Toward Heterogeneous Graph-based Imitation Learning for Autonomous Driving Simulation: Interaction Awareness and Hierarchical Explainability

MAHAN TABATABAIE, School of Computing, University of Connecticut, Storrs, United States
SUINING HE*, School of Computing, University of Connecticut, Storrs, United States
KANG SHIN, School of Computing, University of Michigan, Ann Arbor, United States
HAO WANG, School of Computing, University of Connecticut, Storrs, United States

Understanding and learning the actor-to-X interactions (AXIs), such as those between the focal vehicles (actor) and other traffic participants, such as other vehicles and pedestrians, as well as traffic environments like the city or road map, is essential for the development of a decision-making model and the simulation of autonomous driving. Existing practices on imitation learning (IL) for autonomous driving simulation, despite the advances in the model learnability, have not accounted for fusing and differentiating the heterogeneous AXIs in complex road environments. Furthermore, how to further explain the hierarchical structures within the complex AXIs remains largely under-explored.

To meet these challenges, we propose HGIL, an interaction-aware and hierarchically-explainable **H**eterogeneous **G**raph-based **I**mitation **L**earning approach for autonomous driving simulation. We have designed a novel heterogeneous interaction graph (HIG) to provide local and global representation as well as awareness of the AXIs. Integrating the HIG as the state embeddings, we have designed a hierarchically-explainable generative adversarial imitation learning approach, with local sub-graph and global cross-graph attention, to capture the interaction behaviors and driving decision-making processes. Our data-driven simulation and explanation studies based on the Argoverse v2 dataset (with a total of 40,000 driving scenes) have corroborated the accuracy (e.g., lower displacement errors compared to the state-of-the-art (SOTA) approaches) and explainability of HGIL in learning and capturing the complex AXIs.

Additional Key Words and Phrases: Interaction awareness, hierarchical explainability, heterogeneous graph fusion, imitation learning, autonomous driving simulation.

1 Introduction

Imitation learning (IL) for autonomous driving simulation aims to capture a cost function or a policy from the human driver demonstrations (e.g., real-world driving datasets) [3, 5, 6, 25, 30]. In the IL setting, the actor, i.e., the focal vehicle, interacts with various other traffic participants (e.g., other vehicles, pedestrians) as well as the traffic environments (e.g., map topology), forming the diverse scenes of the actor-to-X interactions (AXIs). These AXIs involve the behaviors of car following, l ane changing, cutting in when interacting with other vehicles and road contexts (e.g., closure and road work), as well as the responses to the presence of pedestrians (e.g., yielding

*Corresponding author.

Authors' Contact Information: Mahan Tabatabaie, School of Computing, University of Connecticut, Storrs, Connecticut, United States; e-mail: mahan.tabatabaie@uconn.edu; Suining He, School of Computing, University of Connecticut, Storrs, Connecticut, United States; e-mail: suining.he@uconn.edu; Kang Shin, School of Computing, University of Michigan, Ann Arbor, Michigan, United States; e-mail: kgshin@umich.edu; Hao Wang, School of Computing, University of Connecticut, Storrs, Connecticut, United States; e-mail: hao.3.wang@uconn.edu.

ACM 2833-0528/2024/12-ART
https://doi.org/10.1145/3708354

at the crosswalks). Understanding and learning such complex AXIs is essential for designing the decision-making models and simulation of autonomous driving systems.

Despite the recent IL advances [3, 4, 27, 41], existing studies have not accounted for the following two major designs that are critical for *interaction awareness* and *hierarchical explainability* toward an autonomous driving simulation framework:

(1) *How to differentiate heterogeneous AXIs for generalizing the contextual dependencies*: Learning the decision-making process of AXIs performed by the human drivers hinges on understanding the contextual dependencies between the actor (the focal vehicle) and other traffic participants as well as the traffic environments. However, the same human driver maneuver behaviors (e.g., turning or deceleration) may result from various *heterogeneous* contexts of AXIs. Existing feature representations such as simple feature vectorization [21], 2D rasterization [17, 19], and homogeneous graphs [18] of the actor's mobility features (e.g., motion information) and surrounding contexts (e.g., map information and topology) may not necessarily differentiate these AXIs, lowering the generalizability of the IL designs.

(2) *How to enable the hierarchical explanation of IL for autonomous driving simulation*: In the model simulation studies, understanding the *global* and *local* contexts of the human driver demonstrations hinges on tracing and dissecting the decisions of the actor. Specifically, responses to the *global contexts*, i.e., incoming general traffic conditions and map topological information (e.g., road work closure or highway exits), and those to the *local contexts*, i.e., the nearby traffic participants, can be highly interleaved and lead to complex AXI outcomes. Transparency requirements for autonomous driving simulation [20, 31] have established the needs of providing the hierarchical explainability to enable more trustworthy human-vehicle interactions [20], which, however, remains to be explored further in the IL designs.

To overcome the above-mentioned gaps, we propose HGIL, a novel **H**eterogeneous **G**raph-based **I**mitation **L**earning framework for interaction awareness and hierarchical explainability in autonomous driving simulation. Toward this framework, we have made the following three major contributions:

(a) **Heterogeneous Interaction Graph Fusion for AXIs**: We have designed a heterogeneous interaction graph (HIG) representation as the state embeddings of our imitation learning designs, characterizing the various objects involved in AXIs as the *nodes* and their interplay as the *edges*. To infuse the complex AXI scenes, we have derived within the HIG the *sub-graph structures*, which account for the heterogeneous interactions among the actor, other traffic participants such as other vehicles and pedestrians in our studies, and lane topology. This way, HGIL enhances its learnability over the existing IL approaches.

(b) **Hierarchically-Explainable IL Designs**: Based on the HIG fusion, we have further designed the hierarchical explanation designs for HGIL, via the *local sub-graph attention* and *global cross-graph attention* within the HIG. The proposed hierarchical explanation designs differentiate the contextual dependencies between the local and global observations, yielding the traceability of the decision-making process within the autonomous driving simulation.

(c) **Data-driven Simulation and Explanation Studies**: We have conducted extensive experimental studies on the Argoverse v2 dataset [34] with a total of 40,000 driving scenes to validate the accuracy and explainability of HGIL in learning and capturing driving behaviors for autonomous driving simulation. Our simulation results have demonstrated that our HGIL outperforms the other state-of-the-art approaches (including [1, 2, 10, 16, 24]) in terms of various displacement error measures (such as final displacement error), and achieves hierarchical explainability (in terms of sparsity and fidelity) regarding various AXIs.

The rest of the paper is organized as follows. We first review the related work in Sec. 2. Then, we present the HIG representation designs, and the problem formulation of HGIL in Sec. 3. Next, we present the core designs of our interaction-aware and hierarchically-explainable heterogeneous graph-based imitation learning in Sec. 4.

This is followed by the results of our experimental studies in Sec. 5, deployment discussion in Sec. 6, and the conclusion of this paper in Sec. 7.

2 Related Work

We briefly review two categories of prior work related to this paper.

2.1 Graph Representations for Motion Modeling

Prior motion modeling and planning studies for autonomous driving [17, 19, 21] considered vectorized feature encoding, such as 2-D rasterization of the bird's-eye view (BEV), of the vehicle's mobility features and surrounding contexts. However, existing 2-D rasterization, processed by feature convolution [19, 22], may not fully capture the interplay of the objects with the actor in the complex traffic scenes. Therefore, graph neural networks have recently attracted attention to model the relations of the objects in the traffic environments [18, 20, 31]. Jia et al. [12] and Zhang et al. [39] have considered the graph-based transformers for motion modeling. Zeng et al. [38] proposed a graph-based approach to incorporate the lane and map topology structures. Deo et al. [8] and Gilles et al. [9] have formulated a graph traversal problem for the motion modeling process. Tang et al. [31] studied the neural relation inference to generate the interactive behavior interpretation. Kosaraju et al. [14] implements graph attention networks based on BycleGAN [42] for multi-modal trajectory forecasting.

In addition, prior graph representation studies [36] often consider post-training and model-agnostic approaches to infer the interactions. Recent studies [5, 16, 38] investigated the interactions among different traffic participants. However, these designs often lack proper reasoning for their motion modeling processes and the subsequent simulation results. These designs often provide limited information about the importance of factors, rather than revealing detailed interactions.

Unlike these efforts, we have designed within HGIL the heterogeneous interaction graph (HIG) fusion, which provides the *hierarchical* characterization and explanation of the interactions and relations of the actor (the focal vehicle) with different types of traffic participants of the complex traffic scenes. HIG consists of the sub-graph structures, which accounts for the heterogeneous interactions among the actor, other traffic participants, and lane topology. This way, HGIL yields high learnability in the complex AXI scenes.

