Interaction-Aware and Hierarchically-Explainable Heterogeneous Graph-based Imitation Learning for Autonomous Driving Simulation

Mahan Tabatabaie[1] Suining He[1] Kang G. Shin[2]
University of Connecticut[1] University of Michigan–Ann Arbor[2]
{mahan.tabatabaie, suining.he}@uconn.edu kgshin@umich.edu

***Abstract*— Understanding and learning the actor-to-X interactions (AXIs), such as those between the focal vehicles (actor) and other traffic participants (e.g., other vehicles, pedestrians) as well as traffic environments (e.g., city/road map), is essential for the development of a decision-making model and simulation of autonomous driving (AD). Existing practices on imitation learning (IL) for AD 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 overcome these challenges, we propose HGIL, an interaction-aware and hierarchically-explainable Heterogeneous Graph-based Imitation Learning approach for AD 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 have corroborated the accuracy and explainability of HGIL in learning and capturing the complex AXIs.**

I. Introduction

Imitation learning (IL) for autonomous driving (AD) simulation aims to capture a cost function or a policy from the human driver demonstrations (e.g., real-world driving datasets) [1], [2]. In the IL setting, the actor, i.e., the focal vehicle, interacts with 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, lane changing, cutting in when interacting with other vehicles and road contexts (e.g., closure and road work), as well as the responses to pedestrians (e.g., yielding at a crosswalk).

Understanding and learning such complex AXIs is essential for designing the decision-making models and simulation of AD systems. Despite the recent IL advances [2]–[4], existing studies have not accounted for the following two major designs that are critical for *interaction awareness* and *model explainability* of an autonomous driving simulation framework:

(1) How to fuse and differentiate heterogeneous AXIs: Learning the decision-making process of AXIs performed by the human drivers resides in 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, deceleration) may result from various *heterogeneous* contexts of AXIs. Existing feature representations such as 2D rasterization [5], and homogeneous graphs [6] 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.

(2) How to enable the hierarchical explanation of IL for AD 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 interleaving with each other, and lead to complex AXI outcomes. Transparency requirements for autonomous driving simulation [7], [8] have established the needs of providing the hierarchical explainability to enable more trustworthy human-vehicle interactions [7], which, however, remains largely under-explored.

To overcome the above-mentioned gaps, we propose HGIL, a novel interaction-aware and hierarchically-explainable **H**eterogeneous **G**raph-based **I**mitation **L**earning framework for autonomous driving simulation. Towards 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 (e.g., other vehicles and pedestrians in our studies), and lane topology, leading to the enhanced learnability compared with the existing IL approaches.

(b) Hierarchically-Explainable IL Designs: Based on the heterogeneous interaction graph fusion, we have designed the novel 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 [9] with 40,000 driving scenes in complex

urban scenarios to validate the accuracy and explainability of our proposed HGIL in learning and capturing driving behaviors for AD simulation. Our simulation results have demonstrated that our HGIL outperforms the other state-of-the-art approaches (including [10]–[12]), and achieves hierarchical explainability regarding the AXIs.

II. Related Work

We briefly review the prior studies in two categories.

• **Graph Representations for Motion Modeling:** Prior motion modeling and planning studies for AD [5] have 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 [13], 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 represent the relations of the objects in the traffic environments [6]–[8]. For instance, Tang et al. [8] studied the neural relation inference to generate the interactive behavior interpretation. Different than the above works, 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 at the complex traffic scenes.

• **IL for Autonomous Driving Simulation:** Deep IL has been recently adopted for AD simulation and model development to capture the cost function or policy from the human driver demonstration data [2], [11]. Compared with inverse reinforcement learning (IRL) that is usually expensive to run and difficult to scale [14], generative adversarial imitation learning (GAIL) [15] generates the policy without capturing the cost function, and is able to scale in complex and spacious traffic environments. Different from the above-mentioned studies, our IL approach in HGIL provides a novel state embedding design based on HIG, which provides heterogeneous representability and hierarchical explainability.

III. Graph & Problem Formulation

A. Heterogeneous Interaction Graph Representation

Towards 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 $t-1$.

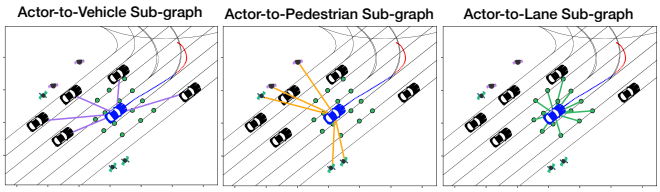

Fig. 1: Illustration of an HIG representation in HGIL.

