

Cooperative Autonomous Driving in Diverse Behavioral Traffic: A Heterogeneous Graph Reinforcement Learning Approach

Qi Liu, Xueyuan Li, Zirui Li, Juhui Gim*

Abstract—Navigating heterogeneous traffic environments with diverse driving styles poses a significant challenge for autonomous vehicles (AVs) due to their inherent complexity and dynamic interactions. This paper addresses this challenge by proposing a heterogeneous graph reinforcement learning (GRL) framework enhanced with an expert system to improve AV decision-making performance. Initially, a heterogeneous graph representation is introduced to capture the intricate interactions among vehicles. Then, a heterogeneous graph neural network with an expert model (HGNN-EM) is proposed to effectively encode diverse vehicle features and produce driving instructions informed by domain-specific knowledge. Moreover, the double deep Q-learning (DDQN) algorithm is utilized to train the decision-making model. A case study on a typical four-way intersection, involving various driving styles of human vehicles (HVs), demonstrates that the proposed method has superior performance over several baselines regarding safety, efficiency, stability, and convergence rate, all while maintaining favorable real-time performance.

Index Terms—Autonomous vehicle, decision-making, graph reinforcement learning, heterogeneous graph neural network, heterogeneous traffic.

I. INTRODUCTION

THE decision-making for autonomous vehicles (AVs) in modern traffic is highly challenging due to the intricate interactions with these heterogeneous traffic participants. Among these participants, human-driven vehicles (HVs) present significant variability in traits and intentions [1]. Considering the Fig. 1 as an intuitive example. Aggressive HVs often exhibit higher speeds and risk-prone behaviors, showing little regard for other drivers during interactions, whereas conservative HVs prioritize safety, maintaining slower speeds and cautious maneuvers. Normal HVs, in contrast, display a driving style that falls between these two extremes. AVs must carefully negotiate with different styles of HVs to execute reasonable driving behaviors, achieving a superior balance between driving safety and efficiency.

Acknowledgement: This research is funded by Global - Learning & Academic research institution for Master's-PhD students, and Postdocs (LAMP) Program of the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education (No. RS2024-00444460) and "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-003).

*Corresponding authors: Juhui Gim

Qi Liu, Xueyuan Li, and Zirui Li are with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing, China. (E-mails: 3120195257@bit.edu.cn; lixueyuan@bit.edu.cn; 3120195255@bit.edu.cn.)

Juhui Gim is with the School of Mechatronics Engineering, Changwon National University, Gyeongsangnam-do, South Korea. Email: juhuigim@changwon.ac.kr.

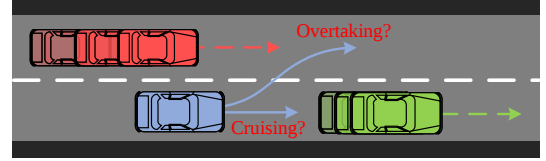


Fig. 1. A schematic of scenario with different styles of HVs. The ego AV (blue) must balance safety and efficiency, while comprehensively considering interactions with the aggressive HV (red) and conservative HV (green).

A comprehensive overview of general decision-making techniques for AVs is presented in [2]. Various studies have addressed the challenges of modeling interactions among heterogeneous traffic participants. In [3], the driving styles of HVs are inferred from prior knowledge, then a level-k cooperative game is proposed to generate driving instructions. In [4], interactions among heterogeneous HVs are directly modeled through the self-attention mechanism based on their motion features, then the actor-critic (AC) algorithm is utilized for policy generation. A similar approach targeting the merging scenario is explored in [5]. However, these approaches fail to accurately represent the intricate interactions among diverse HVs, limiting their decision-making performance in such heterogeneous traffic.