2.2 Imitation Learning (IL) for Autonomous Driving Simulation

Deep IL has recently been adopted for autonomous driving simulation and model development to capture the cost function or policy from the large-scale human driver demonstration data [3, 6, 16, 24, 29, 40]. Compared to the inverse reinforcement learning (IRL) that is usually expensive to run and difficult to scale [35], generative adversarial imitation learning (GAIL) [11] generates the policy without capturing the cost function, and is able to scale in the complex and spacious traffic environments. Zhou et al. [41] proposed a feedback synthesizer for data augmentation in IL to improve the autonomous driving performance in the unobserved environments. Bhattacharyya et al. [4] improved GAIL designs via a parameter sharing mechanism that enhances the generalizability to complex driving scenes. Lee et al. [15] leverages both positive (from expert) and negative (with collisions) demonstrations for fast convergence of the IL model.

Unlike the above-mentioned studies, the IL approach in HGIL provides a novel state embedding design based on HIG, which provides heterogeneous representability and hierarchical explainability. Our data-driven simulation studies have further corroborated our proposed designs in characterizing and explaining the complex AXIs. In addition, beyond the results in [27], we have conducted more model and sensitivity studies (e.g., over important model parameters) and explainability evaluations (based on the metrics of sparsity and fidelity) to corroborate the novel designs of HGIL.

3 Heterogeneous Interaction Graph Representation and Problem Formulation

We first present the representation designs of HIG in Sec. 3.1, followed by the important concepts and problem formulation in Sec. 3.2.

3.1 Heterogeneous Interaction Graph Representation

Toward interaction awareness and hierarchical explainability, we formulate the surrounding contexts of the actor (focal vehicle) at the t-th timestamp into a heterogeneous interaction graph (HIG). Each HIG consists of multiple *sub-graphs* that characterize the actor's local relations with the surrounding objects in different types of AXI scenes. All the sub-graphs share the node of the actor (the focal vehicle). Specifically, at each timestamp t, HGIL accounts for the *node features* of the actor as

$$\mathbf{V}_t^{(\mathrm{f})} = \left[x_t^{(\mathrm{f})}, y_t^{(\mathrm{f})}, v_t^{(\mathrm{f})}, \theta_t^{(\mathrm{f})}, \Delta x_t^{(\mathrm{f})}, \Delta y_t^{(\mathrm{f})} \right] \in \mathbb{R}^6, \tag{1}$$

where $x_t^{(\mathrm{f})}$, $y_t^{(\mathrm{f})}$, $v_t^{(\mathrm{f})}$, and $\theta_t^{(\mathrm{f})}$ correspond to the actor's position coordinates (unit: m), instantaneous speed (unit: m/s), and heading angle (unit: rad) in the global (earth) coordinate system under the bird's eye view (BEV). $\Delta x_t^{(\mathrm{f})} = x_t^{(\mathrm{f})} - x_{t-1}^{(\mathrm{f})}$ and $\Delta y_t^{(\mathrm{f})} = y_t^{(\mathrm{f})} - y_{t-1}^{(\mathrm{f})}$, respectively, denote the displacements of the actor w.r.t. the x and y axes from the preceding timestamp $t-1$.

In this prototype study, we take into account the following three types of sub-graphs within the HIG representation (illustrated in Fig. 1), while our HIG design is general enough to be extended to other types of AXIs given the availability of other interacting objects. HGIL determines the relations of the actor with other objects through the local sub-graph and global cross-graph attention mechanisms (detailed in Sec. 4.2).

3.1.1 Actor-to-Vehicle Sub-graph $\mathbf{G}_t^{(c)}$. We form $\mathbf{G}_t^{(\mathrm{c})}$ by including the actor and the peer vehicles within a range from the actor as the nodes (25m in our study). For each vehicle i of the K nearest peers observed ($i \in \{1, \ldots, K\}$), we find its node feature as

$$\mathbf{V}_{t,i}^{(\mathrm{c})} = \left[x_{t,i}^{(\mathrm{c})}, y_{t,i}^{(\mathrm{c})}, v_{t,i}^{(\mathrm{c})}, \theta_{t,i}^{(\mathrm{c})}, d_{t,i}^{(\mathrm{c})} \right] \in \mathbb{R}^5, \tag{2}$$

i.e., its global coordinates, speed, heading direction, as well as the distance (unit: m) from the actor. We let $\mathbf{V}_t^{(\mathrm{c})} \in \mathbb{R}^{K \times 5}$ be the node features of all the K nearest peer vehicles at the timestamp t. Let $\mathbf{E}_t^{(\mathrm{c})} \in \mathbb{R}^{(K+1)\times(K+1)}$ be the adjacency matrix representing the edges from the actor node to its peer vehicles at the timestamp t, where the elements in $\mathbf{E}_t^{(\mathrm{c})}$ are initialized as those for the edges between the actor and peer vehicle nodes, and zeros otherwise.

3.1.2 Actor-to-Pedestrian Sub-graph $\mathbf{G}_t^{(p)}$. Similar to $\mathbf{G}_t^{(\mathrm{c})}$, we form $\mathbf{G}_t^{(\mathrm{p})}$ that includes the pedestrians within a range (25m in our study) from the actor. We find the corresponding pedestrian node feature $j \in \{1, \ldots, P\}$ as

$$\mathbf{V}_{t,j}^{(\mathrm{p})} = \left[x_{t,j}^{(\mathrm{p})}, y_{t,j}^{(\mathrm{p})}, v_{t,j}^{(\mathrm{p})}, \theta_{t,j}^{(\mathrm{p})}, d_{t,j}^{(\mathrm{p})} \right] \in \mathbb{R}^5, \tag{3}$$

i.e., the global coordinates, velocity, heading direction, and distance of the pedestrian from the actor. We let $\mathbf{V}_t^{(\mathrm{p})} \in \mathbb{R}^{P \times 5}$ be the node features of all the P nearby pedestrians at the timestamp t. We similarly define $\mathbf{E}_t^{(\mathrm{p})} \in \mathbb{R}^{(P+1)\times(P+1)}$ as the adjacency matrix representing the edges from the actor node to the nearby pedestrians, where the elements in $\mathbf{E}_t^{(\mathrm{p})}$ are initialized as those for the edges between the actor and pedestrian nodes, and zeros otherwise.

3.1.3 Actor-to-Lane Sub-graph $\mathbf{G}_t^{(l)}$. To model the interaction between the actor and the map topology (e.g., when approaching the intersection or exit), we divide the road lane into multiple segments (of length 25.45m each on

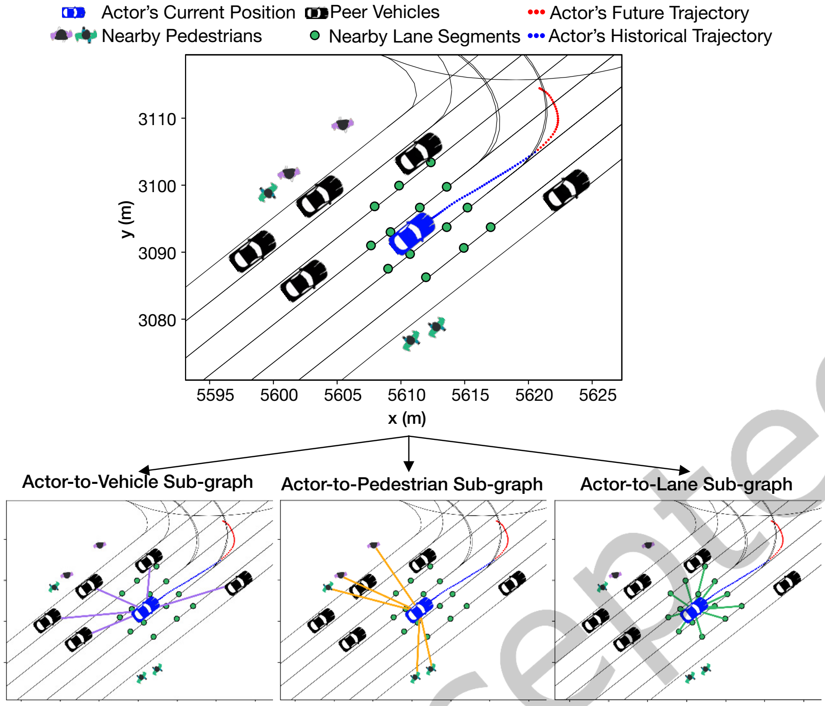

Fig. 1. Illustration of an HIG representation in HGIL.

average), and represent them by the nodes of a series of coordinates in the BEV. For each of the R closest road segment $m \in \{1, \dots, R\}$ within a range (10m in our study) from the actor, we find the lane node feature

$$\mathbf{V}_{t,m}^{(\mathrm{l})} = \left[x_{t,m}^{(\mathrm{l})}, y_{t,m}^{(\mathrm{l})}, d_{t,m}^{(\mathrm{l})}, e_{t,m}^{(\mathrm{l})}\right] \in \mathbb{R}^4, \tag{4}$$

i.e., the global coordinates, distance (unit:m) from the actor, and a binary variable $e_{t,m}^{(\mathrm{l})} \in \{1, 0\}$ indicating whether the road segment is part of an intersection ($e_{t,m}^{(\mathrm{l})} = 1$) or not. We let $\mathbf{V}_t^{(\mathrm{l})} \in \mathbb{R}^{R\times 4}$ be the lane node features of all the R nearby lane segments at the timestamp t. Similar to $\mathbf{E}_t^{(\mathrm{c})}$ and $\mathbf{E}_t^{(\mathrm{p})}$, we form the adjacency matrix for the nodes of the actor and the lane, i.e., $\mathbf{E}_t^{(\mathrm{l})} \in \mathbb{R}^{(R+1)\times(R+1)}$.

