
- HEAT-I-R: The proposed three-channel framework, which combines the target vehicle's individual dynamics feature, its interaction feature, and the selected map feature for trajectory prediction.

Tab. III shows the ADE@8sec and FDE@8sec of above listed implementations. It is observed that the proposed three-channel framework (HEAT-I-R) outperforms its two-channel

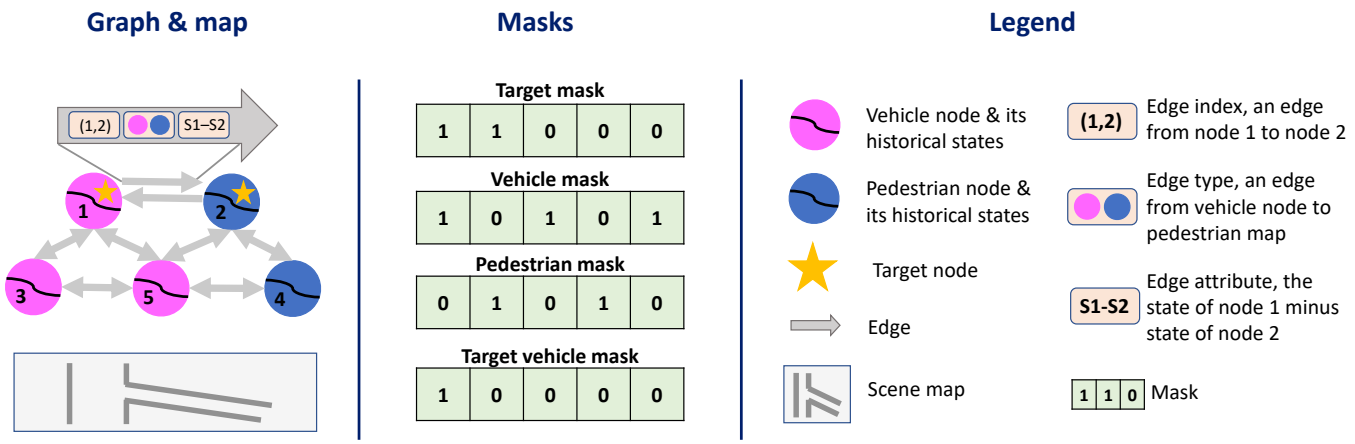


Fig. 4. **Processed data.** *Left*, the directed edge-featured heterogeneous interaction graph (top) obtained after processing and the map (bottom). *Middle*, the mask vectors. *Right*, the legend of this figure. The interaction graph is constructed via close connection strategy, where each node is only connected to its neighboring nodes via directed edges. A directed edge is identified via its edge index and contains edge type and edge attribute.

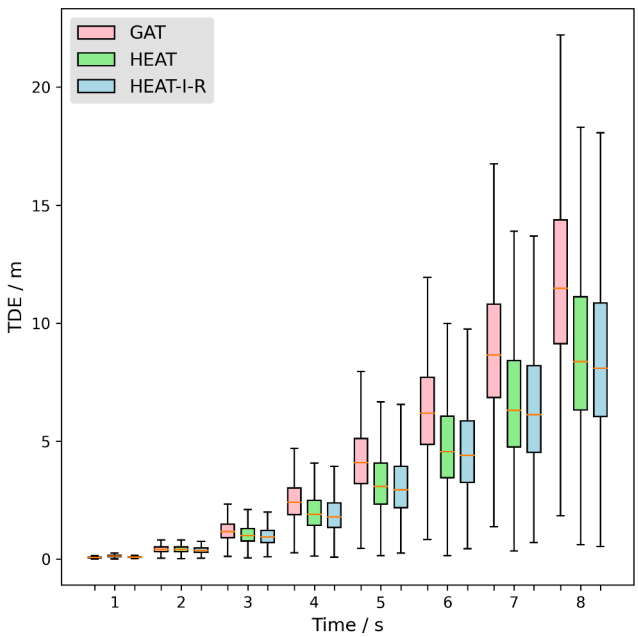


Fig. 5. **Box plots of the TDE of ablative models.** GAT, HEAT, and HEAT-I-R are selected for clarity of the box plots.