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. IV-B).

(a) Actor-to-Vehicle Sub-graph $\mathbf{G}_t^{(\mathrm{c})}$: We form $\mathbf{G}_t^{(\mathrm{c})}$ by including the actor and the peer vehicles within a range from the actor as the nodes (say, 25m in our study). For each vehicle i of the K nearest peers observed ($i \in \{1,\dots,K\}$), 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 ones for the edges between the actor and peer vehicle nodes, and zeros otherwise.

(b) Actor-to-Pedestrian Sub-graph $\mathbf{G}_t^{(\mathrm{p})}$: Similar to $\mathbf{G}_t^{(\mathrm{c})}$, we form $\mathbf{G}_t^{(\mathrm{p})}$ that includes the pedestrians that are within a range (25m in our study) from the actor. We find the corresponding pedestrian node feature $j \in \{1,\dots,P\}$ as $\mathbf{V}_{t,j}^{(\mathrm{p})} = \left[x_{t,j}^{(\mathrm{p})}, y_{t,j}^{(\mathrm{p})}, v_{t,j}^{(\mathrm{p})}, \theta_{t,j}^{(\mathrm{p})}, d_{t,j}^{(\mathrm{p})}\right] \in \mathbb{R}^5$, 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 ones for the edges between the actor and pedestrian nodes, and zeros otherwise.

(c) Actor-to-Lane Sub-graph $\mathbf{G}_t^{(\mathrm{l})}$: 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 (each is 25.45m on 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 $\mathbf{V}_{t,m}^{(\mathrm{l})} = \left[x_{t,m}^{(\mathrm{l})}, y_{t,m}^{(\mathrm{l})}, d_{t,m}^{(\mathrm{l}),}, e_{t,m}^{(\mathrm{l})}\right] \in \mathbb{R}^4$, 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-mentioned sub-graphs, we denote an HIG at a timestamp t as $\mathbf{G}_t = \left\{\mathbf{G}_t^{(\mathrm{c})}, \mathbf{G}_t^{(\mathrm{p})}, \mathbf{G}_t^{(\mathrm{l})}\right\}$.

B. Concepts and Problem Formulation

• **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., $\mathbf{S}_t = \{\mathbf{G}_{t-L}, \mathbf{G}_{t-L+1}, \ldots, \mathbf{G}_t\}$. Furthermore, without loss of generality, we account for the focal vehicle as the actor, while the formulation is general enough to be extended to the multi-agent setting [3], [16].

• **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 $\mathbf{A}_t$, i.e., $\mathbf{A}_t \sim \pi(\mathbf{A}|\mathbf{S}_t)$. In this prototype study, HGIL rolls out and generates a series of planned displacements towards the x and y axes, $\mathbf{A}_t = \left[\left(\Delta x_{(t+1)}^{(\mathrm{f})}, \Delta y_{(t+1)}^{(\mathrm{f})}\right), \ldots, \left(\Delta x_{(t+L)}^{(\mathrm{f})}, \Delta y_{(t+L)}^{(\mathrm{f})}\right)\right] \in \mathbb{R}^{L\times 2}$, for the future L timestamps.

• **Problem Definition:** Give 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), the GAIL in HGIL optimizes the actor's policy π, such that the resulting actions $\mathbf{A}$ of the actor – that is, series of planned displacements – are *indistinguishable* from the expert demonstrations (i.e., human driver demonstration). This can be formalized as finding a Nash equilibrium [15] within a minimax game between a policy generator network approximating π, and a discriminator network ψ, i.e.,

$$\min_{\pi}\max_{\psi} \mathbb{E}_{\mathbf{S},\mathbf{A}\sim\pi}[\log(\psi(\mathbf{S},\mathbf{A}))] + \mathbb{E}_{\mathbf{S},\mathbf{A}\sim\pi^{\mathrm{e}}}[\log(1-\psi(\mathbf{S},\mathbf{A}))], \tag{3}$$

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. IV).