Given the above sub-graphs, we denote an HIG at a timestamp t as

$$\mathbf{G}_t = \left\{\mathbf{G}_t^{(\mathrm{c})}, \mathbf{G}_t^{(\mathrm{p})}, \mathbf{G}_t^{(\mathrm{l})}\right\}. \tag{5}$$

3.2 Concepts and Problem Formulation

3.2.1 State. In our IL setting with the infinite horizon, we formulate the state $\mathbf{S}_t$ of the actor (i.e., the focal vehicle as the agent) based on the historical HIGs for the past L timestamps, i.e.,

$$\mathbf{S}_t = \{\mathbf{G}_{t-L}, \mathbf{G}_{t-L+1}, \dots, \mathbf{G}_t\}. \tag{6}$$

Furthermore, without loss of generality, we can account for the focal vehicle as the actor, while the formulation is general enough to be extended to the multi-agent setting [4, 12].

3.2.2 Actions and Policy. Given an observed state $\mathbf{S}_t$, we aim to determine the decision process as well as the respective actions $\mathbf{A}_t$ of the actor that represents the focal vehicle. The IL designs of HGIL will identify the policy $\pi(\cdot)$, a function that maps the state $\mathbf{S}_t$ to its corresponding action $\mathbf{A}_t$,

$$\mathbf{A}_t \sim \pi(\mathbf{A}|\mathbf{S}_t). \tag{7}$$

In this prototype study, HGIL rolls out and generates a series of planned displacements toward the x and y axes,

$$\mathbf{A}_t = \left[\left(\Delta x^{(\mathrm{f})}_{(t+1)}, \Delta y^{(\mathrm{f})}_{(t+1)}\right), \ldots, \left(\Delta x^{(\mathrm{f})}_{(t+L)}, \Delta y^{(\mathrm{f})}_{(t+L)}\right)\right] \in \mathbb{R}^{L\times 2}, \tag{8}$$

for the future L timestamps.

3.2.3 Problem Definition. Given the above-mentioned states and actions from the human driver demonstration data, we formulate the generative adversarial imitation learning (GAIL) within HGIL to recover the focal vehicle's policy π that can be used to imitate the behaviors of the human drivers by generating $\mathbf{A}_t$, given its observed state $\mathbf{S}_t$.

Given the observed state $\mathbf{S}$ (say, the historical HIGs in Eq. (6)), the GAIL in HGIL optimizes the actor's policy π, such that the resulting actions $\mathbf{A}$ of the actor (i.e., series of planned displacements) are *indistinguishable* from the expert demonstrations (i.e., human driver demonstration). This can be formalized as finding a Nash equilibrium [11] within a minimax game between a policy generator network approximating π, and a discriminator network ψ, i.e.,

$$\min_{\pi} \max_{\psi} \mathbb{E}_{\mathbf{S},\mathbf{A}\sim\pi}[\log(\psi(\mathbf{S},\mathbf{A}))] + \mathbb{E}_{\mathbf{S},\mathbf{A}\sim\pi^{\mathrm{e}}}[\log(1-\psi(\mathbf{S},\mathbf{A}))], \tag{9}$$

where ψ represents the policy discriminator network function of the GAIL and π^{e} denotes the policy of the expert (i.e., human drivers). To further expand the interaction awareness and hierarchical explainability, we design the state embeddings with HIGs for $\mathbf{S}$ (detailed in Sec. 4).

4 Interaction-Aware and Hierarchically-Explainable IL Designs

We first overview the state-embedding processing in Sec. 4.1, then present the state embeddings with HIGs in Sec. 4.2, and finally provide the training design in Sec. 4.3.

4.1 Overview of State Embedding Processing

We overview the state embedding processing of HGIL in Fig. 2, which consists of (I) local sub-graph attention and (II) global cross-graph attention. Specifically, HGIL first creates the HIGs to represent the actor's state in the traffic environment at each timestamp. Then, the local sub-graph attention in HGIL updates the node features of each sub-graph by accounting for the local interactions and relations of the objects involved. Next, HGIL fuses the resulting node features from the HIGs, and further leverages the global cross-graph attention to quantify the actor's interactions in a global context, and generates the state embeddings for policy learning (detailed in Sec. 4.3).

4.2 State Embeddings with HIGs

4.2.1 Local Sub-graph Attention. The human driver may respond to traffic participants and environments with different strategies. In order to capture the interactions between the actor with different objects and the resulting AXI scenes, we design the *local sub-graph attention* for our IL settings, which helps identify the important sub-graphs within our HIG that concern the decision-making process of the actor.

(a) Node Feature Embeddings: Given the set of the node features of the actor and all the sub-graphs for the t-th timestamp,

$$\mathbf{V}_t = \left\{\mathbf{V}^{(f)}_t, \mathbf{V}^{(c)}_t, \mathbf{V}^{(p)}_t, \mathbf{V}^{(l)}_t\right\}, \tag{10}$$

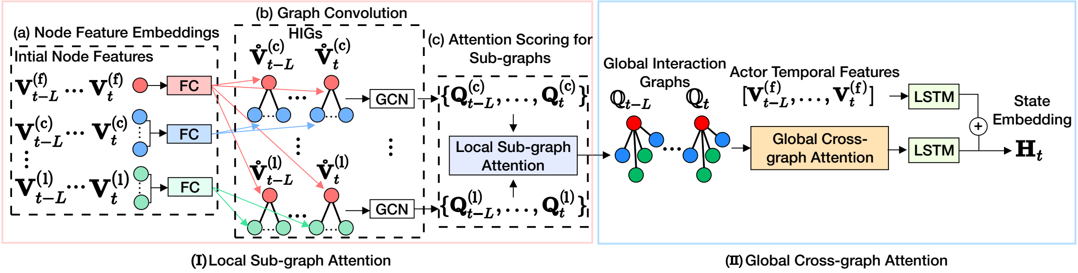
(a) Node Feature Embeddings
Intial Node Features
$\mathbf{V}_{t-L}^{(\mathrm{f})} \cdots \mathbf{V}_t^{(\mathrm{f})}$
$\mathbf{V}_{t-L}^{(\mathrm{c})} \cdots \mathbf{V}_t^{(\mathrm{c})}$
$\mathbf{V}_{t-L}^{(\mathrm{l})} \cdots \mathbf{V}_t^{(\mathrm{l})}$
FC
FC
FC
(b) Graph Convolution
HIGs
$\mathring{\mathbf{V}}_{t-L}^{(\mathrm{c})}$
$\mathring{\mathbf{V}}_t^{(\mathrm{c})}$
$\mathring{\mathbf{V}}_{t-L}^{(\mathrm{l})}$
$\mathring{\mathbf{V}}_t^{(\mathrm{l})}$
GCN
GCN
(c) Attention Scoring for Sub-graphs
$\{\mathbf{Q}_{t-L}^{(\mathrm{c})}, \ldots, \mathbf{Q}_t^{(\mathrm{c})}\}$
Local Sub-graph Attention
$\{\mathbf{Q}_{t-L}^{(\mathrm{l})}, \ldots, \mathbf{Q}_t^{(\mathrm{l})}\}$
Global Interaction Graphs
$\mathbb{Q}_{t-L}$
$\mathbb{Q}_t$
Actor Temporal Features
$[\mathbf{V}_{t-L}^{(\mathrm{f})}, \ldots, \mathbf{V}_t^{(\mathrm{f})}]$
LSTM
Global Cross-graph Attention
LSTM
+
State Embedding
$\mathbf{H}_t$
(I) Local Sub-graph Attention
(II) Global Cross-graph Attention

Fig. 2. Architecture overview of state embeddings within HGIL, which consists of (I) local sub-graph attention and (II) global cross-graph attention.

we first process each node feature in $\mathbf{V}_t$ with an independent fully-connected (FC) layer with B_1 hidden units and the LeakyReLU activation function, to convert them to the B_1-dimensional feature space. This way, we obtain the set of the node embeddings:

$$\overline{\mathbf{V}}_t = \left\{ \overline{\mathbf{V}}_t^{(\mathrm{f})}, \overline{\mathbf{V}}_t^{(\mathrm{c})}, \overline{\mathbf{V}}_t^{(\mathrm{p})}, \overline{\mathbf{V}}_t^{(\mathrm{l})} \right\}, \tag{11}$$

where $\overline{\mathbf{V}}_t^{(\mathrm{f})} \in \mathbb{R}^{1 \times B_1}$, $\overline{\mathbf{V}}_t^{(\mathrm{c})} \in \mathbb{R}^{K \times B_1}$, $\overline{\mathbf{V}}_t^{(\mathrm{p})} \in \mathbb{R}^{P \times B_1}$, $\overline{\mathbf{V}}_t^{(\mathrm{l})} \in \mathbb{R}^{R \times B_1}$.

