Highlights

Heterogeneous Graph Social Pooling for Interaction-Aware Vehicle Trajectory Prediction

Xiaoyu Mo,Yang Xing,Chen Lv

- A unified trajectory prediction framework for autonomous driving, which represents traffic participants and infrastructure in a heterogeneous graph containing spatial and temporal features, is proposed.
- A novel heterogeneous graph social pooling method is proposed to model multi-modal interactions between a variable number of objects.
- Experiments conducted on two large-scale real-world datasets show that the proposed method is advantageous in terms of prediction accuracy, compared to other existing methods.

Transportation Research Part E: Logistics and Transportation Review, Volume 191, November 2024, Article number 103748
DOI: 10.1016/j.tre.2024.103748

Heterogeneous Graph Social Pooling for Interaction-Aware Vehicle Trajectory Prediction

Xiaoyu Mo[a], Yang Xing[b] and Chen Lv[a,*]

[a]School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore
[b]Centre for Autonomous and Cyber-Physical Systems at Cranfield University, MK43 0AL, , UK

ARTICLE INFO

Keywords:
Trajectory prediction
Connected vehicles
Graph neural networks
Heterogeneous interactions

ABSTRACT

Predicting the trajectories of neighboring vehicles is vital for self-driving cars in intricate real-world driving. The challenge lies in accounting for diverse influences on a vehicle's movement, travel needs, neighboring vehicles, and a local map. To address these factors comprehensively, we have developed a framework with a Heterogeneous Graph Social (HGS) pooling approach. The framework represents vehicles and infrastructures in a single graph, with vehicle nodes holding historical dynamics information and infrastructure nodes containing spatial features from map images. HGS captures vehicle-infrastructure interactions in urban driving. Unlike existing methods that are restricted to a fixed vehicle count and highway settings, HGS can accommodate variable interactions and road layouts. By merging all features, our approach predicts the target vehicle's future path. Experiments on real-world data confirm HGS's superiority, boasting quicker training and inference, affirming its feasibility, effectiveness, and efficiency.

1. Introduction

Researches on intelligent transportation systems focus on integrating advanced communication and information technologies into transportation infrastructures and vehicles to improve the efficiency, safety, and sustainability of future transportation networks Giovannelli and Vicente (2023); Huo et al. (2023); Liu et al. (2023); Tengilimoglu et al. (2023); Wang et al. (2023a); Yang et al. (2023). Recent works try to explore the applications of advances in the field of artificial intelligence on many aspects of intelligent transportation systems and autonomous driving Qu et al. (2023); Lin et al. (2023); He et al. (2023). As a key aspect of future transportation systems, autonomous vehicles are expected to improve the safety, smoothness, and efficiency of future mobility systems Liu et al. (2022); Mo et al. (2022). It is of great importance for autonomous vehicles to be aware of the motion intents of surrounding vehicles, particularly under complex driving situations Bogaerts et al. (2020); Wang et al. (2023b); Bahari et al. (2021); Lefèvre et al. (2014); Cao et al. (2022); Wang et al. (2023c). In recent years, trajectory prediction has been widely investigated by researchers using from physics-based and maneuver-based methods to learning-based interaction-aware ones Lefèvre et al. (2014). Despite significant advances in the field, trajectory prediction remains challenging due to the complex interactions among vehicles and infrastructure. Mo et al. (2020); Gao et al. (2020).

Interaction-aware approaches have garnered increasing attention from both academia and industry. Current interaction-aware trajectory prediction methods predominantly employ graph neural networks (GNNs) and/or convolutional neural networks (CNNs). A locally connected graph is constructed in Diehl et al. (2019) to model interactions, demonstrating that representing interactions as a graph enhances prediction accuracy. GRIP applies graph operations on sequential data to account for inter-vehicular interactions, while it overlooks the impact of infrastructure elements. SCALE-Net explores edge features within a graph where physical states are used as node features. The aforementioned works primarily utilize homogeneous graphs, where nodes represent only vehicles, thereby disregarding the influence of infrastructures. In contrast, VectorNet employs a graph to represent interactions among objects represented by vectors. GNNs are utilized to extract interaction features from vector representations of objects Gao et al. (2020). However, the fully connected graph in VectorNet Gao et al. (2020) is inefficient due to the exponential increase in edges with the number of nodes. LaneGCN Liang et al. (2020) designs a GNN to process lane graphs for trajectory prediction, using four edge sets to represent connections between lanelets. Both LaneGCN and VectorNet rely heavily on high-definition (HD) maps, which are costly to acquire, maintain, and update. In comparison to homogeneous

*Corresponding author
lyuchen@ntu.edu.sg (C. Lv)

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution License (CC:BY 4.0).
The final published version (version of record) is available online at DOI:10.1016/j.tre.2024.103748. Please refer to any applicable publisher terms of use.

graph-based methods that focus solely on objects of the same type (e.g., vehicles), this work aims to represent both traffic participants and infrastructure within a single heterogeneous graph. Additionally, unlike methods that depend on expensive HD maps, this work proposes a scene representation and encoding methodology that leverages easily accessible bird's-eye view (BEV) maps for heterogeneous scene encoding.

CNNs are also widely explored for modeling interaction Mo et al. (2020). Convolutional social pooling (CS-LSTM) Deo and Trivedi (2018) is a trajectory prediction model designed for highway driving. It defines a spatial grid around the target vehicle, where each surrounding vehicle is placed in a specific cell. The model uses a two-channel structure to combine the dynamics of the target vehicle and its interaction with other nearby vehicles. It considers spatial features using a 13 by 3 occupancy grid around the target vehicle. It uses LSTM encoders to extract dynamics features from all the vehicles' past trajectories and put these feature vectors into the occupancy grid to form a social tensor. Then a CNN is applied to process the social tensor to obtain a social context feature for the target vehicle. However, in heavy traffic, the model may consider redundant surrounding vehicles where all the cells are occupied, and it did not consider the target vehicle for modeling interaction. More importantly, CS-LSTM does not consider the local road structure, and thus cannot be applied to urban driving situations where road structures can be very diverse and complex. The Multi-Agent Tensor Fusion (MATF) framework Zhao et al. (2019) enhancesCS-LSTM by incorporating a CNN to encode the static scene context from a BEV image. It then fuses this encoded scene context with the dynamic features of multiple agents to predict their future trajectories. MATF achieves this fusion through spatial alignment, which ensures that the spatial relationships between agents and scene features are preserved. While MATF does consider vehicle-infrastructure interactions, its scope is currently limited to the infrastructure represented within the BEV map. This limitation presents a challenge for extending MATF to incorporate additional types of infrastructures in the future, such as traffic signals, pedestrian crossings, and other road elements not captured in a BEV map. Thus, while MATF provides a robust framework for trajectory prediction in its current form, its adaptability to more complex and diverse infrastructural elements remains constrained. MHA-JAM Messaoud et al. (2021) generalizes CS-LSTM by combining the interaction feature and map feature in a way similar to MATF. It discretizes the local area to a grid and places a vehicle into a cell in a similar way to CS-LSTM. The grid can only contain as many interactive vehicles as the number of cells. In contrast, our graph representation can contain a variable number of interacting vehicles.

This study tackles the complex problem of predicting vehicle trajectories by integrating both surrounding vehicles and infrastructure elements. Assuming access to dynamic state information for vehicles and the local map data, we introduce an interaction-aware trajectory prediction framework that leverages a Heterogeneous Graph Social (HGS) pooling method. To represent the scene efficiently and uniformly, we construct a star-like heterogeneous graph, which includes an additional map node encapsulating the local map features of the target vehicle. The inherent sparsity of the star-like graph structure ensures that the number of edges increases linearly with the number of nodes. As illustrated in Fig. 1, our framework utilizes a shared vehicle dynamics encoder across all vehicles to extract individual features. We design a rotation-sensitive CNN to capture map features and construct a target-centered graph to model vehicle-vehicle and vehicle-infrastructure relationships. Additionally, we propose a novel HGS pooling mechanism that can handle an arbitrary number of vehicles, thereby enhancing the exploration and utilization of vehicle-infrastructure interactions for trajectory prediction. Our proposed framework integrates historical vehicle dynamics and local BEV map data. It also supports the integration of other data sources, such as HD maps, seamlessly within the designed structure. This capability allows the framework to leverage detailed environmental information to further enhance the prediction accuracy.

Compared to CS-LSTM Deo and Trivedi (2018), which is designed for highway driving, HGS represents the driving scene using a unified heterogeneous graph that includes an additional map node, making it well-suited for complex urban driving environments. Unlike existing approaches for urban scenarios, such as MATF Zhao et al. (2019) and MHA-JAM Messaoud et al. (2021), which rely heavily on BEV maps, HGS is agnostic to map representation. The key advantages of HGS include its capacity to handle complex urban driving scenarios, its flexibility in map representation, and its improved integration of temporal and spatial features, avoiding the need for meticulous spatial alignment Zhao et al. (2019) between agents and the local map. In contrast to methods that depend heavily on HD maps Gao et al. (2020); Liang et al. (2020), HGS utilizes an easily accessible BEV map of the local area. Additionally, compared to methods employing CNN backbones for processing BEV maps, this work introduces a rotation-sensitive CNN for map encoding, which is significantly smaller than typical CNN backbones. This results in faster training and inference times, which is crucial for real-time applications in autonomous driving.