IV. Model Core Designs

A. Overview of State Embedding Processing

We overview the state embedding processing of HGIL in Fig. 2. 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* (I) 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* (II) to quantify the actor's interactions in a global context, and generates the

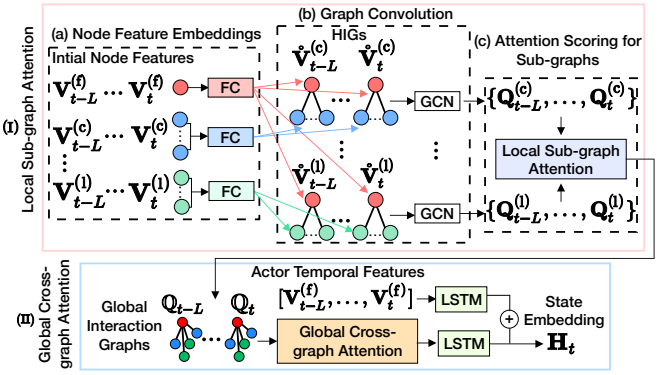

Fig. 2: Architecture overview of state embeddings within HGIL, which consists of (I) local sub-graph attention and (II) global cross-graph attention.

state embeddings for policy learning (detailed in Sec. IV-C).

B. State Embeddings with HIGs

(I) Local Sub-graph Attention: We note that a 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 indicate 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, $\mathbf{V}_t = \left\{\mathbf{V}_t^{(f)}, \mathbf{V}_t^{(c)}, \mathbf{V}_t^{(p)}, \mathbf{V}_t^{(l)}\right\}$, 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 $\overline{\mathbf{V}}_t = \left\{\overline{\mathbf{V}}_t^{(\mathrm{f})}, \overline{\mathbf{V}}_t^{(\mathrm{c})}, \overline{\mathbf{V}}_t^{(\mathrm{p})}, \overline{\mathbf{V}}_t^{(\mathrm{l})}\right\}$, where $\overline{\mathbf{V}}_t^{(\mathrm{f})} \in \mathbb{R}^{1\times B_1}$, $\overline{\mathbf{V}}_t^{(\mathrm{c})} \in \mathbb{R}^{K\times B_1}$, $\overline{\mathbf{V}}_t^{(\mathrm{p})} \in \mathbb{R}^{P\times B_1}$, $\overline{\mathbf{V}}_t^{(\mathrm{l})} \in \mathbb{R}^{R\times B_1}$.

Afterwards, 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 features with a separate graph convolutional (GCN) layer (with 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., $\hat{\mathbf{V}}_t^{(\mathrm{c})} = \left[\overline{\mathbf{V}}_t^{(\mathrm{c})} \middle|\middle| \overline{\mathbf{V}}_t^{(\mathrm{f})}\right]$, which is further fed to the GCN layer, i.e.,

$$\mathbf{Q}_t^{(\mathrm{c})} = \left(\hat{\mathbf{D}}^{(\mathrm{c})}\right)^{-\frac{1}{2}} \cdot \left(\mathbf{E}_t^{(\mathrm{c})} + \mathbf{I}\right) \cdot \left(\hat{\mathbf{D}}^{(\mathrm{c})}\right)^{-\frac{1}{2}} \cdot \hat{\mathbf{V}}_t^{(\mathrm{c})} \cdot \mathbf{W}^{(\mathrm{c})} + \mathbf{b}^{(\mathrm{c})}, \tag{4}$$

where $\hat{\mathbf{D}}^{(\mathrm{c})} \in \mathbb{R}^{(K+1)\times(K+1)}$ represents the diagonal degree matrix, i.e., $\hat{\mathbf{D}}^{(\mathrm{c})}[i,i] = \sum_j \mathbf{E}_t^{(\mathrm{c})}[i,j]$. $(\mathbf{E}_t^{(\mathrm{c})} + \mathbf{I})$ 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 set of 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.,

$$\mathbf{Q}_t^{(\mathrm{f})} = \left[\mathbf{Q}_t^{(\mathrm{c})}[-1,:] \middle|\middle| \mathbf{Q}_t^{(\mathrm{p})}[-1,:] \middle|\middle| \mathbf{Q}_t^{(\mathrm{l})}[-1,:]\right], \tag{5}$$

where $\mathbf{Q}_t^{(\mathrm{c})}[-1,:]$, $\mathbf{Q}_t^{(\mathrm{p})}[-1,:]$, and $\mathbf{Q}_t^{(\mathrm{l})}[-1,:]$ all 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.

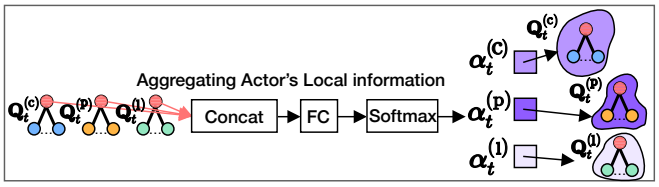

Fig. 3: Illustration of the attention scoring for sub-graphs.