Then, we concatenate the actor node feature $\overline{\mathbf{V}}_t^{(\mathrm{f})}$ with each of the sub-graph node feature embeddings, and obtain the node features of the sub-graphs.

(b) Graph Convolution: We then process each concatenated feature with a separate graph convolutional (GCN) layer (with a total of B_2 hidden units) to account for the local interaction within each of the sub-graphs, resulting in the updated node features $\mathbf{Q}_t^{(\mathrm{c})} \in \mathbb{R}^{(K+1) \times B_2}$, $\mathbf{Q}_t^{(\mathrm{p})} \in \mathbb{R}^{(P+1) \times B_2}$, and $\mathbf{Q}_t^{(\mathrm{l})} \in \mathbb{R}^{(R+1) \times B_2}$.

For instance, to find $\mathbf{Q}_t^{(\mathrm{c})}$, we concatenate the peer vehicles' node features, $\overline{\mathbf{V}}_t^{(\mathrm{c})}$, with the actor node features, $\overline{\mathbf{V}}_t^{(\mathrm{f})}$, i.e.,

$$\mathring{\mathbf{V}}_t^{(\mathrm{c})} = \left[\overline{\mathbf{V}}_t^{(\mathrm{c})} \middle\| \overline{\mathbf{V}}_t^{(\mathrm{f})} \right]. \tag{12}$$

We then further feed it to the GCN layer, i.e.,

$$\mathbf{Q}_t^{(\mathrm{c})} = \left(\hat{\mathbf{D}}^{(\mathrm{c})}\right)^{-\frac{1}{2}} \cdot \left(\mathbf{E}_t^{(\mathrm{c})} + \mathbf{I}\right) \cdot \left(\hat{\mathbf{D}}^{(\mathrm{c})}\right)^{-\frac{1}{2}} \cdot \mathring{\mathbf{V}}_t^{(\mathrm{c})} \cdot \mathbf{W}^{(\mathrm{c})} + \mathbf{b}^{(\mathrm{c})}, \tag{13}$$

where $\hat{\mathbf{D}}^{(\mathrm{c})} \in \mathbb{R}^{(K+1) \times (K+1)}$ represents the diagonal degree matrix, i.e.,

$$\hat{\mathbf{D}}^{(\mathrm{c})}[i, i] = \sum_j \mathbf{E}_t^{(\mathrm{c})}[i, j], \tag{14}$$

where $(\mathbf{E}_t^{(\mathrm{c})} + \mathbf{I})$ adds the self-loops to the graph. $\mathbf{W}^{(\mathrm{c})} \in \mathbb{R}^{B_2 \times B_2}$ and $\mathbf{b}^{(\mathrm{c})} \in \mathbb{R}^{B_2}$ represent the trainable weights. We similarly find $\mathbf{Q}_t^{(\mathrm{p})} \in \mathbb{R}^{(P+1) \times B_2}$ and $\mathbf{Q}_t^{(\mathrm{l})} \in \mathbb{R}^{(R+1) \times B_2}$ with two separate GCN layers.

(c) Attention Scoring for Sub-graphs: We then quantify the importance of different sub-graphs based on the graph embeddings. Specifically, as illustrated in Fig. 3, we first concatenate the actor node's features within the resulting graph embeddings from the three GCN operations into a vector $\mathbf{Q}_t^{(\mathrm{f})}$, i.e.,

$$\mathbf{Q}_t^{(\mathrm{f})} = \left[\mathbf{Q}_t^{(\mathrm{c})}[-1, :] \middle\| \mathbf{Q}_t^{(\mathrm{p})}[-1, :] \middle\| \mathbf{Q}_t^{(\mathrm{l})}[-1, :] \right], \tag{15}$$

where $\mathbf{Q}_t^{(\mathrm{c})}[-1, :]$, $\mathbf{Q}_t^{(\mathrm{p})}[-1, :]$, and $\mathbf{Q}_t^{(\mathrm{l})}[-1, :]$ correspond to the embedded features of the actor node (i.e., the last row) w.r.t. actor-to-vehicle, actor-to-pedestrian, and actor-to-lane sub-graphs.

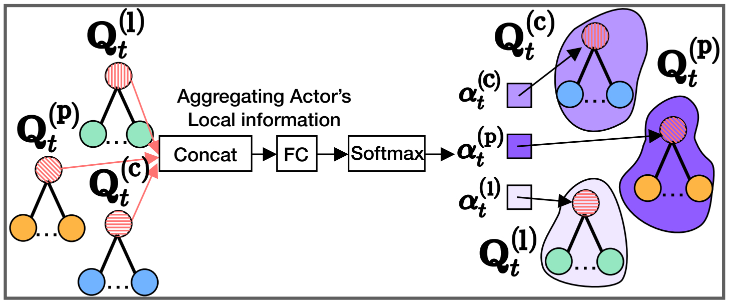

Fig. 3. Illustration of the attention scoring for sub-graphs.

In other words, the vector $\mathbf{Q}_t^{(\mathrm{f})} \in \mathbb{R}^{1\times B_2'}$ ($B_2' = 3B_2$) aggregates the *local context* of different objects near the actor, and can be further leveraged to determine and differentiate the relative importance of the sub-graphs in the AXIs.

We then feed $\mathbf{Q}_t^{(\mathrm{f})}$ to two FC layers with the B_3 hidden units to generate the sub-graph attention scores

$$\alpha_t = \left[\alpha_t^{(\mathrm{c})}, \alpha_t^{(\mathrm{p})}, \alpha_t^{(\mathrm{l})}\right] = \rho\left(\mathtt{FC}\left(\sigma\left(\mathtt{FC}(\mathbf{Q}_t^{(\mathrm{f})})\right)\right)\right) \in \mathbb{R}^3, \tag{16}$$

where $\sigma(\cdot)$ represents the LeakyReLU activation function and $\rho(\cdot)$ is the Softmax function. Each of the three elements in α_t represents the level of interaction between the actor and each of the sub-graphs.

4.2.2 Global Cross-graph Attention. To capture the human driver decisions in joint response to different involved objects (e.g., other traffic participants, map topology) in the global contexts of the traffic environments, we have also designed the *global cross-graph attention* to capture the global interplay in the AXIs.

Recall that $\mathbf{Q}_t^{(\mathrm{c})}[-1,:]$, $\mathbf{Q}_t^{(\mathrm{p})}[-1,:]$, and $\mathbf{Q}_t^{(\mathrm{l})}[-1,:]$ refer to the embedded features of the actor node (i.e., the last row) w.r.t. the three sub-graphs. We further update the actor node features from Eq. (15) by multiplying the sub-graph attention scores with the corresponding actor node features (i.e., the last row) in the sub-graphs, i.e.,

$$\overline{\mathbf{Q}}_t^{(\mathrm{f})} = \left(\alpha_t^{(\mathrm{c})} \cdot \mathbf{Q}_t^{(\mathrm{c})}[-1,:]\right) \oplus \left(\alpha_t^{(\mathrm{p})} \cdot \mathbf{Q}_t^{(\mathrm{p})}[-1,:]\right) \oplus \left(\alpha_t^{(\mathrm{l})} \cdot \mathbf{Q}_t^{(\mathrm{l})}[-1,:]\right),$$

where $\oplus$ denotes the element-wise addition operation.

To find the global cross-graph attention, for each timestamp t, we fuse all the sub-graph nodes and their edges into a *global interaction graph*, denoted as $\mathbb{G}_t$, that consists of $T = 1 + K + P + R$ nodes in total. We form the global node feature embeddings of $\mathbb{G}_t$ by concatenating the updated actor node feature $\overline{\mathbf{Q}}_t^{(\mathrm{f})}$ with those of all other nodes, i.e.,

$$\mathbb{Q}_t = \left[\overline{\mathbf{Q}}_t^{(\mathrm{f})} \| \mathbf{Q}_t^{(\mathrm{c})}[1:K,:] \| \mathbf{Q}_t^{(\mathrm{p})}[1:P,:] \| \mathbf{Q}_t^{(\mathrm{l})}[1:R,:]\right], \tag{17}$$

where $\mathbb{Q}_t \in \mathbb{R}^{T\times B_2}$.