The main contributions of this work are summarized as follows:

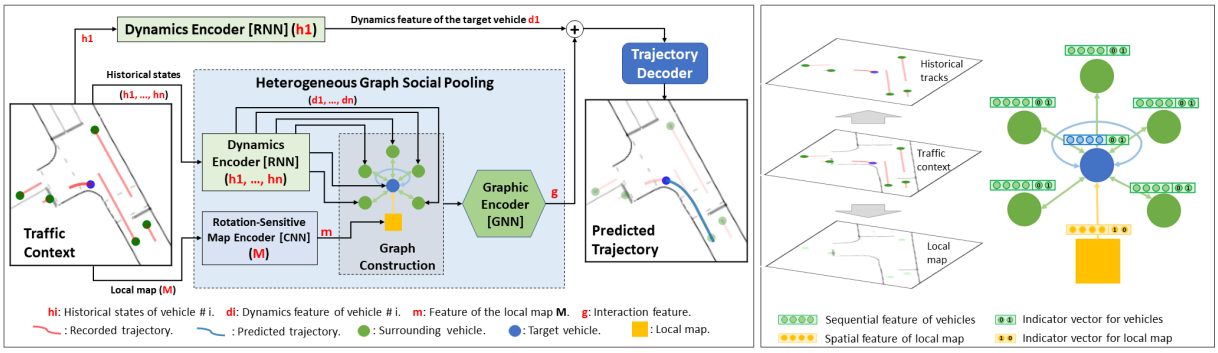


Figure 1: The proposed scheme with the constructed graph. *Left*, it consists of three encoders: a dynamics encoder (implemented using a recurrent neural network (RNN)) for vehicle dynamics feature extraction, a map encoder (implemented using a CNN) for understanding road structure information (m), and an HGS pooling (implemented using a GNN) for obtaining interaction features among traffic participants and the local map. The encoded dynamics $(d_i, \cdots, d_n)$ and interaction (g) features are then concatenated and sent to the trajectory decoder for trajectory prediction. *Right*, the traffic context is split into two parts: vehicles' historical states and the local map, and the target-centered heterogeneous graph is constructed with local connections.

- We propose an efficient interaction-aware trajectory prediction framework that achieves faster inference time and reduced training cost without compromising performance, utilizing a novel HGS pooling method for urban autonomous driving. HGS effectively models the interdependencies among a variable number of traffic participants and the local map by representing the driving scene with a star-like heterogeneous graph and applying a GNN to process the graph.
- We utilize a BEV image of the local map and design a rotation-sensitive CNN to extract spatial features, significantly reducing the model size compared to approaches using CNN backbones (e.g., ResNet He et al. (2016)) for BEV maps.
- Within the proposed efficient prediction framework, we investigate the optimal selection of input data (states and traceback horizon) for trajectory prediction, providing insights that can reduce computational efforts in trajectory prediction modules (e.g., avoiding the use of excessively long historical states).

2. Methodology

The primary objective of this study is to accurately forecast the future trajectory of a target vehicle navigating through complex urban driving environments. This prediction hinges on understanding and modeling the intricate interactions that the target vehicle engages in with its surrounding traffic participants and the road infrastructure. These interactions are dynamic and complicated, encompassing not only the immediate movements of nearby vehicles but also the broader spatial context provided by the road layout. Our approach integrates both temporal and spatial dimensions of information to predict the path ahead. Temporally, we analyze the historical dynamics of the target vehicle and its surrounding traffic, predicting future movements based on observed behaviors and trends. Spatially, we leverage detailed maps, such as BEV maps or HD maps, to enhance our understanding of the surrounding infrastructure.

The input is two-fold: $\mathbf{X}_t = [\mathcal{H}_t, \mathcal{M}_t]$, where $\mathcal{H}_t = \{h_t^1, \cdots, h_t^n\}$ represents historical states of n vehicles at time t, and $h_t^i = [s_{t-T_h+1}^i, s_{t-T_h+2}^i, \cdots, s_t^i]$ represents vehicle i's ($i = 1$ for target vehicle) historical states at time t. T_h is the traceback horizon. $\mathcal{M}_t$ denotes the local map centered around the target vehicle.

The output is the predicted trajectory of the target vehicle:

$$f_t^1 = [(x_{t+1}^1, y_{t+1}^1), (x_{t+2}^1, y_{t+2}^1), \cdots, (x_{t+T_f}^1, y_{t+T_f}^1)], \tag{1}$$

where T_f is the prediction horizon.

2.1. Trajectory Prediction with HGS Pooling

As depicted in Fig. 1, The proposed framework comprises several key components: shared dynamics encoders, a map encoder, an HGS pooling block, and a trajectory decoder. The temporal dynamics encoders analyze the historical states of individual vehicles, while the rotation-sensitive map encoder processes the local map to extract spatial features. The framework integrates vehicle dynamics and road structure features within a target-centered heterogeneous graph, facilitating the extraction of high-level interaction features by the HGS pooling block.

2.1.1. Temporal Dynamics Encoder

This study employs RNNs to process vehicles' historical states, recognizing that these past states form a sequence with temporal dependencies. RNN variants Hochreiter and Schmidhuber (1997); Chung et al. (2014) have demonstrated proficiency in modeling sequences for various complex tasks, including statistical machine translation, where input and output sequences may vary in length. It is worth noting that the proposed framework is agnostic to the dynamics encoder.

A shared temporal dynamics encoder is utilized for each vehicle to capture their sequential features from historical states. These historical states, which extend beyond mere positions in XY-coordinates, can include additional information such as vehicle velocities and yaw angles. The extendability of the proposed model is further elaborated in Sub.Sec. 4.2. Eq. 2 illustrates the dynamics encoder applied to h_t^1 to extract its dynamics feature d_t^1:

$$d_t^1 = \mathrm{FC}_1(\mathrm{Dyn}_{\mathrm{enc}}(\mathrm{Emb}(h_t^1))), \tag{2}$$

where Emb() stands for input embedding. $\mathrm{Dyn}_{\mathrm{enc}}$ denotes the RNN for dynamics encoding. FC_1 is a fully connected layer. All vehicles' dynamics features serve as vehicle-node features of the spatiotemporal interaction graph (refer to Fig. 1).

2.1.2. Rotation-Sensitive Map Encoder

Maps play a crucial role in shaping the driving behaviors of vehicles, making them indispensable for trajectory prediction, particularly in urban environments. In this study, we adopt a BEV representation of the map due to its accessibility and comprehensive depiction of road structures. Specifically, a grayscale image map serves to convey road information. While existing CNNs are widely utilized for many tasks Krizhevsky et al. (2017); Redmon et al. (2016), their reliance on max-pooling layers, which enforce rotation invariance, renders them less suitable for map processing. In driving applications, maps also contain directional cues, and a rotated map could potentially mislead vehicle navigation. Therefore, instead of leveraging pre-trained CNN models, we designed a novel CNN architecture (Sub.Sec. 3.3) (devoid of max-pooling layers) specifically tailored for map feature extraction:

$$m_t = \mathrm{FC}_2(\mathrm{Map}_{\mathrm{enc}}(\mathcal{M}_t)), \tag{3}$$

where $\mathrm{Map}_{\mathrm{enc}}$ is the designed map encoder. FC_2 is a fully connected layer. m_t is the local map feature in the spatiotemporal interaction graph (Fig. 1). Note that m_t and d_t^i share the same shape.

2.1.3. Heterogeneous Graph Social (HGS) Pooling

In this study, the relationships between vehicles, as well as between vehicles and infrastructure, are modeled using a directed heterogeneous graph. Additionally, an HGS pooling block is devised for interaction modeling. In the constructed interaction graph, each node represents either a vehicle or a map element, while an edge from one node to another shows the directed influence of the former on the latter. Then these heuristically represented relationships are processed by the designed HGS. The GNN layer in HGS allows information to flow from the source node to the target node. The information flow is regulated by the weights assigned to each neighboring node. The chosen representation is motivated by several key factors. Firstly, graphs provide a natural framework for representing entities as nodes and their interactions as edges, akin to social networks. This structure is particularly advantageous as it allows for the dynamic modeling of relationships in complex systems like traffic scenarios, where the number of interactive vehicles can vary. Moreover, the use of a heterogeneous graph accommodates the simultaneous inclusion of different object types, such as vehicles and infrastructure elements like roads, facilitating a comprehensive analysis of their interactions within a unified framework.