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

(II) Global Cross-graph Attention: To capture the human driver decisions in a joint response to different involved objects (for instance, other traffic participants and map topology) in the global context of the traffic environments, we have further 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. (5) by multiplying the sub-graph attention scores with the corresponding actor node features (i.e., the last row) in the sub-graphs, i.e.,

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

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.,

$$\mathbb{Q}_t = \left[\overline{\mathbf{Q}}_t^{(\mathrm{f})} \middle|\middle| \mathbf{Q}_t^{(\mathrm{c})}[1:K,:] \middle|\middle| \mathbf{Q}_t^{(\mathrm{p})}[1:P,:] \middle|\middle| \mathbf{Q}_t^{(\mathrm{l})}[1:R,:]\right], \tag{6}$$

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

Afterwards, we 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

$$\Gamma_t[i,j] = \frac{\exp(\mu_t[i,j])}{\sum_{o=1}^{T}\exp(\mu_t[i,o])}, \tag{7}$$

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

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

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

$$\mathbf{F}_t = \Gamma_t \cdot \mathbf{W}_g + \mathbf{b}_g, \tag{8}$$

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., $\mathbf{S}_t = \{\mathbf{G}_{t-L}, \mathbf{G}_{t-L+1}, \ldots, \mathbf{G}_t\}$. 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., $\mathbf{H}_t' = \mathtt{LSTM}\left([\mathbf{F}_{(t-L)}[-1,:],\ldots,\mathbf{F}_t[-1,:]]\right)$. The sequence embeddings from the global interaction graphs are further added with the temporal feature embeddings of the actor node features generated by another LSTM module, i.e.,

$$\mathbf{H}_t = \mathbf{H}_t' \oplus \mathtt{LSTM}\left([\mathbf{V}_{t-L}^{(\mathrm{f})},\ldots,\mathbf{V}_t^{(\mathrm{f})}]\right), \tag{9}$$

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

C. Training Designs of HGIL

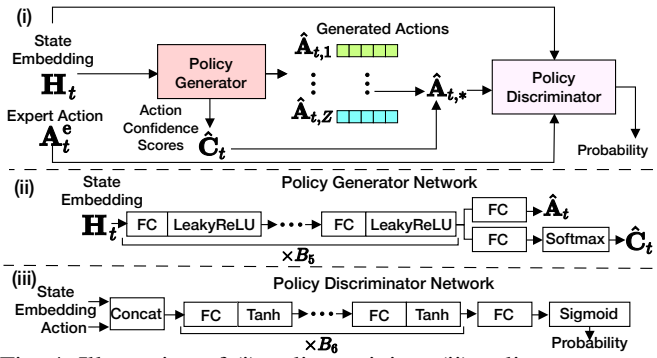

Fig. 4: Illustration of (i) policy training; (ii) policy generator network; and (iii) policy discriminator network.

- **Policy Generator and Discriminator Networks:** Fig. 4(i) illustrates the model training process given the state embeddings $\mathbf{H}_t$. Based on the state embeddings, HGIL provides a policy generator network consisted 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 demonstra-

tion). We further show the structures of the two networks in Figs. 4(ii) and (iii).

(a) The policy generator network takes in the state embeddings of the actor $\mathbf{H}_t$, and returns the set of Z possible sequences of displacement actions, $\{\hat{\mathbf{A}}_{t,i}\}$ ($i \in \{1,\ldots,Z\}$), 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 further outputs the confidence scores $\hat{\mathbf{C}}_t \in \mathbb{R}^Z$ (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 corresponds to the maximum confidence score in $\hat{\mathbf{C}}_t$); or (ii) the actual actions $\mathbf{A}_t^{\mathrm{e}}$ performed in the human driver demonstration data. Thus, given the concatenation of input actions ($\mathbf{A}_{t,*}$ or $\mathbf{A}_t^{\mathrm{e}}$) as well as state embeddings $\mathbf{H}_t$, ψ estimates the probability (i.e., $\psi\left(\left[\mathbf{H}_t \middle|\middle| \mathbf{A}_t^{\mathrm{e}}\right]\right)$ or $\psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)$) that the input action resembles the human driver demonstration.