We then model the levels of interactions at the timestamp t, denoted as $\Gamma_t \in \mathbb{R}^{T\times T}$, across all the nodes in the global interaction graph $\mathbb{G}_t$, where the level of interaction between each pair of nodes is quantified by the attention score of

$$\Gamma_t[i,j] = \frac{\exp(\mu_t[i,j])}{\sum_{o=1}^{T}\exp(\mu_t[i,o])}, \tag{18}$$

and $\mu_t[i,j]$ is given by

$$\mu_t[i,j] \triangleq (\mathbf{W}_v)^\top \cdot \sigma\left((\mathbb{Q}_t[i,:] \cdot \mathbf{W}_g) \,\|\, (\mathbb{Q}_t[j,:] \cdot \mathbf{W}_g)\right).$$

Here $\sigma(\cdot)$ represents the LeakyReLU activation function, and $\mathbf{W}_v \in \mathbb{R}^{B_3'}$ ($B_3' = 2B_2$) and $\mathbf{W}_g \in \mathbb{R}^{B_3\times B_3}$ represent the trainable parameter matrices.

Then, we generate the weighted node embeddings $\mathbf{F}_t \in \mathbb{R}^{T \times B_3}$ based on the following linear operation,

$$\mathbf{F}_t = \Gamma_t \cdot \mathbf{W}_g + \mathbf{b}_g, \tag{19}$$

where $\mathbf{W}_g \in \mathbb{R}^{T \times B_3}$ and $\mathbf{b}_g \in \mathbb{R}^{B_3}$ are trainable parameters.

Recall that each observed state is given by a series of HIGs, i.e., $\mathbf{S}_t = \{\mathbf{G}_{t-L}, \mathbf{G}_{t-L+1}, \dots, \mathbf{G}_t\}$. For the timestamps from $(t-L)$ to t, HGIL finds the node embeddings of the global interaction graphs $\mathbb{G}_{t-L}$ to $\mathbb{G}_t$, i.e., $\mathbf{F}_{t-L}$ to $\mathbf{F}_t$. We feed the corresponding actor node feature embeddings (i.e., the last row of each $\mathbf{F}_t$) from the L historical timestamps to the long short-term memory (LSTM) with the LealyReLU activation function. Then, we obtain the sequence embeddings of the global interaction graphs, i.e.,

$$\mathbf{H}'_t = \texttt{LSTM}\left([\mathbf{F}_{(t-L)}[-1,:], \dots, \mathbf{F}_t[-1,:]]\right). \tag{20}$$

The sequence embeddings from the global interaction graphs are added with the temporal feature embeddings of the actor node features generated by another LSTM module, i.e.,

$$\mathbf{H}_t = \mathbf{H}'_t \oplus \texttt{LSTM}\left([\mathbf{V}^{(\mathrm{f})}_{t-L}, \dots, \mathbf{V}^{(\mathrm{f})}_t]\right), \tag{21}$$

which forms the final state embeddings $\mathbf{H}_t \in \mathbb{R}^{B_4}$ for the training of HGIL (detailed in Sec. 4.3).

4.3 Training Designs of HGIL

In what follows, we present the training designs of HGIL.

4.3.1 Policy Generator and Discriminator Networks. Fig. 4 illustrates the model training process given the state embeddings $\mathbf{H}_t$. Based on the state embeddings, HGIL provides a policy generator network consisting of FC layers to approximate and generate the policy π that resembles the decision-making process of the human drivers. In the meantime, HGIL provides the policy discriminator network ψ to distinguish the actions performed (i.e., trajectories) by the policy generator network against the human driver demonstration data (i.e., expert action from the demonstration). We show the structures of the two networks in Fig. 5.

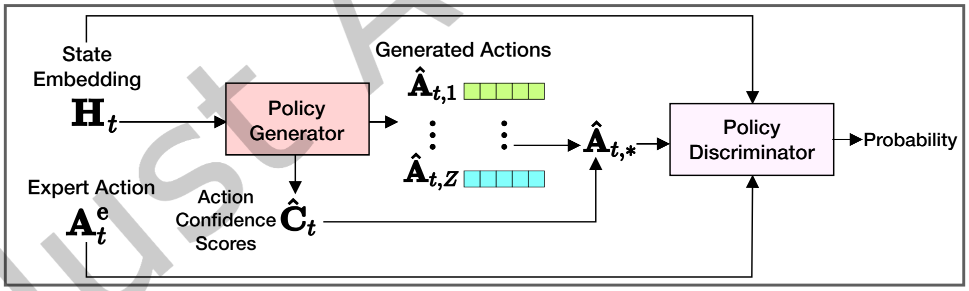

Fig. 4. Illustration of the policy learning designs in HGIL.

(a) The policy generator network takes in the state embeddings of the actor $\mathbf{H}_t$, and returns a set of Z possible sequences of displacement actions, $\{\hat{\mathbf{A}}_{t,i}\}$ ($i \in \{1, \dots, Z\}$), through the fully-connected (FC) network. Here we take into account multiple sequences of displacement actions to accommodate the *decision uncertainty* of motion planning in the autonomous driving simulation. To this end, the policy generator network outputs the confidence score $\hat{\mathbf{C}}_t \in \mathbb{R}^Z$ (in terms of probability) for each of $\{\hat{\mathbf{A}}_{t,i}\}$.

(b) The policy discriminator network ψ aims to discriminate the actions generated from the policy generator as well as the human driver demonstration (expert). ψ takes in (i) the policy generator's output actions (say, $\hat{\mathbf{A}}_{t,*}$ that

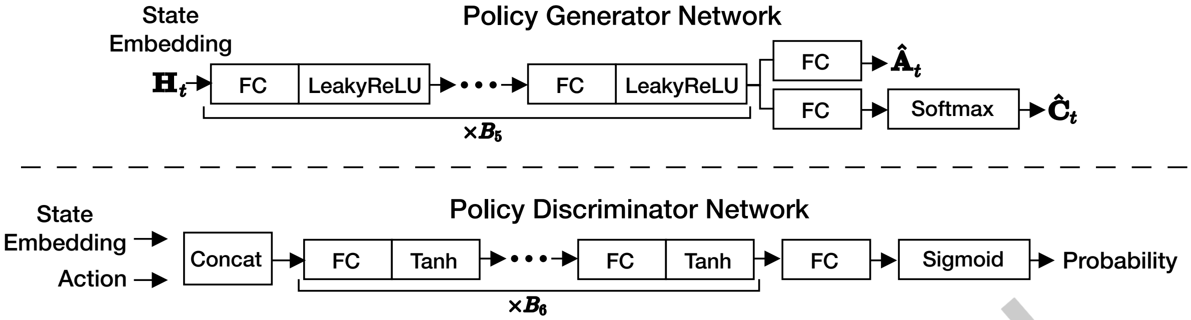

Fig. 5. Structures of the policy generator and discriminator networks.

corresponds to the maximum confidence score in $\hat{\mathbf{C}}_t$); or (ii) the actual actions $\mathbf{A}_t^e$ performed in the human driver demonstration data. Thus, given the concatenation of input actions ($\mathbf{A}_{t,*}$ or $\mathbf{A}_t^e$) as well as state embeddings $\mathbf{H}_t$, ψ estimates the probability (i.e., $\psi\left(\left[\mathbf{H}_t \| \mathbf{A}_t^e\right]\right)$ or $\psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)$) that the input action resembles the human driver demonstration.

4.3.2 Model Training Loss. In order to capture the discrepancy between the generated actions and the human driver demonstration, we consider the following types of loss within HGIL, i.e., (a) displacement regression loss ℓ_r and (b) confidence cross-entropy loss ℓ_c. We integrate them within the training loss of HGIL, i.e., (i) policy generator network loss L_g and (ii) discriminator network loss L_d.

(a) Displacement Regression Loss ℓ_r: The displacement regression loss ℓ_r is given by the mean squared error (MSE) between the generated sequence of actions (i.e., a series of planned displacements) with the highest score (probability) in $\hat{\mathbf{C}}_t$, denoted as $\hat{\mathbf{A}}_{t,*}$, and the actual action in the human driver demonstration, i.e.,

$$\ell_r \triangleq \frac{1}{Z}\sum_{i=1}^{Z}\left(\hat{\mathbf{A}}_{t,i} - \mathbf{A}_{t,i}\right)^2 . \tag{22}$$

We here leverage ℓ_r to generate the state embeddings before the adversarial optimization of the entire network [37].

(b) Confidence Cross-Entropy Loss ℓ_c: We define a one-hot encoding vector as a label for the confidence scores, $\mathbf{B}_t \in \mathbb{R}^Z$, to indicate the set of actions among all generated ones that is the closest to the human driver demonstration. For instance, we denote $\mathbf{B}_t = [0, 1, 0, \ldots, 0]$, if the second set of the generated actions has the least Euclidean distance from $\mathbf{A}_t$ in the human driver demonstration. Based on the above, we find the cross-entropy loss ℓ_c between the generated actions and the human driver demonstrations, i.e.,

$$\ell_c \triangleq -\sum_{i=1}^{Z}\left(\mathbf{B}_t[i] \cdot \log\left(\hat{\mathbf{C}}_t[i]\right)\right). \tag{23}$$

Based on the above designs, we have the loss in the policy generator and discriminator networks as follows.