A key challenge in using graphs for traffic representation is the task-specific construction of the graph. Two extreme cases highlight this issue: 1) A graph without inter-node edges ignores inter-node relationships, and 2) A fully

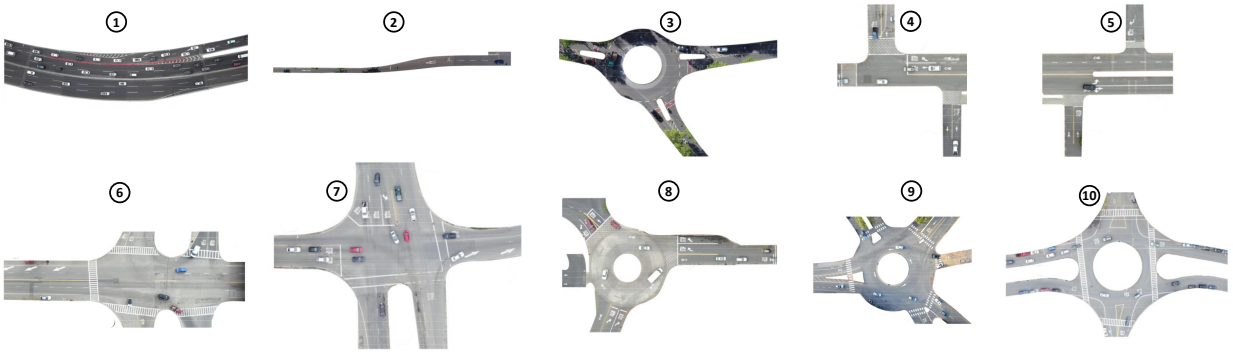


Figure 2: The interactive driving scenarios Zhan et al. (2019b).

connected graph introduces unnecessary edges, increasing quadratically with the number of nodes. A more practical approach involves constructing the interaction graph with neighboring connections. The graph structure we designed is represented by an edge set E, where all the traffic participants are connected with the target vehicle with bi-directional edges and the map node is connected to it with a directed edge. The target-centered interaction graph (see an example in Fig. 1) reflects the mutual influence between traffic participants and the map's influence on the target vehicle's future motion.

Vehicle nodes encapsulate the dynamics features, and a map node captures the spatial feature. It is noteworthy that the selection of distance thresholds and map sizes can vary across different studies, and for this study, these parameters were empirically determined. Specifically, neighboring vehicles are identified based on a distance threshold of 20 meters from the target vehicle. The local map is 40×40 m^2 area (400×400 pixels), centered around the target vehicle. To manage the heterogeneity of nodes within the graph, node features are concatenated with an indicator one-hot vector: [0, 1] for vehicle nodes and [1, 0] for road nodes.

GNNs are used in processing graph-represented interactions to extract meaningful features. They have demonstrated superior performance across a range of tasks involving graph data Veličković et al. (2018). GNNs encode interactions among nodes by aggregating information from variable-sized neighborhoods, thereby generating comprehensive feature vectors that capture the relational dependencies within the graph structure. Graph Convolutional Networks (GCNs) pioneer to generalize convolutional operations to graphs, allowing neural networks to operate directly on irregular structures (graphs). In a GCN, convolution is performed by aggregating neighboring node features (vectors learned by neural networks) using the normalized adjacency matrix of the graph. GCNs have been applied successfully to various tasks involving graph-structured data, including node classification, link prediction, and graph classification. However, their reliance on a fixed aggregation scheme (e.g., predefined adjacency matrix) limits their application to distinguish nodes of different importance, especially in autonomous driving situations, where the graph structure (i.e., the adjacency matrix) is determined by the user. Graph Attention Networks (GATs) generalized GCNs by introducing attention mechanisms into the information aggregation to assign weights to edges based on the importance of the relationship between nodes. Let h_i be the feature vector of node i. The attention coefficient e_{ij} for nodes i and j is computed as: $e_{ij} = \text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}h_i \| \mathbf{W}h_j])$, where $[\cdot\|\cdot]$ denotes concatenation, $\mathbf{W}$ is a learnable weight matrix, and $\mathbf{a}$ is the attention mechanism. The coefficients are normalized using the softmax function: $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}(i)}\exp(e_{ik})}$. The node features are updated as a weighted sum of their neighbors' features:

$$h_i' = \sigma\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\mathbf{W}h_j\Big), \tag{4}$$

where σ is a non-linear activation function.

We design an HGS pooling block to process the spatiotemporal graph. Given all the vehicles' dynamics features ($D_t = \text{FC}_1(\text{Dyn}_{\text{enc}}(\text{Emb}(\mathcal{H}_t)))$ is obtained by applying Eq. 2 to all the vehicles in parallel) and the target vehicle's local map feature (m_t) out from corresponding encoders, HGS needs to first construct the node feature matrix ($[D_t \| m_t]$, indicator vectors are added to each node feature as shown in the right part of Fig. 1) for the GNN used in this module.

The calculation of the HGS block is shown in Eq. 5.

$$g_t = \mathrm{FC}_3(\mathrm{Inter}_{\mathrm{enc}}([D_t \| m_t], E_t)), \tag{5}$$

where $[\cdot\|\cdot]$ stands for concatenation, E_t is the edge set, $\mathrm{Inter}_{\mathrm{enc}}$ denotes the GNN used for interaction, FC_3 is a fully connected layer, and g_t is the target vehicle's interaction feature. HGS models the interactions between vehicles and the map in parallel. It can distinguish the map and vehicle nodes because of the indicator vectors and can capture the spatial relationships between them since all the vehicles and the map are represented in the target-centered coordinate framework. Note that $\mathrm{Inter}_{\mathrm{enc}}$ can be implemented with different GNNs.

HGS pooling can handle a wide range of driving scenarios. Firstly, it can be applied to different road structures due to its ability to represent the local map as a BEV image, which can capture varying road configurations. Secondly, the HGS pooling method can accommodate an arbitrary number of interacting vehicles by using a graph representation that is flexible enough to incorporate any number of objects (such as vehicles) and their interrelationships. This makes it a versatile approach for handling complex driving scenarios.

2.1.4. Trajectory Decoder

We apply an RNN to the concatenation of g_t and d_t^1 for trajectory decoding. RNN is chosen for its inherent capability to capture sequential dependencies within trajectories. Eq. 6 outlines the computation of the trajectory decoder:

$$f_t^1 = \mathrm{FC}_4(\mathrm{Traj}_{\mathrm{dec}}([g_t \| d_t^1])), \tag{6}$$

where $\mathrm{Traj}_{\mathrm{dec}}$ represents the decoder, and FC_4 is a fully connected layer that maps the high-dimensional features to positions (a sequence of XY-coordinates).

3. Experimental Setup

3.1. Data

3.1.1. Dataset

The model development is based on the INTERACTION dataset Zhan et al. (2019b), which comprises diverse and interactive driving scenarios captured globally. These scenarios include complex driving situations such as highway ramps, roundabouts, and intersections. Each scenario in the dataset provides recorded trajectories of vehicles and other traffic participants (e.g., pedestrians, bicyclists), along with HD maps of the environments. Notably, trajectories are not pre-segmented, allowing flexibility in segmentation according to specific experimental needs.

In this study, ten driving scenarios recommended by the online benchmark Zhan et al. (2020a) (as depicted in Fig. 2) are selected for analysis. The entire INTERACTION dataset is partitioned following guidelines in Zhan et al. (2020b). Trajectories within the dataset are segmented into intervals of 10 seconds each, where the first five seconds serve as historical states used for input, while the subsequent five seconds are treated as ground truth for trajectory prediction evaluation. After preprocessing, the training set consists of 317, 335 data segments, while the validation set comprises 90, 219 data segments.

3.1.2. Data Processing

We adopt a fixed stationary reference frame centered on the target vehicle's position at time t for all trajectory computations. Velocities are maintained consistent with their recorded values, while the yaw angle of the target vehicle at t is standardized to zero, with adjustments made to other yaw angles as necessary. The local map encompasses a 40×40 square meters area centered around the target vehicle, oriented parallel to its current heading. This spatial representation is captured by a 160×160 array.