For encoding heterogeneous traffic information, graph reinforcement learning (GRL)-based methods have demonstrated significant potential. A notable approach in [6] models a highway ramping scenario as an undirected graph, facilitating cooperative lane-change decisions for heterogeneous AVs. Building upon this, a multi-view graph representation technique is introduced in [7] to more effectively capture vehicle interactions. In [8], scenarios involving heterogeneous vehicles are represented using both a bird's-eye view (BEV) image and an undirected graph, with the graph attention network (GAT) [9] encoding the information to generate AV maneuvers. Moreover, in [10], an intersection with pedestrians and vehicles of varying driving styles is constructed. Then, a fusion framework consisting of spatial-temporal graph neural network (STGNN), state inference, and trajectory prediction is proposed for AV's decision-making. However, these studies depend on homogeneous technologies to process the features of heterogeneous traffic, which limits the potential for further improvements in decision-making performance.

In conclusion, current approaches still encounter several limitations:

- **Limitations on heterogeneous interaction modeling:** Ex-

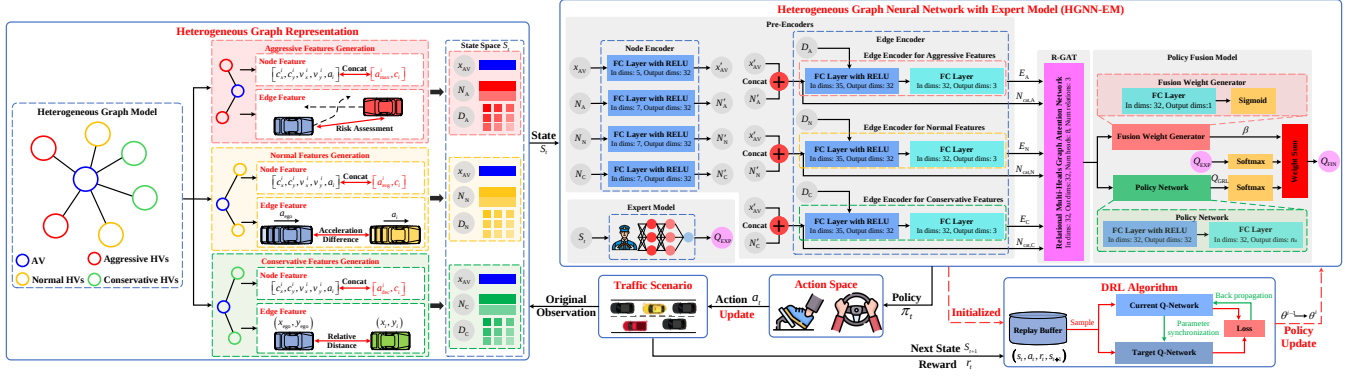


Fig. 2. The schematic diagram of the proposed method.

isting studies fail to comprehensively and accurately model the intricate interactions among diverse traffic participants, limiting AVs' understanding of heterogeneous traffic scenarios and reducing decision-making performance.

- **Limitations on policy generation:** Most studies still depend on homogeneous techniques for processing heterogeneous features, potentially leading to suboptimal or unsafe decisions and limiting the training efficiency of decision-making models.

The objective of this paper is to tackle the above decision-making challenges of AVs' current technologies in such heterogeneous traffic by focusing on two critical aspects: 1) Accurate and comprehensive modeling of interactions among heterogeneous traffic participants. 2) Efficient encoding of heterogeneous traffic information and generation of expert-level driving strategies that can well balance safety and efficiency. Specifically, a heterogeneous graph neural network with an expert model (HGNN-EM) is proposed to satisfy the above demands. The main contributions are summarized as follows:

- A heterogeneous graph representation is presented to comprehensively characterize the mutual effects between the ego AV and various styles of HVs, including driving risk evaluation, acceleration difference, and relative distance.
- A HGNN-EM is proposed to effectively encode pertinent features from heterogeneous traffic and generate refined driving instructions. By integrating GRL-based policies with expert model-based policies, this approach significantly enhances the decision-making capabilities of AVs.
- Comprehensive experiments are conducted on a typical four-way intersection with heterogeneous traffic participants. Results demonstrate that the proposed method outperforms baseline approaches in terms of safety, efficiency, stability, and convergence speed.