(HEAT-R) and one-channel (HEAT) ablations, which supports the intuition that individual dynamics, inter-agent interactions, and road information all benefit trajectory prediction. It is also noticed that one-channel GAT-based method (GAT in Tab. III) performs as poorly as the non-interaction-aware method (R in Tab. III.), but their combination (GAT-R) improves prediction accuracy. GAT-based methods are not suitable for trajectory prediction with exclusive coordinate systems, where the spatial relationships of agents are stored in edge features, because GATs ignore edge features.

Fig. 5 shows box plots of the TDE (displacement error over time) of three ablatives models (GAT, HEAT, and HEAT-I-R) over an eight-second prediction horizon. The red boxes show the results of GAT, the green boxes the results of HEAT, and the blue boxes the result of the proposed HEAT-I-R. Outliers and the results of other ablative models are not plotted for clarity. It can be seen that the proposed HEAT-based model shows more stable performance (with shorter interquartile range (IQR)) than the GAT-based model, and the proposed three-channel framework (HEAT-I-R) further improves accuracy and stability.

B. Validation On Homogeneous Dataset

The proposed HEAT is an immediate extension to GAT [33] while the GAT is designed for homogeneous graphs. To compare the proposed HEAT with GAT on the single-agent trajectory prediction task, we construct two homogeneous datasets using vehicle trajectories provided by the public accessible NGSIM US-101 dataset [48]. The trajectories in NGSIM US-101 dataset are recorded from a segment of U.S. Highway 101 at 10 Hz. Since most of the trajectories did not change lanes throughout the study area, we reprocess the dataset to build a roughly balanced dataset where lane-keeping trajectories do not dominate the dataset. The dataset is constructed by first selecting target vehicles that have changed their lanes only once during recording, then selecting trajectory segments for the target vehicle and its neighboring vehicles. The data processing procedure is similar to that in CNN-LSTM [17], a previous work focusing on the NGSIM dataset, except that an arbitrary number of neighboring vehicles is allowed in this work. After above processing, totally $63,176$ data pieces are selected for training $(53,176)$ and validation $(10,000)$. The selected data is further processed to formulate two different datasets, one with exclusive coordinate system, and the other one with shared coordinate system. In the former dataset, each agent is placed in its own coordinate system, in which the agent's current position and yaw angle are zeros, while in the latter one, all the agents share the coordinate system whose origin is fixed at the current position of the target vehicle and horizontal axis points to the direction of the target vehicle's current velocity. The former dataset has edge attributes containing the relative position of the agent in the source node to the agent in the target node, while the latter one

does not have edge attributes because of the shared stationary frame of reference.

Similar to the ablative study on the heterogeneous dataset, we implement five models, namely R, GAT, GAT-R, HEAT, and HEAT-R, to show the the effectiveness of the proposed HEAT layer. Descriptions of these models can be fund in V-A3. All these models are trained and validated on the dataset with the exclusive coordinate system, and the GAT-based baselines are further implemented in the dataset with shared coordinate system.

We compare the proposed model with state-of-the-art methods and report the results in Tab. IV. The prediction results in this section are evaluated using root-mean-square error (RMSE) in meters to compare with the existing works [13], [17], [23].

Tab. IV compares the proposed method with existing works. It shows that the proposed HEAT-R method outperforms existing models at longer prediction horizons (3-5 sec) and matches the state-of-the-art methods in short-term prediction (1-2 sec). It can be seen that the accuracy of the proposed HEAT-R is quite close to that of CNN-LSTM [17] which uses a share coordinate system and accepts exactly eight close neighboring vehicles. However, HEAT-R is able to handle an arbitrary number of neighboring vehicles and use the exclusive coordinate system that standardises the input sequence and narrows down the search space for the input encoder.

Tab. V compares HEAT with GAT-based and RNN-based methods, where the first five rows (#1-5) show the results on the dataset with the exclusive coordinate system, and the last two rows (#6-7) show the results of GAT-based models on the dataset with the shared coordinate system. It can be seen that the graph-based interaction-aware methods (#2-7) outperform the non-interaction-aware RNN-based one (#1). This observation is consistent with previous works [13], [17], [18], which shows again the necessity to model interaction for trajectory prediction. The GAT-based methods show better performance on the second dataset (shared, #6,7) compared to the results on the first dataset (exclusive, #2,3). This is quite reasonable in that GAT ignores the spatial relationship among agents contained in the edge features in the first dataset, while the spatial relationship is preserved by the shared coordinate system in the second dataset. The proposed HEAT-based methods (#4-5) for the dataset with the exclusive coordinate system outperform all GAT-based methods. This shows the advantage of using the exclusive coordinate system and applying HEAT for interaction extraction.