3.2. Evaluation Metrics

In this work, the performance of the prediction method is evaluated using three metrics, i.e., the Displacement Error at time τ (DE_τ), the Average Displacement Error (ADE) over trajectories, and the Final Displacement Error (FDE). ADE and FDE are two commonly used metrics in evaluating the performance of trajectories Liang et al. (2020); Zhan et al. (2019a). They are all measured in meters. The lower, the better. DE_τ, ADE, and FDE between two trajectories can be calculated by Eq. 7, Eq. 8, Eq. 9, respectively.

$$DE_\tau = \sqrt{(\hat{x}_\tau - x_\tau)^2 + (\hat{y}_\tau - y_\tau)^2}, \tag{7}$$

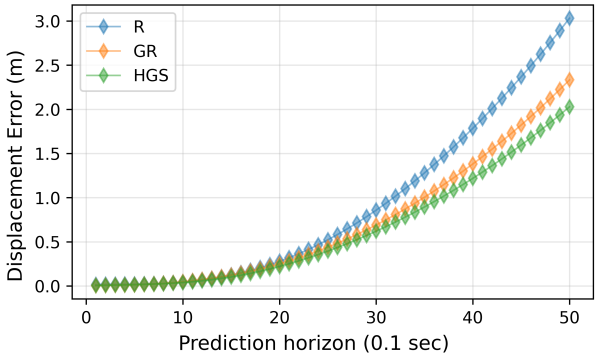


Figure 3: Results of the ablation study. State inputs: (x, y, v_x, v_y, ψ). $T_h = 30$ (3 sec). $T_f = 50$ (5 sec).

$$ADE = \frac{1}{T_f}\sum_{\tau=1}^{T_f} DE_\tau = \frac{1}{T_f}\sum_{\tau=1}^{T_f}\sqrt{(\hat{x}_\tau - x_\tau)^2 + (\hat{y}_\tau - y_\tau)^2}, \tag{8}$$

$$FDE = DE_{T_f} = \sqrt{(\hat{x}_{T_f} - x_{T_f})^2 + (\hat{y}_{T_f} - y_{T_f})^2}, \tag{9}$$

where $(\hat{x}_\tau, \hat{y}_\tau)$ is the predicted position in the XY-coordinate at time τ, (x_τ, y_τ) is the ground truth, and T_f is the prediction horizon.

3.3. Implementation Details

PyTorch and PyTorch Geometric Fey and Lenssen (2019) are used for implementations. The model undergoes end-to-end training for 10 epochs using the Adam optimizer Kingma and Ba (2014) with a scheduled learning rate. The initial learning rate of 0.001 halves at epochs 1, 2, 4, and 6 to minimize the Average Displacement Error (ADE) defined in Eq. 8. Historical states are first embedded into a 64-dimensional space before being processed by the temporal dynamics encoder. The dynamics encoder consists of a single-layer RNN with a hidden size of 64, while the trajectory decoder comprises two layers with a hidden size of 128. The rotation-sensitive map encoder employs a three-layer convolutional architecture with configurations ([#_$filters$, $kernel_size$, $stride$]) [8, 16, 4], [16, 8, 4], and [32, 4, 2] for its respective layers (each layer is structured as [Conv, LeakyReLU, BatchNorm]). For higher-level interaction feature extraction, the HGS pooling block utilizes two GNN layers with Leaky-ReLU activation (slope = 0.1) consistently applied throughout the implementation.

4. Results

In this section, we begin by conducting an ablation study of the proposed method (Sub.Sec. 4.1). We then explore the extendibility and renewability of HGS (Sub.Sec. 4.2). Following this, we analyze the impact of the traceback horizon on prediction performance (Sub.Sec. 4.3), followed by a study on inference time (Sub.Sec. 4.4). More results on different scenarios are reported in Sub.Sec. 4.5 Finally, we compare HGS with existing methods using test sets from two naturalistic driving datasets (Sub.Sec. 4.6). Please note that all results in Fig. 3, Fig. 4, and Fig. 5 are averaged over all scenarios. Fig. 6, Fig. 7, and Fig. 8 present results from different study areas (10 road structures in Fig. 2) with varying prediction horizons.

4.1. Ablation Study

The ablation study aims to validate the effectiveness and superiority of the proposed approach by comparing the following methods:

- **R**: This model predicts the future trajectory of a vehicle using only its individual dynamics feature extracted by a single dynamics encoder. It focuses solely on the vehicle's internal state without considering interactions with other vehicles or infrastructures.

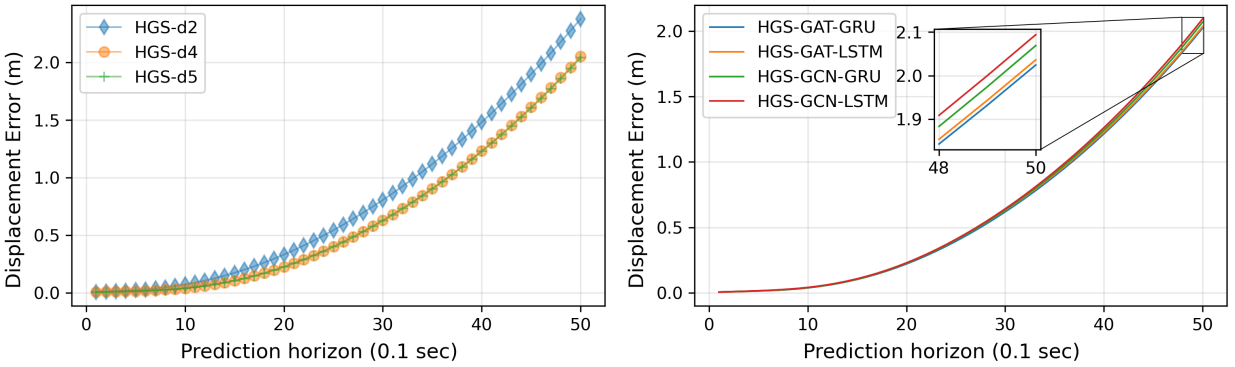


Figure 4: *Left*, **Extendability of HGS.** HGS-d2: HGS using (x, y) as input states; HGS-d4: HGS using (x, y, v_x, v_y) for input states; HGS-d5: HGS using (x, y, v_x, v_y, ψ) for input states. *Right*, **Renewability of HGS.** State inputs: (x, y, v_x, v_y). $T_h = 30$ (3 sec). $T_f = 50$ (5 sec).

- **GR**: This model employs a homogeneous graph where each node represents a vehicle, enabling it to capture inter-vehicle interactions. It models how vehicles influence each other, but it does not account for interactions with infrastructures such as roads or traffic signals.
- **HGS**: This model represents the implementation of the proposed Heterogeneous Graph Social (HGS) framework. Unlike the previous models, HGS incorporates both vehicle-vehicle interactions and vehicle-infrastructure interactions using a heterogeneous graph. This allows it to model complex scenarios where vehicles interact not only with each other but also with road structures and other environmental factors.

All the models use $((x, y, v_x, v_y, \psi))$ for historical states.

The results are shown in Fig. 3. It can be observed that the interaction-aware models (GR and HGS) outperform the non-interaction-aware one (R) with lower displacement error (DE_τ) over time, especially when the prediction horizon exceeds 3 seconds. It can be observed that HGS shows a lower FDE ($DE_{\tau=T_f}$) than its ablations. It can also be deduced from the figure that HGS also shows a lower ADE as the green (HGS) line is under the blue (R) and orange (GR) lines. This observation aligns with findings from previous studies Deo and Trivedi (2018); Mo et al. (2020), underscoring the importance of modeling interactions for trajectory prediction. Furthermore, the proposed HGS method demonstrates superior performance over the homogeneous GR method. This outcome highlights several advancements of the proposed approach: 1) modeling vehicle-infrastructure interactions using a heterogeneous graph, 2) extracting features from different nodes using distinct encoders, and 3) utilizing a graphical encoder to derive high-level interaction features. In this study, we showcase two types of nodes, with the potential for incorporating additional node types into the framework in future research.

4.2. Extendability and Renewability of HGS

The proposed scheme models interactions among vehicles and infrastructure using a heterogeneous graph, which imposes no limitations on input types. In this framework, the heterogeneous graph comprises two types of nodes: vehicle nodes and a road node. This structure can be further extended by incorporating additional types of nodes to represent other traffic infrastructure, such as traffic signals, as well as other agents like pedestrians and bicyclists. This flexibility allows for a more comprehensive representation of the traffic environment, enhancing the accuracy and robustness of trajectory predictions in diverse driving scenarios. We leave this extension for future work and demonstrate its flexibility in the selection of encoders as following.

The dynamics encoder in our proposed framework is designed to be versatile, accommodating various types of information without constraints, thus allowing for the integration of richer data when available. To demonstrate this flexibility, we instantiated our model with three different input sets: (x, y), (x, y, v_x, v_y), and (x, y, v_x, v_y, ψ). The results, shown in Fig. 4, indicate that including velocity components in addition to positions reduces displacement error, while adding the yaw angle does not significantly improve performance. It is noteworthy that this implementation employs GAT as the graph encoder within the HGS block and Gated Recurrent Units (GRU) as the dynamics encoder.