(i) Policy Generator Loss L_g: In order to train the policy generator network, we integrate the regression loss ℓ_r and confidence loss ℓ_s to account for the discrepancy between the actions performed by the actor and the human driver demonstration. In the meantime, based on the formulation in Eq. (9), HGIL maximizes the probability $\psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)$ (i.e., minimize $1 - \psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)$) such that the discriminator network cannot discriminate the actions generated by the generator network from those of the human driver demonstration.

In summary, the policy generator minimizes

$$\mathrm{L}_g \triangleq \beta_r \cdot \ell_r + \beta_c \cdot \ell_c + \beta_d \cdot \log\left(1 - \psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)\right), \tag{24}$$

where β_r, β_c, and β_d represent the corresponding weights.

(ii) Policy Discriminator Loss L_d: Based on the formulation in Eq. (9), the policy discriminator network further performs the opposite optimization against the generator, by maximizing

$$\mathrm{L}_d \triangleq \log\left(\psi\left(\left[\mathbf{H}_t \middle\| \mathbf{A}^e_t\right]\right)\right) + \log\left(1 - \psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)\right). \tag{25}$$

Since there is a minimax game between the policy generator and discriminator networks [11], we train them iteratively based on Eqs. (24) and (25) until convergence.

5 Data-driven Model Emulation Studies

We first review the imitation learning-based baseline approaches used for performance comparison in Sec. 5.1. Then, we provide the details of the simulation settings and the network parameters in Sec. 5.2 followed by the experimental results and data visualization in Sec. 5.3.

5.1 Baseline Approaches

We compare HGIL with the following baseline and state-of-the-art approaches on IL for autonomous driving simulation.

(1) DualDisc [2]: which implements a spatio-temporal model (along with the conventional long short-term memory) based on dual-discriminator GAIL.
(2) DualDisc-GRU: which adapts DualDisc [2] based on the gated recurrent unit (GRU) to capture the spatio-temporal correlations.
(3) DualDisc-BiLSTM: which adapts DualDisc [2] based on the bidirectional long short-term memory (BiLSTM) to capture the spatio-temporal correlations.
(4) CGAIL [16, 24]: which adopts and adapts the conditional GAIL for trajectory prediction.
(5) SocialGAN [1, 10]: which integrates the social pooling operation [1] with GAIL.
(6) LaneGCN-GAIL [18]: which implements a graph neural network architecture based on GAIL.
(7) HGAIL [5]: which provides the hierarchical model-based GAIL.
(8) SeqST-GAN [33]: which implement a sequence-to-sequence model based on the recurrent neural networks and the generative adversarial networks.

5.2 Simulation Settings

5.2.1 Dataset Studied & Performance Metrics. We leverage the large-scale human driver demonstration dataset Argoverse v2 [34] for our experimental studies. Specifically, we select 35,000 driving scenes for IL training and 5,000 scenes for evaluation.

We evaluate the effectiveness of HGIL and other baselines in learning the human driving behaviors based on final displacement error (i.e., distance of the final generated position from the true position in the demonstration; denoted as FDE) and average displacement error (i.e., average distance of all locations in the generated and actual actions; denoted as ADE). We also find the minimum final displacement error (minFDE) and the minimum average displacement error (minADE) that represent the errors of the actions with the lowest FDE/ADE. We also find the miss rate (MR) regarding the percentage of all scenes when minFDE is over 2m.

5.2.2 Model Parameter Settings. Unless otherwise stated, we use the following parameters by default. Since the Argoverse v2 dataset is collected with a 10Hz frequency, we set $L = 30$ to leverage 3s of historical information to generate next 3s of actions. Like the prior studies [18, 38], we set $Z = 6$, i.e., 6 sets of candidate actions given an observed state $\mathbf{S}_t$, and estimate their uncertainty based on the confidence score $\hat{\mathbf{C}}_t \in \mathbb{R}^6$. For the local sub-graph and global interaction attention components, we use an FC layer with $B_1 = 64$ units to convert the node features.

Furthermore, we set the number of the hidden units of all the subsequent graph layers to $B_2 = B_3 = 64$. We set the number of the hidden units for the LSTM modules to $B_4 = 32$ to generate the state embeddings. Besides, we leverage $B_5 = B_6 = 2$ FC layers in each of the policy generator and discriminator networks (Fig. 5), and each FC layer is with 32 hidden units. We set $\beta_s = \beta_d = 1$ and $\beta_c = 0.3$ in Eq. (24).

5.2.3 Simulation Environment & Model Training Setup. Our networks are implemented on Pytorch 1.13.1 and Python 3.8.16. We performed experiments on an HPC server equipped with Linux Ubuntu 18.04.5 LTS, an AMD Ryzen Threadripper 3960X 24-Core CPU, 4×GeForce RTX 3090 with GDDR5 24GB, and 128GB RAM. With these settings, the training of our HGIL took an average of 361.3ms per AXI scene (each driving scene lasts for 6s on average).

HGIL is trained as follows. We first pre-train the policy generator network with the learning rate decay (from 0.01 to 0.001) for 300 iterations (Adam optimizer is adopted for this). We then train the policy generator and discriminator networks according to the Eqs. (24) and (25) with a learning rate of 0.001 for 200 iterations. At each iteration, we sample 1,000 driving scenes from the dataset and train the networks. We note that HGIL is overall efficient, with a total of 71,473 model parameters, average training time per sample as 5.506ms, and average inference time per sample as 4.486ms based on our computing platform.

Table 1. Overall performance and evaluation results of all approaches.

Model	**FDE**	**ADE**	**minFDE**	**minADE**	**MR**
HGIL	**2.88**	**1.19**	**2.43**	**1.02**	**23%**
DualDisc	3.77	1.83	3.95	1.92	41%
DualDisc-GRU	3.73	1.61	3.13	1.41	42%
DualDisc-BiLSTM	3.69	1.59	2.97	1.37	38%
CGAIL	3.11	1.33	2.71	1.15	28%
SocialGAN	3.07	1.29	2.71	1.18	30%
LaneGCN-GAIL	3.01	1.26	2.60	1.10	27%
HGAIL	3.45	1.42	2.91	1.22	27%
SeqST-GAN	3.54	1.46	3.01	1.25	29%

5.3 Performance Evaluation Results and Case Studies

5.3.1 Overall Performance. We present the overall performance of HGIL in Table 1, and compare HGIL with other IL-based methods. HGIL is observed to outperform the other baselines in learning the human driving behaviors in the AXIs. In particular, our HGIL achieves 18.79%, 23.84%, 23.41%, 29.90%, and 42.39% lower in terms of FDE, ADE, minFDE, minADE, and MR on average compared with the baseline approaches. DualDisc (as well as the variations of DualDisc-GRU and DualDisc-BiLSTM), CGAIL, SocialGAN, and SeqST-GAN may not account for the complex AXIs in the traffic scenes, and hence lead to lower accuracy as the actor (the focal vehicle) actively interacts with other traffic participants and environments. We can also observe that inclusion of the bidirectional long short-term memory helps improve the performance compared with the conventional sequence learning for DualDisc due to enhanced learnability on the spatio-temporal correlations. While LaneGCN-GAIL accounts for the map topology and HGAIL aims to understand the hierarchy of the interactions, their interaction designs may not further differentiate other traffic participants and their global and local contexts. Using the local sub-graph attention and the global cross-graph attention, HGIL achieves better performance in learning the human drivers. In particular, HGIL has achieved more than 4.32%, 5.56%, 6.54%, 7.27%, and 14.81% performance improvements in terms of FDE, ADE, minFDE, minADE, and MR compared with LaneGCN-GAIL, demonstrating the effectiveness of our hierarchical graph designs.

5.3.2 Model Ablation Studies. Table 2 presents the results of our model ablation studies on HGIL that evaluate the importance of different designs. Specifically, we compare the performance of complete HGIL designs with the following variations: w/o HIG, w/o the local sub-graph attentions, w/o the global cross-graph attention, and w/o the edge weights of AXIs.

We can see that the highest performance drop is caused by removing the global cross-graph and local sub-graph attention mechanisms. This implies the importance of the local sub-graph attention and global cross-graph attention in learning and capturing the human driving behaviors. In addition, we can also observe that that relying upon temporal information only without the HIGs degrades the performance. This demonstrates the necessity of the HIGs for capturing the interactions within the complex traffic environment.

Table 2. The results of our ablation studies of HGIL.

Variations	FDE	ADE	minFDE	minADE	MR
HGIL	**2.88**	**1.19**	**2.43**	**1.02**	**23%**
HGIL w/o HIG	3.05	1.32	2.73	1.20	29%
HGIL w/o Local	3.22	1.39	2.82	1.24	30%
HGIL w/o Global	3.83	1.74	3.29	1.52	39%
HGIL w/o Edge Weights	3.59	1.62	3.36	1.53	43%

5.3.3 Sensitivity Studies. We have also evaluated the sensitivity of the important parameters of HGIL. Fig. 6 illustrates our results in terms of FDE; Fig. 6(a) shows the FDE vs. the hidden units for the graph operations in the local sub-graph attention (denoted as B_2). The performance is found to start to decrease after 64 due mainly to the fitting over the complex and potentially noisy traffic scenes.