It can also be found that the results reported in this section seem poorer than that of the previous section. But we avoid comparing methods across datasets for the following reasons: 1) in this section, the main model HEAT-R does not consider road structure, since the highways are usually relatively straight; 2) the average speed of highway driving is usually higher than that of the urban driving, which makes the prediction task more challenging; 3) the proposed method is designed for heterogeneous multi-agent prediction, so it is more suitable for multiple agents' prediction.

TABLE IV
PREDICTION PERFORMANCE COMPARISON WITH EXISTING WORKS (RMSE IN METERS)

	Methods	Prediction horizon				
		1 sec	*2 sec*	*3 sec*	*4 sec*	*5 sec*
1	HEAT-R (Ours)	0.68	0.92	**1.15**	**1.45**	**2.05**
2	CS-LSTM [13]	0.61	1.27	2.09	3.10	4.37
3	GRIP [14]	**0.37**	**0.86**	1.45	2.21	3.16
4	CNN-LSTM [17]	0.64	0.96	1.22	1.53	2.09
5	CS-LSTM(M) [13]	0.62	1.29	2.13	3.20	4.52
6	GAIL-GRU [53]	0.69	1.51	2.55	3.65	4.71
7	Scale-Net [23]	0.46	1.16	1.97	2.91	-
8	MATF-GAN [18]	0.66	1.34	2.08	2.97	4.13

TABLE V
PREDICTION PERFORMANCE COMPARISON OVER ABLATIVE IMPLEMENTATIONS (RMSE IN METERS)

	Methods	Prediction horizon				
		1 sec	*2 sec*	*3 sec*	*4 sec*	*5 sec*
1	R-1	0.6931	1.7275	3.0850	4.7735	6.7855
2	GAT-1	0.7808	1.4916	2.3914	3.4981	4.8713
3	GAT-R-1	0.8228	1.6987	2.7385	3.9672	5.4129
4	HEAT-1	0.6590	0.8556	1.0469	1.3216	1.8894
5	HEAT-R-1	0.6794	0.9212	1.1528	1.4457,	2.0518
6	GAT-2	0.7685	1.2115	1.7343	2.4533	3.5466
7	GAT-R-2	0.6439	0.9592	1.2643	1.6489	2.3124

C. Implications And Limitations

It can be seen from the validation results that the proposed method can simultaneously predict multiple heterogeneous objects' trajectories with state-of-the-art performance. The assumption and concept of this method are in line with the real-world implementation scenarios, where autonomous vehicles interact with different types of other road users and need to predict their future motions for better decision-making. The prediction results can be used by the downstream decision-making and planning modules to improve safety and efficiency. In addition, not limited to autonomous driving, this method can be expanded and applied to many other robotic application domains.

Despite the performance and scalability, the proposed method still has limitations. Currently, it can only make deterministic predictions, while the motion of moving agents would have inherent multi-modality. This will be our future work for further exploration. In this work, we assume that the input data is ready to use, but the quality and availability of the data will be affected by the hardware settings in real-world applications, and the integration of the proposed method with other sub-modules and algorithms still needs to be configured in vehicle implementations.

VI. CONCLUSION

In this work, we propose a three-channel framework for simultaneous heterogeneous multi-agent trajectory prediction. We represent the inter-agent interaction in traffic with a directed edge-featured heterogeneous graph, design a novel heterogeneous edge-enhanced graph attention network for inter-agent interaction modeling, and introduce a gate mechanism for selective map sharing across all target agents. Validations

on both urban (INTERACTION) and highway (NGSIM) driving datasets show that the proposed method achieves state-of-the-art performance while is able to simultaneously predict multi-agent trajectories of an arbitrary number of heterogeneous agents.

For future works, one promising direction is to handle the multi-modality of traffic participants' behaviors by introducing multi-modal prediction. This will largely reduce the minimum ADE. Another direction is to incorporate rich infrastructure information, such as traffic lights, into our framework, to enhance the prediction accuracy.

References

[1] B. Shuai, Q. Zhou, J. Li, Y. He, Z. Li, H. Williams, H. Xu, and S. Shuai, "Heuristic action execution for energy efficient charge-sustaining control of connected hybrid vehicles with model-free double q-learning," *Applied Energy*, vol. 267, p. 114900, 2020.