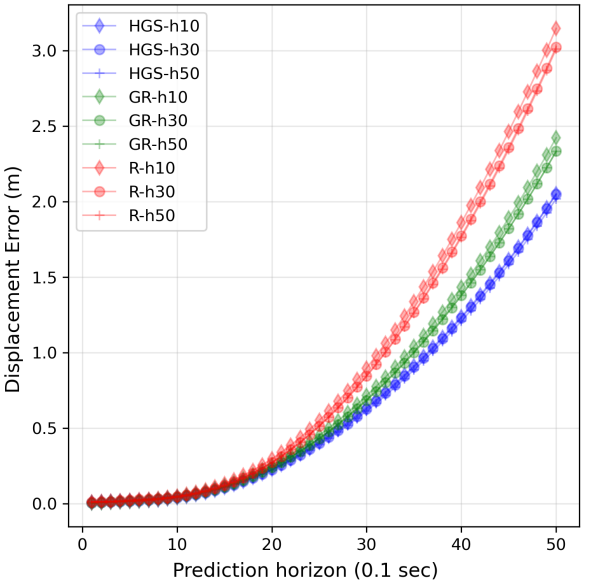


Figure 5: Impacts of traceback horizon. $T_h = 10, 30, 50$ (1, 3, 5 sec). $T_f = 50$ (5 sec). State inputs: (x, y, v_x, v_y). HGS-GAT-GRU.

The main modules of the proposed framework are designed to be decoupled from each other, allowing for independent updates without affecting the overall system. The model does not impose any specific requirements on the recurrent unit used for the dynamics encoder, which could be LSTM Hochreiter and Schmidhuber (1997), GRU Chung et al. (2014), or other variants. In this study, we utilize a graph encoder to capture interactions among vehicles and infrastructure within the constructed graph, which incorporates the relevant features. This graph encoder can be a GCN Kipf and Welling (2016), GAT Veličković et al. (2018), or any other user-designed graph encoder.

To demonstrate the versatility of the HGS framework, we implemented it using different dynamics and graph encoders and evaluated their respective and combined performances. We selected GCN for their simplicity and effectiveness, and GAT for their ability to differentiate neighboring nodes using an attention mechanism. The results presented in Fig. 4 indicate that methods based on GAT slightly outperform those based on GCN, with the combination of GAT and GRU achieving the highest performance in trajectory prediction. It's noteworthy that while our experiments show the HGS framework can be updated with different encoders, the differences observed were not statistically significant. Exploring HGS with alternative temporal and graph operators remains a promising avenue for future research.

4.3. Impacts of Traceback Horizon

We investigate the impact of T_h on model performance. We evaluate three different traceback horizons $(10, 30, 50)$ using input data (x, y, v_x, v_y). As shown in Fig. 5, extending T_h from 10 to 30 significantly improves the performance of the R and GR models. However, further extending T_h to 50 results in minimal additional improvement. For the Heterogeneous Graph Social (HGS) model, the differences in performance across varying traceback horizons are marginal. Fig. 5 also illustrates that the effects of traceback horizons vary among different methods. Notably, the proposed framework consistently outperforms its ablations, even with a very short traceback horizon. This enhancement improves both accuracy and computational efficiency. The increased performance is attributed to the reduced difficulty and lower computational resources required to collect data over shorter traceback horizons, making the proposed method highly effective for real-time applications.

4.4. Computational Complexity

Computational complexity is a crucial factor for prediction methods designed for onboard implementation, especially on platforms with limited computing resources, such as CPU-only environments. To evaluate the efficiency of our proposed Heterogeneous Graph Social (HGS) pooling approach and its ablations, we conducted a series of

Table 1
Computational complexity analysis

R (CPU)			GR (CPU)			HGS (CPU)		
No. data	T_h (ms)	Time (s)	No. data	T_h (ms)	Time (s)	No. data	T_h (ms)	Time (s)
1,000×1	10	**2.99**	1,000×1	10	**3.75**	1,000×1	10	**4.28**
1,000×1	30	3.58	1,000×1	30	4.32	1,000×1	30	4.87
1,000×1	50	4.03	1,000×1	50	4.81	1,000×1	50	5.33
1,000×100	10	**36.62**	1,000×100	10	**47.31**	1,000×100	10	**52.89**
1,000×100	30	44.39	1,000×100	30	55.97	1,000×100	30	61.19
1,000×100	50	53.75	1,000×100	50	63.41	1,000×100	50	70.59
R (GPU)			GR (GPU)			HGS (GPU)		
No. data	T_h (ms)	Time (s)	No. data	T_h (ms)	Time (s)	No. data	T_h (ms)	Time (s)
1,000×100	10	26.93	1,000×100	10	32.73	1,000×100	10	33.75
1,000×100	30	27.37	1,000×100	30	33.25	1,000×100	30	34.39
1,000×100	50	27.52	1,000×100	50	34.27	1,000×100	50	34.84

experiments to measure inference times under various conditions. These experiments were performed on a CPU (Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz) using different batch sizes (1 and 100) and traceback horizons ($T_h = 10, 30, 50$) over 1,000 batches. The input data consisted of sequences of (x, y, v_x, v_y). The results of these experiments, which are summarized in Tab. 1, provide valuable insights into the computational demands of our method and its variants under different operational conditions. The results indicate that a larger traceback horizon (T_h) consistently leads to longer inference times across all models. Specifically, HGS with $T_h = 50$ requires significantly more computational time compared to HGS with $T_h = 10$. Notably, reducing the traceback horizon to $T_h = 10$ can save about 20% of the inference time without a significant loss in prediction accuracy. This finding suggests that a shorter traceback horizon can enhance computational efficiency while maintaining reliable performance, making it a practical choice for real-time applications in resource-constrained environments. We also show the computational complexity on a GPU (GeForce RTX 2080) with a batch size of 100. It can be observed that a longer traceback horizon leads to a longer computation time on the GPU. But the difference is not as obvious as on the CPUs.

4.5. Comparison on Ten Scenarios

The INTERACTION dataset provides 10 different study areas as shown in Fig. 2. It can be observed that those study areas have different road structures, which implies different traffic complexities. So it is worth comparing the proposed method (HGS) with its ablations in different traffic complexities. We break down the validation set and report the results in 10 study areas in Fig. 6, Fig. 7, and Fig. 8. Fig. 6 shows the break-down results when at a prediction horizon of 2 seconds, Fig. 7 shows the results at 3 seconds, and Fig. 8 shows that at the longest prediction horizon. It can be observed that the proposed HGS outperforms its ablations across all the study areas at all prediction horizons, showing the effectiveness of the designed model. It is clear that study area 3 is the hardest, all methods show high displacement errors. In this hardest condition, HGS shows a great improvement compared to its ablation GR, showing that incorporating the map information with a heterogeneous graph into the model design is quite important, especially in situations with rare road structures.

4.6. Comparison with Existing Methods

For comparison with existing methodologies, our approach was trained with data augmentation and evaluated on the INTERACTION test set, a platform for competitive benchmarking. Our proposed method, designated as GH_29_3 for submission, achieved a Final Displacement Error (FDE) of 0.6381 meters for a prediction horizon of 30 frames (3 seconds), outperforming other participants.

Additionally, we conducted a comparative evaluation of our method against existing approaches using the Argoverse dataset Chang et al. (2019), introducing the miss rate (MR) as an additional metric. The MR quantifies

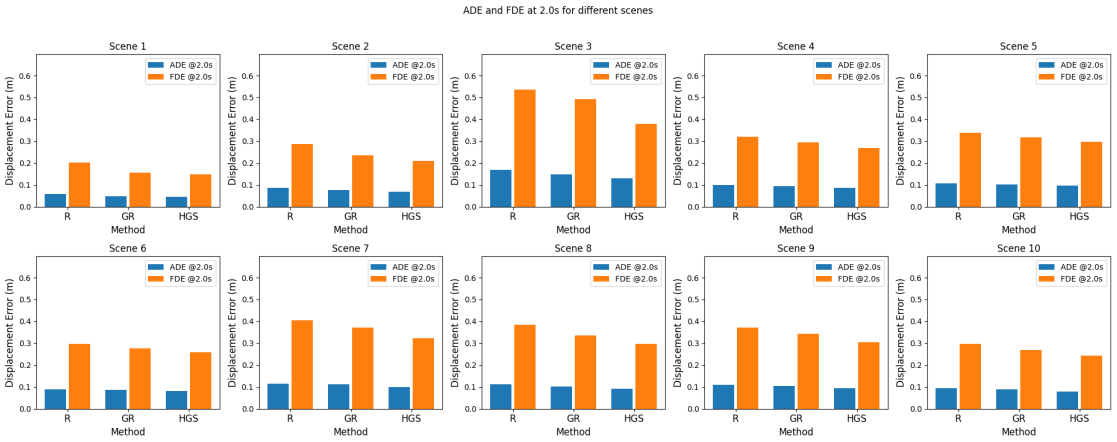


Figure 6: Results at 2 second. Traceback horizon: $T_h = 30$ (3 sec). $T_f = 20$ (2 sec). Same setting with Fig. 5 except the traceback and prediction horizons.