• **Model Training Loss:** In order to capture the discrepancy between the generated actions and the human driver demonstration, we take into account the following types of loss within HGIL, i.e., (a) displacement regression loss ℓ_r and (b) confidence cross-entropy loss ℓ_c, and further integrate them in 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 $\hat{\mathbf{C}}_t$, $\hat{\mathbf{A}}_{t,*}$, and the actual action in the human driver demonstration.

(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.,

$$\ell_c \triangleq -\sum_{i=1}^{Z} \left(\mathbf{B}_t[i] \cdot \log\left(\hat{\mathbf{C}}_t[i]\right)\right). \tag{10}$$

Based on the above designs, we further 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. (3), 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. To summarize, the policy generator minimizes

$$\mathrm{L}_g \triangleq \beta_r \cdot \ell_r + \beta_c \cdot \ell_c + \beta_d \cdot \log\left(1-\psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)\right), \tag{11}$$

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

(ii) Policy Discriminator Loss L_d: Based on the formulation in Eq. (3), the policy discriminator network further performs the opposite optimization against the generator, by maximizing

$$\mathrm{L}_d \triangleq \log\left(\psi\left(\left[\mathbf{H}_t \middle|\middle| \mathbf{A}_t^{\mathrm{e}}\right]\right)\right) + \log\left(1-\psi\left(\left[\mathbf{H}_t || \hat{\mathbf{A}}_t\right]\right)\right). \tag{12}$$

Since there is a minimax game between the policy generator and discriminator networks [15], we train them iteratively based on Eqs. (11) and (12) until convergence.

V. Data-driven Model Emulation Studies

A. Baseline Approaches & Simulation Settings

• **Baseline Approaches:** For AD simulation, we compare performance of HGIL with DualDisc [10], CGAIL [11], SocialGAN [12], LaneGCN-GAIL [6], MGAIL [1], and SeqST-GAN [17].

• **Dataset Studied & Performance Metrics:** We leverage the large-scale human driver demonstration dataset Argoverse v2 [9] to perform 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 based on final displacement error – that is, distance of the final generated position from the true one in the demonstration (denoted as FDE) – and average displacement error – that is, 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 concern 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.

• **Model Parameter Settings:** Since the Argoverse v2 dataset is collected with a 10Hz frequency and therefore we set $L = 30$ to leverage 3s of historical information to generate next 3s of actions. Similar to the prior studies [6], [18], 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 (Figs. 4(i) and (ii), and each FC layer is with 32 hidden units. We set $\beta_s = \beta_d = 1$ and $\beta_c = 0.3$ in Eq. (11).

• **Simulation Environment & Model Training Setup:** We implemented our networks based on Pytorch 1.13.1 and Python 3.8.16. We performed our 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 the above settings, the training time of our HGIL is on average 361.3ms

per AXI scene (each driving scene lasts for 6s on average).

The training setting of HGIL is as follows. We first pre-train the policy generator network with a learning rate decay (from 0.01 to 0.001) for 300 iterations (Adam optimizer is adopted). We then train the policy generator and discriminator networks according to Eqs. (11) and (12) with a learning rate of 0.001 for 200 iterations. At each iteration we sample 1,000 driving scenes to train the networks.

B. Performance Results and Visualization Studies

• **Overall Performance:** We present the overall performance of HGIL in Table I, and compare HGIL with other IL-based methods. We can observe that HGIL outperforms the other baselines in learning the human driving behaviors in the AXIs. In particular, our HGIL achieves 15.4%, 20.3%, 22.7%, 27.7%, and 7.3% lower in terms of FDE, ADE, minFDE, minADE, and MR. Via the local sub-graph attention and the global cross-graph attention, HGIL yields better performance in learning the human drivers.

TABLE I Overall performance results & ablation studies.

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%
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%
MGAIL	3.45	1.42	2.91	1.22	27%
SeqST-GAN	3.54	1.46	3.01	1.25	29%
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 Confidence loss	3.05	1.23	2.55	1.17	25%

• **Model Ablation Studies:** We show the model ablation studies on HGIL to evaluate the importance of different designs in Table I. Specifically, we compare the performance of complete HGIL designs with the following variations: w/o HIG, w/o the local sub-graph attention, and w/o the global cross-graph attention. We can observe the overall higher importance of the local sub-graph attention and global cross-graph attention in learning and capturing the human driving behaviors.