We can observe a similar trend in Figs. 6(b) and 6(c) for the number of the hidden units used in the global cross-graph attention (denoted as B_3) as well as that of the hidden units for the LSTM layers (denoted as B_4). So, we adopt $B_2 = B_3 = 64$ and $B_4 = 32$ to balance between the model learnability and generalizability.

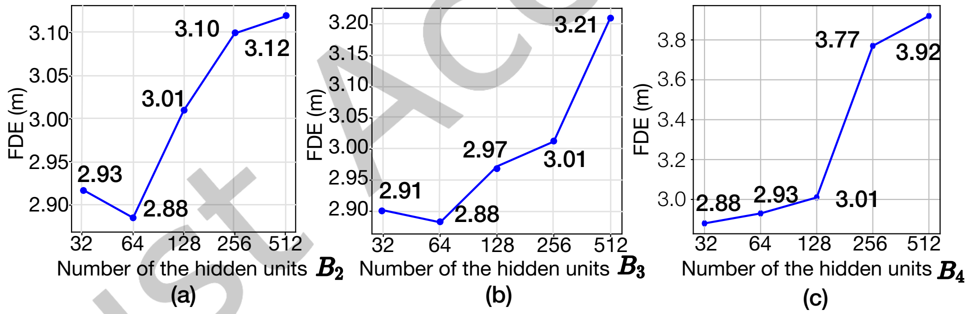

Fig. 6. Model parameter sensitivities in HGIL.

5.3.4 Model Explanationability. We have further conducted explainability studies on HGIL based on measures of *sparsity* and *fidelity* [23]. Specifically, we measure the sparsity as the ratio of the number of the graph nodes in each HIG i that have been identified as important by HGIL (i.e., with the attention score greater than a certain threshold of 0.7), denoted as m_i, over the total number of nodes in the HIG, M_i. The average sparsity of all N HIGs from all driving scenes is then given by

$$\text{sparsity} \triangleq \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{m_i}{M_i}\right). \tag{26}$$

This quantifies the explainability of HGIL in differentiating the interactions.

In addition, we find the fidelity that characterizes the performance drop when the nodes of the HIGs with high attention scores (say, above 0.7) are removed. Specifically, we measure the average percentage of drops in terms of the final displacement errors (FDEs), i.e.,

$$\text{fidelity} \triangleq \frac{1}{N}\sum_{i=1}^{N}\frac{|\text{FDE}'_i - \text{FDE}_i|}{\text{FDE}_i}, \tag{27}$$

where FDE_i and FDE'_i, respectively, represent the final displacement errors with and without the nodes with high attention scores. The fidelity represents the model explainability in terms of capturing the essential AXIs toward improved performance.

We illustrate the explanation quantification results in Fig 7. In terms of sparsity, we evaluate percentage of graph nodes (in local sub-graph attention and global cross-graph attention) that have been identified as important. In terms of fidelity, we evaluate the performance of drop of HGIL given removal of nodes in local sub-graph attention and global cross-graph attention in HIGs. Higher sparsity and fidelity indicate a model's explainability. Both local sub-graph and global cross-graph attentions (denoted as "local" and "global") have high sparsity and fidelity values, indicating that HGIL captures and differentiates more important nodes for AXIs. We have also shown the sparsity and fidelity values by SuperGAT [13] and AGNN (based on the conventional graph attention [32]), and HGIL is found to outperform them with higher quantified explainability. Furthermore, we can observe that the local sub-graph attention has an even higher fidelity value, implying the more importance of the interactions identified by the local sug-graph attention that deals with the IL performance.

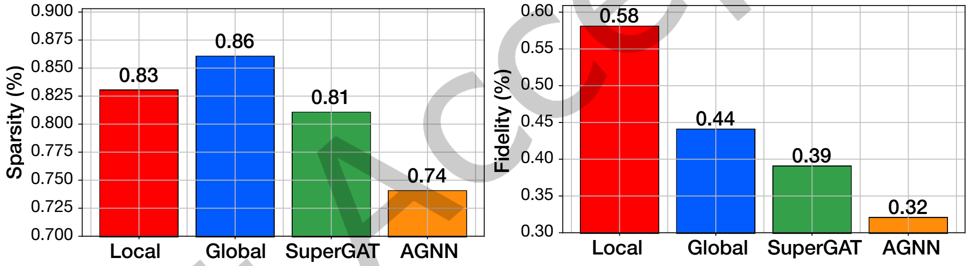

Fig. 7. Explainability quantification studies of HGIL and the other two baseline approaches.

5.3.5 Hierarchical Visualization. The learned interactions by the local sub-graph and global cross-graph attentions are illustrated in Figs. 8 and 9, respectively. Fig. 8 shows the local sub-graph attention where different types of objects in the three sub-graphs are linked with edges of colors representing their weights. We can see from the highlighted sub-graphs that the behaviors of the actor were mainly resulting from the *local contexts* at the lane segments near the intersection. Fig. 9 further visualizes the global cross-graph attention where the actor is actively interacting with the *global contexts* where an incoming pedestrian was walking toward the cross-walk of the intersection. From these two figures, we can further infer HGIL's capability in interpreting various AXIs based on our proposed HIG representations.

6 Discussion

We briefly discuss the deployment of HGIL in the following three aspects.

- **Extension to multi-agent scenarios**: In this paper, we focused on one focal vehicle as the agent to forecast its future trajectories. In addition, these traffic participants, say, the pedestrians or the peer vehicles in the complex AXIs, may vary their mobility or driving styles. Our formulation is general and can be further extended

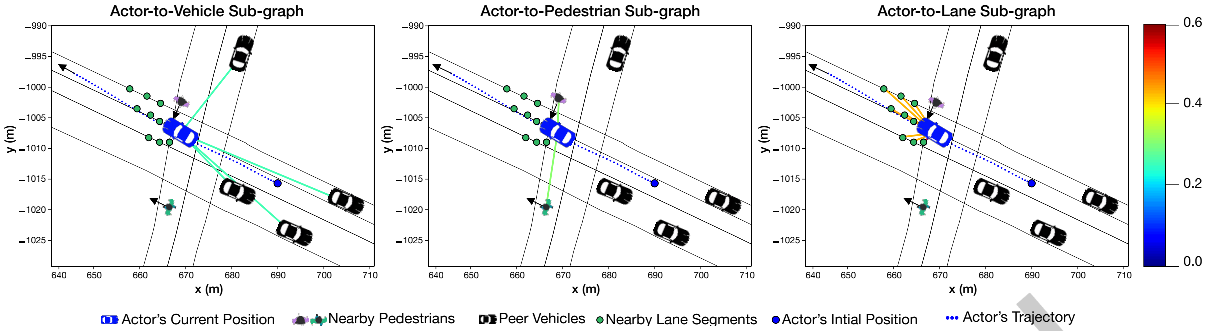

Fig. 8. Visualization of local sub-graph attention in AXIs. We illustrate the actor-to-vehicle, actor-to-pedestrian, and actor-to-lane sub-graphs.

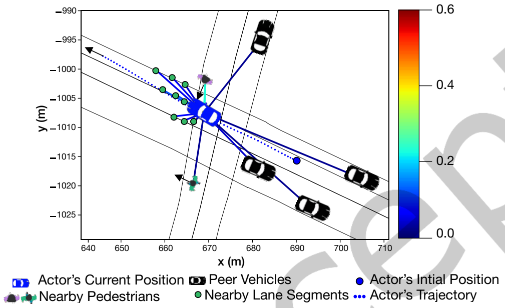

Fig. 9. Visualization of global cross-graph attention in AXIs.

to a multi-agent setting [4, 12, 16] by simultaneously creating the HIGs for different actors of interest at each timestamp. This will be considered in our future studies.

- **Extension to other data or sensing modalities:** To prepare the inputs to our HGIL, we leveraged the situation awareness information [26, 28] about the nearby traffic participants, which can be acquired through the sensors commonly available in the autonomous vehicles (e.g., LiDAR, cameras, or mmWave) [7]. However, our designs within HGIL are general enough to be extended upon availability of other data or sensing modalities (e.g., traffic signals).
- **Extension to complex deployment scenarios:** Our current studies focus on interaction awareness and hierarchical explainability for autonomous driving simulation. Further extension to practical and complex deployment scenarios will take into account aspects such as model complexity (e.g., parameter pruning and model compression) and uncertainty modeling when interacting with various traffic elements (e.g., noise in the perception module). We will explore these in our future work.