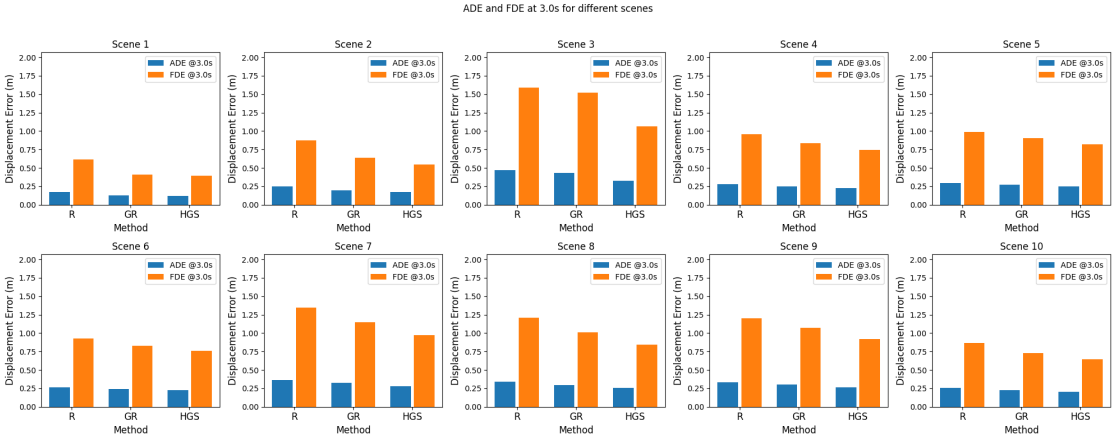


Figure 7: Results at 3 second. $T_f = 30$ (3 sec). Same setting as Fig. 6 except the prediction horizon.

scenarios where none of the predicted trajectories fall within 2.0 meters of the ground truth, assessed by endpoint error, for up to $K = 6$ predicted trajectories. Given our method's focus on single trajectory prediction, results are exclusively reported for $K = 1$. The test set is hosted online to ensure transparency and equitable comparisons. As illustrated in Tab. 2, our method consistently outperforms all baseline approaches ((Argo-(CV), Argo-(NN), Argo-(LSTM), Argo-(NN+map), Argo-(LSTM+map)) provided by the dataset holder Chang et al. (2019) in all metrics. HGS also outperforms WIMP Khandelwal et al. (2020), TNT Zhao et al. (2021), mmTransformer Liu et al. (2021) in terms of all three metrics. HGS demonstrates superior performance compared to Jean Mercat et al. (2020), in terms of FDE and MR metrics. HGS also outperforms PRIME Song et al. (2022) in ADE metric and Scene Transformer Ngiam et al. (2021) in both ADE and FDE metrics. Additionally, HGS shows comparable performance with LaneGCN Liang et al. (2020). It is worth noting that HGS achieves competitive performance on the Argoverse benchmark with greatly reduced training efforts. We compare the training efforts of HGS and two baselines in Tab. 3, where M stands for million. LaneGCN is trained on 4 TITAN-X GPUs using a batch size of 128 for 11.5 hours. Scene Transformer is trained on a TPU using a batch size of 64 for 3 days. Our method is trained on a single GPU using a batch size of 128 for 3 hours. The smaller model size and simpler structure of HGS also lead to a faster inference. Overall, HGS offers competitive results while requiring a smaller model size and fewer training efforts (note that the number of the overall HGS method (2.5 M) is much less than that of ResNet-18 (12 M) He et al. (2016), the CNN backbone used in many prediction methods). Additionally, it has faster inference times, which makes it more efficient and suitable for real-time

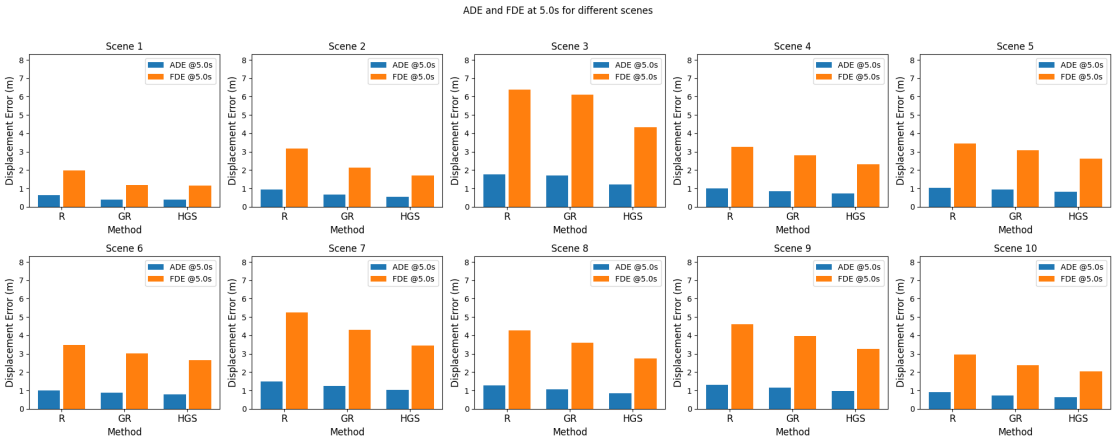


Figure 8: Results at 5 second. Prediction horizon: $T_f = 50$ (5 sec). Same setting as Fig. 6 and Fig. 7 except the prediction horizon.

Table 2
Results on the Argoverse test set

Methods	Metrics (K=1)		
	ADE (m)	FDE (m)	MR
Argo-(CV) Chang et al. (2019)	3.53	7.89	0.83
Argo-(NN) Chang et al. (2019)	3.45	7.88	0.87
Argo-(LSTM) Chang et al. (2019)	2.15	4.97	0.75
Argo-(NN+map) Chang et al. (2019)	3.65	8.12	0.94
Argo-(LSTM+map) Chang et al. (2019)	2.92	6.45	0.98
Jean Mercat et al. (2020)	1.74	4.24	0.69
WIMP Khandelwal et al. (2020)	1.82	4.03	0.63
LaneGCN Liang et al. (2020)	**1.71**	**3.78**	**0.59**
TNT Zhao et al. (2021)	2.17	4.96	0.71
mmTransformer Liu et al. (2021)	1.77	4.00	0.62
PRIME Song et al. (2022)	1.91	3.82	**0.59**
Scene Transformer Ngiam et al. (2021)	1.81	4.06	**0.59**
HGS (Ours)	**1.75**	**3.89**	**0.61**

Table 3
Comparison on complexity

Method	Training device	Training time (No. epochs)	Batch size	No. parameter	Inference time per batch (32)
LaneGCN Liang et al. (2020)	4 × TITAN X	11.5 hours (36)	128	3.7 M	0.08 sec (GeForce RTX 2080 Ti)
mmTransformer Liu et al. (2021)	-	-	-	2.6 M	0.03 sec (GeForce RTX 2080 Ti)
DenseTNT Gu et al. (2021)	8 × 2080 Ti	5 hours (16)	64	-	0.63 sec (GeForce RTX 3080)
Scene Transformer Ngiam et al. (2021)	1 × TPU Jouppi et al. (2017)	3 days (150)	64	15 M	-
HGS (Ours)	**1 × GeForce RTX 2080**	**3 hours (36)**	128	**2.5 M**	**0.02 sec (GeForce RTX 2080)**

applications. Therefore, the proposed HGS provides a more efficient and practical solution without sacrificing much performance, making it an attractive choice for scene encoding.

5. Visualization

Visualized Results. Visualization of the HGS predictions on the INTERACTION validation set is presented in Fig. 9, where GT is the ground truth for future trajectory and HGS is the corresponding predicted trajectory. Fig. 9