• **Hierarchical Visualization:** We illustrate the explained interactions by local sub-graph and global cross-graph attentions in Fig. 5. We show in Figs. 5(a)–(c) the local sub-graph attention where different types of objects (black arrows represent the moving directions of the focal vehicle and the pedestrians) in 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. 5(d) further visualizes the global cross-graph attention where the actor is actively interacting with the *global contexts* where an incoming pedestrian was walking towards the crosswalk of the intersection.

VI. Conclusion

We propose HGIL, a heterogeneous graph-based imitation learning approach for AD simulation. We have designed a heterogeneous interaction graph (HIG) representation to provide local and global representation and awareness of AXIs. HGIL leveraged the HIGs to generate the state embeddings,

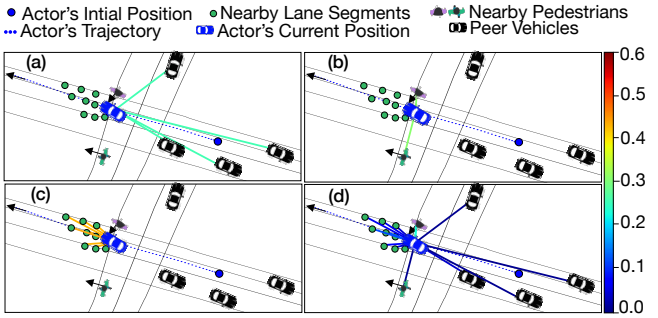

Fig. 5: (a)–(c) Local attention (actor-to-vehicle, actor-to-pedestrian, & actor-to-lane) and (d) global attention in AXIs.

and a hierarchically-explainable GAIL approach captures the interactions and driving decision-making processes of the focal vehicle. Extensive data-driven simulation and explanation studies have demonstrated the accuracy and explainability of HGIL in learning and capturing the complex AXIs.

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] E. Bronstein, M. Palatucci *et al.*, "Hierarchical model-based imitation learning for planning in autonomous driving," in *IEEE/RSJ IROS*, 2022.

[2] R. Bhattacharyya, B. Wulfe *et al.*, "Modeling human driving behavior through generative adversarial imitation learning," *IEEE T-ITS*, 2022.

[3] R. P. Bhattacharyya, D. J. Phillips *et al.*, "Multi-agent imitation learning for driving simulation," in *IEEE/RSJ IROS*, 2018.

[4] J. Zhou, R. Wang *et al.*, "Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization," in *IEEE/RSJ IROS*, 2021.

[5] L. L. Li, B. Yang *et al.*, "End-to-end contextual perception and prediction with interaction transformer," in *IEEE/RSJ IROS*, 2020.

[6] M. Liang, B. Yang *et al.*, "Learning lane graph representations for motion forecasting," in *ECCV*. Springer, 2020.

[7] S. C. Limeros, S. Majchrowska *et al.*, "Towards explainable motion prediction using heterogeneous graph representations," *arXiv:2212.03806*, 2022.

[8] C. Tang, N. Srishankar *et al.*, "Grounded relational inference: Domain knowledge driven explainable autonomous driving," *arXiv:2102.11905*, 2021.

[9] B. Wilson, W. Qi *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *NeurIPS*, 2021.

[10] P. Bao, Z. Chen *et al.*, "Multiple agents' spatiotemporal data generation based on recurrent regression dual discriminator GAN," *Neurocomputing*, 2022.

[11] J. Li, H. Ma *et al.*, "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," in *IEEE ICRA*, 2019.

[12] A. Alahi, K. Goel *et al.*, "Social LSTM: Human trajectory prediction in crowded spaces," in *IEEE/CVF CVPR*, 2016.

[13] T. Phan-Minh, E. C. Grigore *et al.*, "CoverNet: Multimodal behavior prediction using trajectory sets," in *IEEE/CVF CVPR*, 2020.

[14] Z. Wu, L. Sun *et al.*, "Efficient sampling-based maximum entropy inverse reinforcement learning with application to autonomous driving," *IEEE Robotics and Automation Letters*, 2020.

[15] J. Ho and S. Ermon, "Generative adversarial imitation learning," *NeurIPS*, 2016.

[16] X. Jia, P. Wu *et al.*, "HDGT: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding," *arXiv:2205.09753*, 2022.

[17] S. Wang, J. Cao *et al.*, "SeqST-GAN: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction," *ACM TSAS*, 2020.

[18] W. Zeng, M. Liang *et al.*, "LaneRCNN: Distributed representations for graph-centric motion forecasting," in *IEEE/RSJ IROS*, 2021.