7 Conclusion

We have proposed HGIL, a heterogeneous graph-based imitation learning approach for autonomous driving simulation. We have designed a heterogeneous interaction graph (HIG) representation to provide local and global representations and awareness of AXIs. HGIL leverages the HIGs to generate the state embeddings, and a hierarchically-explainable GAIL approach captures the interactions and driving decision-making processes of the focal vehicle. We have performed extensive data-driven simulation and explanation studies, and demonstrated the accuracy, interaction awareness, and hierarchical explainability of HGIL in learning and capturing the complex

AXIs. We have compared HGIL with various baselines and state-of-the-art approaches, and our scheme outperforms the other methods in terms of displacement errors, sparsity, and fidelity.

Acknowledgment

This project is supported, in part, by the National Science Foundation (NSF) under Grant 2239897, Google Research Scholar Program Award (2021–2022), and NVIDIA Applied Research Accelerator Program Award (2021–2022).

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. Social LSTM: Human trajectory prediction in crowded spaces. In *Proc. IEEE/CVF CVPR.* 961–971.

[2] Peng Bao, Zonghai Chen, Jikai Wang, and Deyun Dai. 2022. Multiple agents' spatiotemporal data generation based on recurrent regression dual discriminator GAN. *Neurocomputing* 468 (2022), 370–383.

[3] Raunak Bhattacharyya, Blake Wulfe, Derek J Phillips, Alex Kuefler, Jeremy Morton, Ransalu Senanayake, and Mykel J Kochenderfer. 2022. Modeling human driving behavior through generative adversarial imitation learning. *IEEE T-ITS* (2022).

[4] Raunak P Bhattacharyya, Derek J Phillips, Blake Wulfe, Jeremy Morton, Alex Kuefler, and Mykel J Kochenderfer. 2018. Multi-agent imitation learning for driving simulation. In *Proc. IEEE/RSJ IROS.* 1534–1539.

[5] Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, et al. 2022. Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving. In *Proc. IEEE/RSJ IROS.* 8652–8659.

[6] Jianyu Chen, Bodi Yuan, and Masayoshi Tomizuka. 2019. Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety. In *Proc. IEEE/RSJ IROS.* 2884–2890.

[7] Chiho Choi, Joon Hee Choi, Jiachen Li, and Srikanth Malla. 2021. Shared cross-modal trajectory prediction for autonomous driving. In *Proc. IEEE/CVF CVPR.* 244–253.

[8] Nachiket Deo, Eric Wolff, and Oscar Beijbom. 2022. Multimodal trajectory prediction conditioned on lane-graph traversals. In *Proc. CoRL.* PMLR, 203–212.

[9] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. 2022. GOHOME: Graph-oriented heatmap output for future motion estimation. In *Proc. IEEE ICRA.* 9107–9114.

[10] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. 2018. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *Proc. IEEE/CVF CVPR.* 2255–2264.

[11] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Proc. NeurIPS* 29 (2016).

[12] Xiaosong Jia, Penghao Wu, Li Chen, Hongyang Li, Yu Liu, and Junchi Yan. 2022. HDGT: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *arXiv preprint arXiv:2205.09753* (2022).

[13] Dongkwan Kim and Alice Oh. 2022. How to find your friendly neighborhood: Graph attention design with self-supervision. *arXiv preprint arXiv:2204.04879* (2022).

[14] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. 2019. Social-BiGAT: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Proc. NeurIPS* 32 (2019).

[15] Gunmin Lee, Dohyeong Kim, Wooseok Oh, Kyungjae Lee, and Songhwai Oh. 2020. MixGAIL: Autonomous driving using demonstrations with mixed qualities. In *Proc. IEEE/RSJ IROS.* 5425–5430.

[16] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. 2019. Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning. In *Proc. IEEE ICRA.* 6658–6664.

[17] Lingyun Luke Li, Bin Yang, Ming Liang, Wenyuan Zeng, Mengye Ren, Sean Segal, and Raquel Urtasun. 2020. End-to-end contextual perception and prediction with interaction transformer. In *Proc. IEEE/RSJ IROS.* 5784–5791.

[18] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. 2020. Learning lane graph representations for motion forecasting. In *Proc. ECCV.* Springer, 541–556.

[19] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. 2020. PnPNet: End-to-end perception and prediction with tracking in the loop. In *Proc. IEEE/CVF CVPR.* 11553–11562.

[20] Sandra Carrasco Limeros, Sylwia Majchrowska, Joakim Johnander, Christoffer Petersson, and David Fernández Llorca. 2022. Towards Explainable Motion Prediction using Heterogeneous Graph Representations. *arXiv preprint arXiv:2212.03806* (2022).

[21] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. 2020. Multi-head attention for multi-modal joint vehicle motion forecasting. In *Proc. IEEE ICRA.* 9638–9644.

[22] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. 2020. CoverNet: Multimodal behavior prediction using trajectory sets. In *Proc. IEEE/CVF CVPR.* 14074–14083.

[23] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. 2019. Explainability methods for graph convolutional neural networks. In *Proc. IEEE/CVF CVPR.* 10772–10781.

[24] Debaditya Roy, Tetsuhiro Ishizaka, C Krishna Mohan, and Atsushi Fukuda. 2019. Vehicle trajectory prediction at intersections using interaction based generative adversarial networks. In *Proc. IEEE ITSC*. 2318–2323.

[25] Mahan Tabatabaie and Suining He. 2024. Driver Maneuver Interaction Identification with Anomaly-Aware Federated Learning on Heterogeneous Feature Representations. *Proc. ACM IMWUT* 7, 4, Article 180 (Jan. 2024), 28 pages.

[26] Mahan Tabatabaie, Suining He, and Kang G. Shin. 2023. Cross-Modality Graph-based Language and Sensor Data Co-Learning of Human-Mobility Interaction. *Proc. ACM IMWUT* 7, 3, Article 125 (Sept. 2023), 25 pages.

[27] Mahan Tabatabaie, Suining He, and Kang G. Shin. 2023. Interaction-Aware and Hierarchically-Explainable Heterogeneous Graph-based Imitation Learning for Autonomous Driving Simulation. In *Proc. IEEE/RSJ IROS*. 3576–3581.

[28] Mahan Tabatabaie, Suining He, Hao Wang, and Kang G Shin. 2024. Beyond "Taming Electric Scooters": Disentangling Understandings of Micromobility Naturalistic Riding. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 3 (2024), 1–24.

[29] Mahan Tabatabaie, Suining He, and Xi Yang. 2021. Reinforced Feature Extraction and Multi-Resolution Learning for Driver Mobility Fingerprint Identification. In *Proc. ACM SIGSPATIAL*. 69–80.

[30] Mahan Tabatabaie, Suining He, and Xi Yang. 2022. Driver Maneuver Identification with Multi-Representation Learning and Meta Model Update Designs. *Proc. ACM IMWUT* 6, 2, Article 74 (July 2022), 23 pages.

[31] Chen Tang, Nishan Srishankar, Sujitha Martin, and Masayoshi Tomizuka. 2021. Grounded relational inference: Domain knowledge driven explainable autonomous driving. *arXiv preprint arXiv:2102.11905* (2021).

[32] Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. 2018. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735* (2018).

[33] Senzhang Wang, Jiannong Cao, Hao Chen, Hao Peng, and Zhiqiu Huang. 2020. SeqST-GAN: Seq2Seq generative adversarial nets for multi-step urban crowd flow prediction. *ACM TSAS* 6, 4 (2020), 1–24.

[34] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. 2021. Argoverse 2: Next Generation Datasets for Self-driving Perception and Forecasting. In *Proc. NeurIPS*.

[35] Zheng Wu, Liting Sun, Wei Zhan, Chenyu Yang, and Masayoshi Tomizuka. 2020. Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving. *IEEE Robotics and Automation Letters* 5, 4 (2020), 5355–5362.

[36] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph neural networks. *Proc. NeurIPS* 32 (2019).

[37] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. 2017. Action-decision networks for visual tracking with deep reinforcement learning. In *Proc. IEEE/CVF CVPR*. 2711–2720.

[38] Wenyuan Zeng, Ming Liang, Renjie Liao, and Raquel Urtasun. 2021. LaneRCNN: Distributed representations for graph-centric motion forecasting. In *Proc. IEEE/RSJ IROS*. 532–539.

[39] Kunpeng Zhang, Xiaoliang Feng, Lan Wu, and Zhengbing He. 2022. Trajectory prediction for autonomous driving using spatial-temporal graph attention transformer. *IEEE T-ITS* 23, 11 (2022), 22343–22353.

[40] Guanjie Zheng, Hanyang Liu, Kai Xu, and Zhenhui Li. 2020. Learning to simulate vehicle trajectories from demonstrations. In *Proc. IEEE ICDE*. 1822–1825.

[41] Jinyun Zhou, Rui Wang, Xu Liu, Yifei Jiang, Shu Jiang, Jiaming Tao, Jinghao Miao, and Shiyu Song. 2021. Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization. In *Proc. IEEE/RSJ IROS*. 1450–1457.

[42] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017. Toward multimodal image-to-image translation. *Proc. NeurIPS* 30 (2017).

Received 25 July 2024; revised 21 November 2024; accepted 1 December 2024