II. PROBLEM FORMULATION

The formulation of the decision-making process within the GRL-based framework can be modeled as a Markov Decision Process (MDP), formulated by tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Specifically, P is the transition probability, $\gamma \in (0, 1]$ is the discount factor. The other elements are described as follows:

- 1) **State Space $\mathcal{S}$:** The intersection is modeled as a heterogeneous undirected graph, where the state space encompasses

features of all HVs and the AV within the intersection, generated through the proposed heterogeneous graph representation. The construction process of the state space will be illustrated in Section III-A.

- 2) **Action Space $\mathcal{A}$:** The action space of the ego AV consists of five high-level discrete driving commands, including *accelerate*, *slow down*, *cruising*, *change lane to the left*, and *change lane to the right*.

- 3) **Reward Function R :** A reward function is designed to encourage the AV to turn left safely at the intersection as fast as possible without collisions. Specifically,

$$R = r_{\text{goal}} + r_{\text{col}} + r_{\text{vel}} \quad (1)$$

where r_{goal} denotes the task reward, $r_{\text{goal}} = 2$ if the ego AV successfully navigates the intersection, otherwise $r_{\text{goal}} = 0$. r_{col} denotes the collision reward to ensure driving safety, $r_{\text{col}} = -2$ if the ego AV collides with HVs, otherwise $r_{\text{col}} = 0$. r_{vel} denotes the dense velocity reward to encourage driving efficiency; specifically, $r_{\text{vel}} = 0.04 \cdot \min \{v_{\text{ego}}/v_{\text{max}}, 1\}$, where v_{max} is the maximum permissible longitudinal speed for AV.

III. METHODS

The proposed method comprises two primary components: the heterogeneous graph representation and the HGNN-EM, as shown in Fig. 2.

A. Heterogeneous Graph Representation

The intersection scenario is modeled as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_i, i \in \{1, 2, \dots, n\}\}$ denotes a set of node attributes (vehicles); $\mathcal{E} = \{\varepsilon_{ij}, i, j \in \{1, 2, \dots, n\}\}$ denotes the set of edge attributes (interactions); n represents the total number of all vehicles. Then, the heterogeneous graph representation is utilized to integrate diverse node and edge attributes, reflecting the different styles of vehicles, to construct the graphical state space at each time step t . For ease of description, the symbol t is omitted in the following equation. Specifically:

$$G = [x_{\text{AV}}, N_A, N_N, N_C, D_A, D_N, D_C] \quad (2)$$

where x_{AV} denotes the feature vector of the ego AV related to basic motion, including longitudinal coordinate c_x , lateral coordinate c_y , longitudinal speed v_x , lateral speed v_y , and acceleration a_x , i.e., $x_{\text{AV}} = [c_x, c_y, v_x, v_y, a_x]$. N_A, N_N , and

N_C denote the node feature matrix for aggressive HVs, normal HVs, and conservative HVs, respectively. Similarly, D_A , D_N , and D_C denotes the adjacency matrix for aggressive HVs, normal HVs, and conservative HVs, respectively.

1) *Node Feature Matrix*: The node feature matrix encapsulates the driving features of each vehicle. Feature vectors of all HVs include basic motion characteristics $\tilde{x} = [c_x, c_y, v_x, v_y, a_x]$ and unique features specific to their driving style:

- **Aggressive HVs**: The maximum acceleration $a_{\max}$ is used to highlight the vehicle's aggressive driving styles, representing its tendency for risk-prone behaviors.
- **Normal HVs**: The average acceleration over T time steps a_{avg} is selected to provide a stable view of its typical behavior in traffic.
- **Conservative HVs**: The maximum deceleration $a_{\text{dec}} = -4.5 \text{ m/s}^2$ is chosen to emphasize the vehicle's avoidance capability, showcasing its cautious driving manner.