[2] F. Farivar, M. S. Haghighi, A. Jolfaei, and S. Wen, "On the security of networked control systems in smart vehicle and its adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3824–3831, 2021.

[3] S. Ghane, A. Jolfaei, L. Kulik, K. Ramamohanarao, and D. Puthal, "Preserving privacy in the internet of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[4] M. Usman, M. A. Jan, and A. Jolfaei, "Speed: A deep learning assisted privacy-preserved framework for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] Q. Zhou, D. Zhao, B. Shuai, Y. Li, H. Williams, and H. Xu, "Knowledge implementation and transfer with an adaptive learning network for real-time power management of the plug-in hybrid vehicle," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[6] X. Song, K. Chen, X. Li, J. Sun, B. Hou, Y. Cui, B. Zhang, G. Xiong, and Z. Wang, "Pedestrian trajectory prediction based on deep convolutional lstm network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3285–3302, 2020.

[7] Y. Cai, L. Dai, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "Pedestrian motion trajectory prediction in intelligent driving from far shot first-person perspective video," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[8] Y. Wang, S. Zhao, R. Zhang, X. Cheng, and L. Yang, "Multi-vehicle collaborative learning for trajectory prediction with spatio-temporal tensor fusion," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[9] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[10] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, pp. 1–14, 2014.

[11] S. Ammoun and F. Nashashibi, "Real time trajectory prediction for collision risk estimation between vehicles," in *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*. IEEE, 2009, pp. 417–422.

[12] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 299–310, 2009.

[13] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1468–1476.

[14] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3960–3966.

[15] X. Mo, Y. Xing, and C. Lv, "Recog: A deep learning framework with heterogeneous graph for interaction-aware trajectory prediction," *arXiv preprint arXiv:2012.05032*, 2020.

[16] J. Li, H. Ma, Z. Zhang, and M. Tomizuka, "Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph double-attention network," *arXiv preprint arXiv:2002.06241*, 2020.

[17] X. Mo, Y. Xing, and C. Lv, "Interaction-aware trajectory prediction of connected vehicles using cnn-lstm networks," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2020, pp. 5057–5062.

[18] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. Baker, Y. Zhao, Y. Wang, and Y. N. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 126–12 134.

[19] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143–163, 2016.

[20] F. Diehl, T. Brunner, M. T. Le, and A. Knoll, "Graph neural networks for modelling traffic participant interaction," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 695–701.

[21] 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.

[22] 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.

[23] 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.

[24] J. Li, F. Yang, M. Tomizuka, and C. Choi, "Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning," *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

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

[27] X. Mo, Z. Chen, and H.-T. Zhang, "Effects of adding a reverse edge across a stem in a directed acyclic graph," *Automatica*, vol. 103, pp. 254–260, 2019.

[28] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[29] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *arXiv preprint arXiv:1606.09375*, 2016.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[31] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5115–5124.

[32] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035.

[33] 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

[34] L. Gong and Q. Cheng, "Exploiting edge features for graph neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9211–9219.

[35] X. Jiang, R. Zhu, S. Li, and P. Ji, "Co-embedding of nodes and edges with graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[36] Y. Yang and D. Li, "Nenn: Incorporate node and edge features in graph neural networks," in *Asian Conference on Machine Learning*. PMLR, 2020, pp. 593–608.

[37] J. Chen and H. Chen, "Edge-featured graph attention network," *arXiv preprint arXiv:2101.07671*, 2021.

[38] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[39] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of The Web Conference 2020*, 2020, pp. 2704–2710.

[40] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[41] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, 2020.

[42] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

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

[45] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[46] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[47] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088 [cs, eess]*, Sep. 2019.

[48] J. H. James Colyar, "Us highway 101 dataset," https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm, 2007.

[49] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.

[50] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.

[51] "Interaction-dataset-based prediction challenge," http://challenge.interaction-dataset.com/prediction-challenge/intro, 2020.

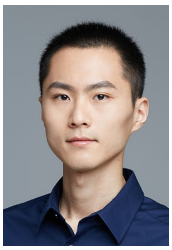
[52] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou *et al.*, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," *arXiv preprint arXiv:2104.10133*, 2021.

[53] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 204–211.