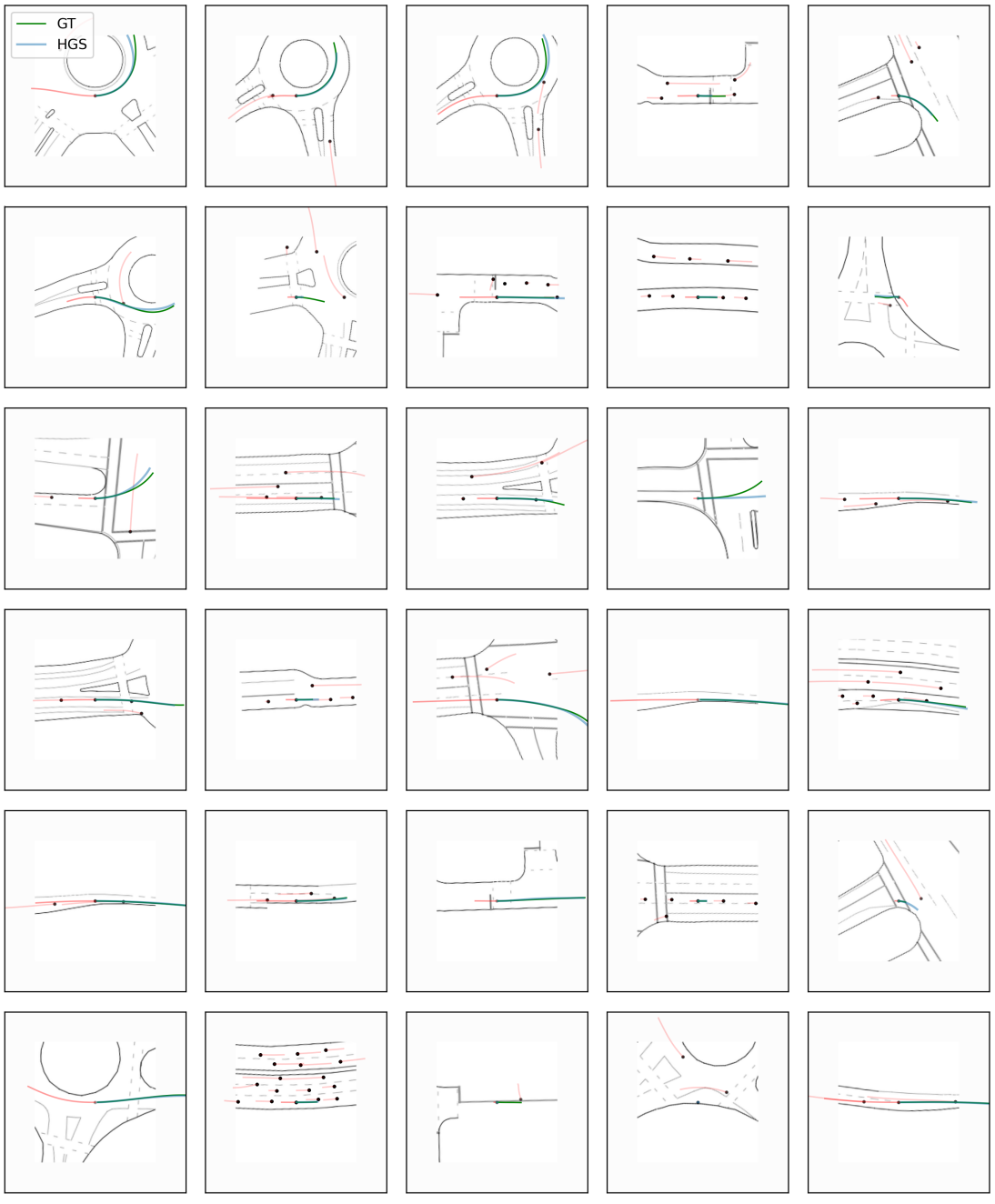


Figure 9: Visualized results of HGS using the INTERACTION validation dataset. Traceback horizon: $T_h = 30$. Prediction horizon: $T_f = 50$. Input data: (x, y, v_x, v_y). Graphic encoder: GAT. Dynamics encoder: GRU. GT: Ground truth trajectory. HGS: HGS predicted trajectory. Black dots and red lines are current positions and historical tracks of vehicles.

shows that the proposed scheme HGS is able to accurately predict the future trajectory of the target vehicle under various driving situations.

6. Discussions

In this study, we introduce and validate a unified interaction-aware trajectory prediction scheme for autonomous driving using real-world datasets collected from various cities. Experimental results demonstrate that our proposed method outperforms others by incorporating map information into the HGS pooling process. The HGS framework extends the convolutional social pooling approach (CS-LSTM) Deo and Trivedi (2018) in two significant ways. First, HGS employs a rotation-sensitive map encoder to account for the map structure, which is critical for urban driving scenarios and for modeling interactions between traffic participants and the map. In contrast, CS-LSTM overlooks road structures as highways typically feature straight roads. Second, HGS uses a unified heterogeneous graph to represent the interactions between traffic participants and the map, accommodating an arbitrary number of traffic participants. This differs from CS-LSTM, which models inter-vehicular interactions using an occupancy grid with thirty-nine cells.

The ablation study underscores the critical role of integrating interactions between traffic participants and the road map to achieve precise trajectory prediction. It highlights that neglecting these interactions can significantly compromise prediction accuracy. Furthermore, the examination of extendability reveals that the incorporation of richer information does not uniformly enhance predictive performance. This finding underscores the importance of judiciously selecting predictive features that contribute meaningfully to accuracy without introducing unnecessary complexity or computational burden. Moreover, the renewability demonstration illustrates the versatility of the HGS framework across various sequential and graphical models. While our experimentation did not yield definitive superiority among specific sequential and graphical models for HGS implementation, the framework's adaptability remains evident. This adaptability suggests potential enhancements through integration with advanced encoder architectures, fostering future exploration in model refinement. In investigating the traceback horizon, our results indicate that HGS achieves superior performance with a shorter traceback horizon compared to its counterparts. Extending this horizon does not consistently yield proportional improvements in prediction accuracy and comes at the cost of increased computational demands, as confirmed by our inference time comparisons. These insights collectively guide the design of efficient predictors, advocating for strategies that minimize data collection and computational overhead while preserving accuracy levels suitable for practical deployment.

The driving scene is represented by a heterogeneous graph and modeled using the proposed Heterogeneous Graph Social (HGS) pooling approach for trajectory prediction. It is important to emphasize that this scene representation and modeling with HGS are versatile and can be applied to a variety of other autonomous driving tasks that require detailed scene modeling. This flexibility makes our approach broadly applicable, enhancing its utility beyond trajectory prediction to potentially include tasks such as driving intention recognition Shi and Zhang (2021), driving behavior modeling Xing et al. (2021), decision-making He et al. (2022); Li et al. (2022), path planning Ma et al. (2021); Sun et al. (2022); Huang et al. (2023), and end-to-end autonomous driving Amini et al. (2020).

One limitation of this work is the constrained factors considered in scene representations due to data availability. For instance, traffic signals and regulatory rules are excluded because the data were collected from unsignalized road junctions. The absence of such critical elements may reduce the comprehensiveness of the interaction modeling in urban scenarios. Another notable limitation is the deterministic nature of the predictions, which fails to capture the inherent uncertainty in driving behaviors. Future work can address these limitations by incorporating uncertainty in two key ways. First, instead of predicting an averaged position, the model can predict a two-dimensional Gaussian distribution for each time step, providing a probabilistic representation of possible positions. Second, the multimodality of driving can be modeled by predicting multiple potential trajectories for the target vehicle, each assigned with an estimated probability. This dual approach to incorporating uncertainty can be integrated within the HGS framework by employing appropriate decoders and training losses, as suggested in previous studies Deo and Trivedi (2018); Chai et al. (2020); Cui et al. (2019). Furthermore, the current model's focus on predicting the trajectory of a single target agent represents another limitation, as it does not fully leverage the potential for parallel processing. Addressing this limitation could expedite inference time and enhance the model's applicability in real-time systems. By expanding the framework to consider the trajectories of multiple agents simultaneously, future research could significantly improve both the robustness and efficiency of trajectory prediction models.

7. Conclusions

In this study, we present an innovative framework for vehicle trajectory prediction that incorporates heterogeneous interactions between vehicles and infrastructures. Our approach involves constructing a heterogeneous graph to accurately represent these interactions, with nodes embedding features extracted from their respective encoders. To

enhance the interaction modeling, we introduce a novel Heterogeneous Graph Social (HGS) pooling block, which is designed to extract high-level interaction features. This versatile framework can also be adapted for various driving scenarios, including highway environments. Our extensive experiments on real-world driving datasets demonstrate the superior performance of the proposed method. The results indicate that our method achieves higher accuracy in predicting vehicle trajectories. Furthermore, the ablation study provides evidence that integrating heterogeneous vehicle-infrastructure interactions into the model significantly improves prediction accuracy compared to methods that only account for inter-vehicular interactions. This enhancement highlights the critical role of considering a broader range of interactions for more precise trajectory prediction.

One future improvement is to incorporate more informative HD maps rather than pictorial representations. Additionally, nodes representing other elements such as traffic signals could be integrated using appropriate encoders to enhance overall performance. Furthermore, the proposed method can be extended from predicting trajectories for a single target vehicle to simultaneous multi-agent predictions.

Acknowledgement

This work was supported in part by the Wallenberg-NTU Presidential Postdoctoral Fellowship (Award number: 023485-00001) of Nanyang Technological University, Singapore, the Agency for Science, Technology and Research (A*STAR), Singapore, under the MTC Individual Research Grant (M22K2c0079), the ANR-NRF Joint Grant (No.NRF2021-NRF-ANR003 HM Science), and the Ministry of Education (MOE), Singapore, under the Tier 2 Grant (MOE-T2EP50222-0002).

References

Amini, A., Gilitschenski, I., Phillips, J., Moseyko, J., Banerjee, R., Karaman, S., Rus, D., 2020. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. IEEE Robotics and Automation Letters 5, 1143–1150.

Bahari, M., Nejjar, I., Alahi, A., 2021. Injecting knowledge in data-driven vehicle trajectory predictors. Transportation research part C: emerging technologies 128, 103010.

Bogaerts, T., Masegosa, A.D., Angarita-Zapata, J.S., Onieva, E., Hellinckx, P., 2020. A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data. Transportation Research Part C: Emerging Technologies 112, 62–77.