Xiaoyu Mo received the B.E. degree from Yangzhou University, Yangzhou, China, in 2015, the M.E. degree from Huazhong University of Science and Technology, Wuhan, China, in 2017. He was a research associate with Nanyang Technological University, Singapore, from Sep. 2017 to Jan. 2019. He is now a Ph.D. candidate at Nanyang Technological University. His research interests include trajectory prediction and decision making for connected autonomous vehicles.

Zhiyu Huang received his B.E. degree from the School of Automotive Engineering, Chongqing University, Chongqing, China, in 2019. He is currently pursuing his Ph.D. degree with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. His current research interests include deep reinforcement learning, motion prediction, and data-driven planning for automated driving.

Yang Xing received his Ph. D. degree from Cranfield University, UK, in 2018. He is currently a Lecturer in Applied Artificial Intelligence at Cranfield University. Before joining Cranfield, he was a Research Associate with the University of Oxford and a Research Fellow with Nanyang Technological University. His research interests include human behavior modeling, intelligent multi-agent collaboration, and intelligent/autonomous vehicles. He received the IV 2018 Best Workshop/Special Issue Paper Award. Dr. Xing serves as a Guest Editor for IEEE Internet of Thing Journal, IEEE Intelligent Transportation Systems Magazine, and Frontiers in Mechanical Engineering.



Chen Lv is currently an Assistant Professor at School of Mechanical and Aerospace Engineering, and the Cluster Director in Future Mobility Solutions at ERI@N, Nanyang Technology University, Singapore. He received the Ph.D. degree at the Department of Automotive Engineering, Tsinghua University, China in 2016. He was a joint PhD researcher at EECS Dept., University of California, Berkeley, USA during 2014-2015, and worked as a Research Fellow at Advanced Vehicle Engineering Center, Cranfield University, UK during 2016-2018. His research focuses on advanced vehicles and human-machine systems, where he has contributed over 100 papers and obtained 12 granted patents.

APPENDIX

A. Processed Data

A processed data are stored as:

- *Historical states*: The historical states within a 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. See Fig. 3 for the illustration of the exclusive coordinate system.
- Edge indexes: The graph connectivity represented as a set of directed edges. A directed edge from node j to node i means that agent j is within the neighborhood of agent i and affects the behavior of agent i. In this work, if agent j is within 30 meters to agent i, then it is treated as a neighbor 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. In this work, the relative states contain $(\Delta x, \Delta y, \Delta v_x, \Delta v_y, \Delta \psi)$.
- 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. In this work, the edge type of a directed edge from node j of type $[1, 0, 0]$ to node i of type $[0, 0, 1]$ is set to $[1, 0, 0, 0, 0, 1]$.
- 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's set to 0.
- Vehicle masks: The mask of of the vehicles in a scene with 1 represents a vehicle and 0 represents a non-vehicle.
- Pedestrian masks: The mask of the pedestrians in a scene with 1 represents a pedestrian and 0 represents a non-pedestrian.
- Target vehicle masks: The mask of the vehicles to be predicted with 1 means that the vehicle's future trajectories is to be predicted.
- Scene map: The map of the scene represented by a top-view image. Since we propose a learned map selector to share the map across all agents, the map can be stored outside each piece of data for just once. That saves a great amount of disk space. A corresponding map is saved for each of the eleven scenarios in the INTERACTION dataset.Examples of the scenario maps, please refer to Fig. 6. The image map of a scenario is shared across all the agents in this scenario via the designed map selector.
- Vehicle-to-map attributes: The states of all agents relative to the map's center at the current time t.
- Ground truth future trajectories: The recorded future trajectories of all target agents over the prediction horizon.

B. Visualization

Prediction results of the proposed framework with the proposed framework on several scenarios in the INTERACTION dataset are shown in Fig. 6.