Cao, D., Wang, X., Li, L., Lv, C., Na, X., Xing, Y., Li, X., Li, Y., Chen, Y., Wang, F.Y., 2022. Future directions of intelligent vehicles: Potentials, possibilities, and perspectives. IEEE Transactions on Intelligent Vehicles 7, 7–10.

Chai, Y., Sapp, B., Bansal, M., Anguelov, D., 2020. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction, in: Conference on Robot Learning, PMLR. pp. 86–99.

Chang, M.F., Lambert, J.W., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., Hays, J., 2019. Argoverse: 3d tracking and forecasting with rich maps, in: Conference on Computer Vision and Pattern Recognition (CVPR).

Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling, in: NIPS 2014 Workshop on Deep Learning, December 2014.

Cui, H., Radosavljevic, V., Chou, F.C., Lin, T.H., Nguyen, T., Huang, T.K., Schneider, J., Djuric, N., 2019. Multimodal trajectory predictions for autonomous driving using deep convolutional networks, in: 2019 International Conference on Robotics and Automation (ICRA), pp. 2090–2096. doi:10.1109/ICRA.2019.8793868.

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

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

Fey, M., Lenssen, J.E., 2019. Fast graph representation learning with PyTorch Geometric, in: ICLR Workshop on Representation Learning on Graphs and Manifolds.

Gao, J., Sun, C., Zhao, H., Shen, Y., Anguelov, D., Li, C., Schmid, C., 2020. Vectornet: Encoding hd maps and agent dynamics from vectorized representation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11525–11533.

Giovannelli, T., Vicente, L.N., 2023. An integrated assignment, routing, and speed model for roadway mobility and transportation with environmental, efficiency, and service goals. Transportation research part C: emerging technologies 152, 104144.

Gu, J., Sun, C., Zhao, H., 2021. Densetnt: End-to-end trajectory prediction from dense goal sets, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15303–15312.

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

He, X., Chen, H., Lv, C., 2023. Robust multiagent reinforcement learning toward coordinated decision-making of automated vehicles. SAE International Journal of Vehicle Dynamics, Stability, and NVH 7, 475.

He, X., Yang, H., Hu, Z., Lv, C., 2022. Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach. IEEE Transactions on Intelligent Vehicles .

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

Huang, Z., Liu, H., Wu, J., Lv, C., 2023. Conditional predictive behavior planning with inverse reinforcement learning for human-like autonomous driving. IEEE Transactions on Intelligent Transportation Systems .

Huo, J., Liu, C., Chen, J., Meng, Q., Wang, J., Liu, Z., 2023. Simulation-based dynamic origin–destination matrix estimation on freeways: A bayesian optimization approach. Transportation Research Part E: Logistics and Transportation Review 173, 103108.

Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al., 2017. In-datacenter performance analysis of a tensor processing unit, in: Proceedings of the 44th annual international symposium on computer architecture, pp. 1–12.

Khandelwal, S., Qi, W., Singh, J., Hartnett, A., Ramanan, D., 2020. What-if motion prediction for autonomous driving. arXiv preprint arXiv:2008.10587 .

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM 60, 84–90.

Lefèvre, S., Vasquez, D., Laugier, C., 2014. A survey on motion prediction and risk assessment for intelligent vehicles. ROBOMECH journal 1, 1–14.

Li, G., Yang, Y., Li, S., Qu, X., Lyu, N., Li, S.E., 2022. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. Transportation research part C: emerging technologies 134, 103452.

Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R., 2020. Learning lane graph representations for motion forecasting, in: European Conference on Computer Vision, Springer. pp. 541–556.

Lin, H., Yan, Y., Cheng, Q., 2023. Future role of artificial intelligence in advancing transportation electrification. Journal of Intelligent and Connected Vehicles .

Liu, Y., Jia, R., Ye, J., Qu, X., 2022. How machine learning informs ride-hailing services: A survey. Communications in Transportation Research 2, 100075.

Liu, Y., Wu, F., Liu, Z., Wang, K., Wang, F., Qu, X., 2023. Can language models be used for real-world urban-delivery route optimization? The Innovation 4.

Liu, Y., Zhang, J., Fang, L., Jiang, Q., Zhou, B., 2021. Multimodal motion prediction with stacked transformers. Computer Vision and Pattern Recognition .

Ma, C., Yu, C., Yang, X., 2021. Trajectory planning for connected and automated vehicles at isolated signalized intersections under mixed traffic environment. Transportation research part C: emerging technologies 130, 103309.

Mercat, J., Gilles, T., El Zoghby, N., Sandou, G., Beauvois, D., Gil, G.P., 2020. Multi-head attention for multi-modal joint vehicle motion forecasting, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 9638–9644.

Messaoud, K., Deo, N., Trivedi, M.M., Nashashibi, F., 2021. Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation, in: 2021 IEEE Intelligent Vehicles Symposium (IV), IEEE. pp. 165–170.

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 .

Mo, X., Xing, Y., Lv, C., 2020. 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. pp. 5057–5062.

Ngiam, J., Caine, B., Vasudevan, V., Zhang, Z., Chiang, H.T.L., Ling, J., Roelofs, R., Bewley, A., Liu, C., Venugopal, A., et al., 2021. Scene transformer: A unified multi-task model for behavior prediction and planning. arXiv e-prints , arXiv–2106.

Qu, X., Lin, H., Liu, Y., 2023. Envisioning the future of transportation: Inspiration of chatgpt and large models. Communications in Transportation Research 3.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788.

Shi, Q., Zhang, H., 2021. An improved learning-based lstm approach for lane change intention prediction subject to imbalanced data. Transportation research part C: emerging technologies 133, 103414.

Song, H., Luan, D., Ding, W., Wang, M.Y., Chen, Q., 2022. Learning to predict vehicle trajectories with model-based planning, in: Conference on Robot Learning, PMLR. pp. 1035–1045.

Sun, D., Chen, J., Mitra, S., Fan, C., 2022. Multi-agent motion planning from signal temporal logic specifications. IEEE Robotics and Automation Letters 7, 3451–3458.

Tengilimoglu, O., Carsten, O., Wadud, Z., 2023. Implications of automated vehicles for physical road environment: A comprehensive review. Transportation research part E: logistics and transportation review 169, 102989.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks, in: International Conference on Learning Representations.

Wang, K., Xiao, Y., He, Y., 2023a. Charting the future: Intelligent and connected vehicles reshaping the bus system. Journal of Intelligent and Connected Vehicles .

Wang, Y., Wei, H., Hu, B., Lv, C., 2023b. Robust estimation of vehicle dynamic state using a novel second-order fault-tolerant extended kalman filter. SAE International Journal of Vehicle Dynamics, Stability, and NVH 7, 301.

Wang, Y., Wei, H., Yang, L., Hu, B., Lv, C., 2023c. A review of dynamic state estimation for the neighborhood system of connected vehicles. SAE International Journal of Vehicle Dynamics, Stability, and NVH 7, 367.

Xing, Y., Lv, C., Cao, D., Velenis, E., 2021. Multi-scale driver behavior modeling based on deep spatial-temporal representation for intelligent vehicles. Transportation research part C: emerging technologies 130, 103288.

Yang, H., Du, L., Zhang, G., Ma, T., 2023. A traffic flow dependency and dynamics based deep learning aided approach for network-wide traffic speed propagation prediction. Transportation research part B: methodological 167, 99–117.

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

Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kümmerle, J., Königshof, H., Stiller, C., de La Fortelle, A., Tomizuka, M., 2020a. Interaction-dataset-based prediction challenge. http://challenge.interaction-dataset.com/prediction-challenge/intro.

Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kümmerle, J., Königshof, H., Stiller, C., de La Fortelle, A., Tomizuka, M., 2020b. Python scripts for the interaction dataset and the interpret competition. https://github.com/interaction-dataset/interaction-dataset.

Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., et al., 2019b. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint arXiv:1910.03088 .

Zhao, H., Gao, J., Lan, T., Sun, C., Sapp, B., Varadarajan, B., Shen, Y., Shen, Y., Chai, Y., Schmid, C., et al., 2021. Tnt: Target-driven trajectory prediction, in: Conference on Robot Learning, PMLR. pp. 895–904.

Zhao, T., Xu, Y., Monfort, M., Choi, W., Baker, C., Zhao, Y., Wang, Y., Wu, Y.N., 2019. Multi-agent tensor fusion for contextual trajectory prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 12126–12134.



https://dspace.org/

Heterogeneous graph social pooling for interaction-aware vehicle trajectory prediction

Mo, Xiaoyu; Xing, Yang; Lv, Chen

2024-11
Elsevier
https://doi.org/10.1016/j.tre.2024.103748
Mo X, Xing Y, Lv C. (2024) Heterogeneous graph social pooling for interaction-aware vehicle trajectory prediction. Transportation Research Part E: Logistics and Transportation Review, Volume 191, November 2024, Article number 103748

Downloaded from a DSpace repository. https://dspace.org/