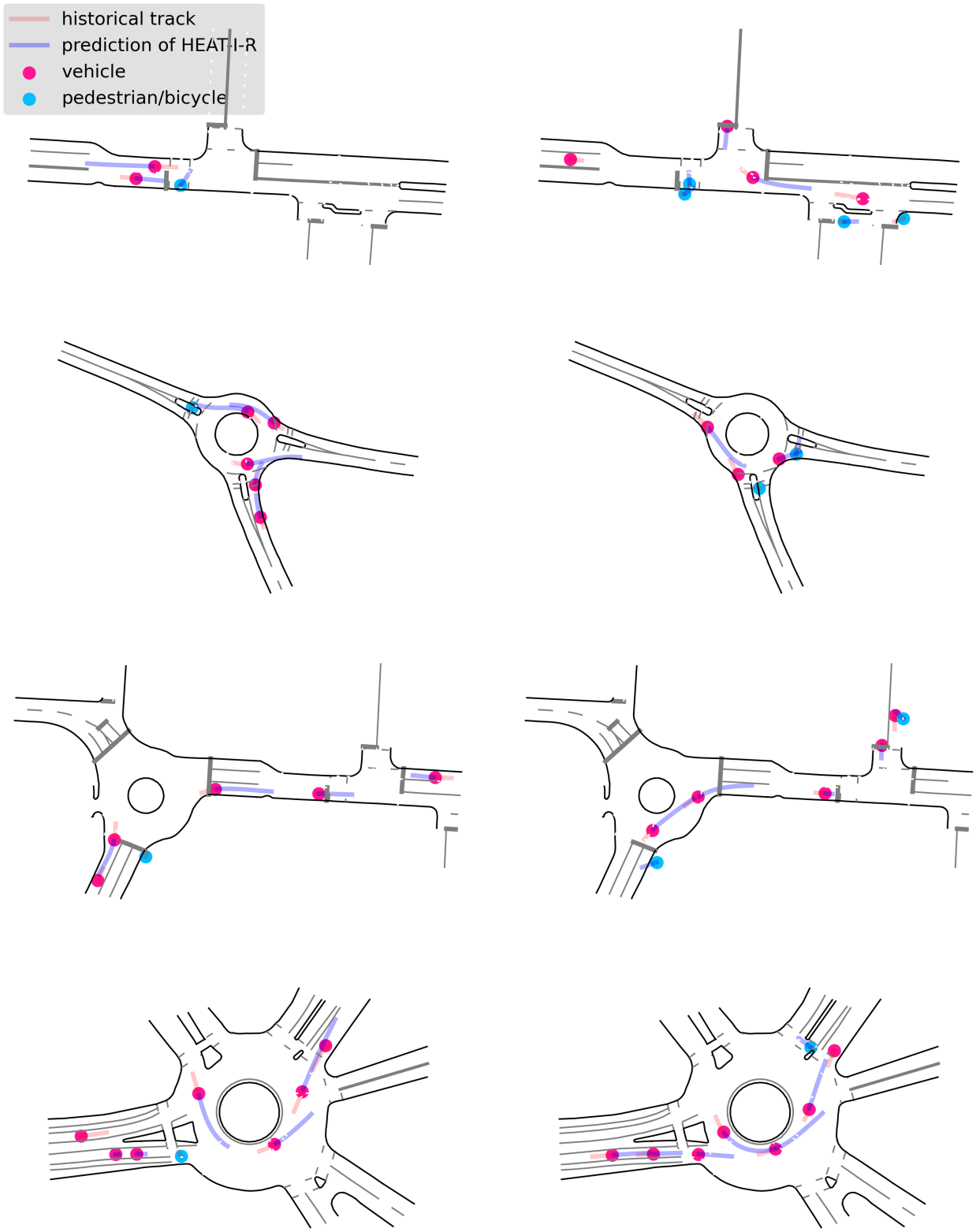


Fig. 6. **Visualized prediction results.** This figure visualizes prediction results of the proposed HEAT-based multi-agent trajectory prediction method on various driving scenarios in the INTERACTION dataset. Deep pink dots (vehicle in the legend) are the vehicles' current positions. Deep sky blue dots (pedestrian/bicyclist in the legend) are the pedestrian/bicyclist agents' current positions. Red lines (historical track in the legend) are their one-second historical trajectories and blue lines (prediction of HEAT-I-R in the legend) are the trajectories predicted by the proposed framework with HEAT. It shows that the proposed method is able to simultaneously predict trajectories of a variable number of heterogeneous agents (vehicle and pedestrian/bicyclist) in different scenarios.

Cranfield University

CERES https://dspace.lib.cranfield.ac.uk

School of Aerospace, Transport and Manufacturing (SATM) Staff publications (SATM)

2022-02-01

Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network

Mo, Xiaoyu

IEEE

Mo X, Huang Z, Xing Y, Lv C. (2022) Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network. IEEE Transactions on Intelligent Transportation Systems, Volume 23, Number 7, July 2022, pp. 9554-9567

https://doi.org/10.1109/TITS.2022.3146300

Downloaded from Cranfield Library Services E-Repository