Let $\delta^i \in \{0 \text{ (aggressive)}, 1 \text{ (normal)}, 2 \text{ (conservative)}\}$ be the category value of the i_{th} HV, the final node features matrix can be formulated as follows:

$$\begin{aligned} N_A &= [x_A^1, x_A^2, \dots, x_A^{n_A}], x_A^i = [\tilde{x}_A^i, a_{\max}^i, \delta^i] \\ N_N &= [x_N^1, x_N^2, \dots, x_N^{n_N}], x_N^i = [\tilde{x}_N^i, a_{\text{avg}}^i, \delta^i] \\ N_C &= [x_C^1, x_C^2, \dots, x_C^{n_C}], x_C^i = [\tilde{x}_C^i, a_{\text{dec}}^i, \delta^i] \end{aligned} \quad (3)$$

2) *Adjacency Matrix*: The adjacency matrix represents the mutual effects among vehicles through specific edge values e^{ij} . To account for HVs' varying driving styles, the ego AV needs to focus on different interaction patterns based on each HV's unique characteristics to make informed and adaptive decisions.

• Interaction model for Aggressive HVs

Given that aggressive HVs tend to exhibit higher risk-taking behaviors, the ego AV must evaluate the potential for collisions or unsafe situations resulting from their assertive maneuvers. To assess these driving risks, a two-dimensional time to collision (2D-TTC) model [11] is established, illustrated in Fig. 3.

Then, the general pattern of the edge value e_A^{ij} within D_A can be formulated as follows:

$$e_A^{ij} = \frac{d_{i,j}}{|\Delta \mathbf{v}_p^{i,j}|} = \frac{d_{i,j} |\Delta \mathbf{p}_{i,j}|}{\Delta \mathbf{v}_{i,j} \cdot \Delta \mathbf{p}_{i,j}} \quad (4)$$

where $d_{i,j}$ is the relative distance between the i_{th} and j_{th} vehicle; $\Delta \mathbf{v}_p^{i,j}$ is the projection of velocity difference vector $\Delta \mathbf{v}_{i,j}$ in the direction of unit projection vector $\Delta \mathbf{p}_{i,j}$. It should be noted that if e_A^{ij} is negative, it will be adjusted to 10 in statistical analysis.

• Interaction model for Normal HVs

Normal HVs tend to exhibit stable driving behavior, making it essential to evaluate the driving similarity between the AV and these HVs to ensure smooth and predictable interactions. Considering that acceleration is an important factor in reflecting the driving styles, the difference of average acceleration over T time steps is chosen to calculate the edge value e_N^{ij} within D_N :

$$e_N^{ij} = |a_{\text{avg}}^j - a_{\text{avg}}^i| \quad (5)$$

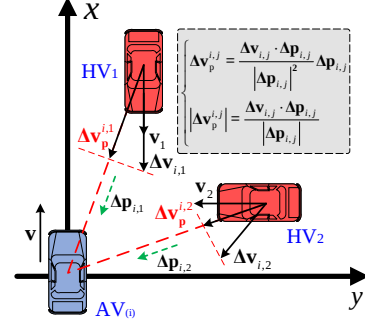


Fig. 3. The schematic diagram of the 2D-TTC model.

• Interaction model for Conservative HVs

Conservative HVs generally exhibit cautious driving behavior, the ego AV needs to focus on the potential influences of its actions on these vehicles to adjust its decision-making style dynamically. Given that distance is a direct indicator of driving influence, the relative distance is selected to generate the edge value e_C^{ij} within D_C :

$$e_C^{ij} = \sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2} \quad (6)$$

where $\Delta x_{i,j}$ and $\Delta y_{i,j}$ denote the relative longitudinal and lateral distance, respectively.

B. HGNN-EM

The function of the HGNN-EM is to encode the heterogeneous graphical state space efficiently to generate reasonable driving policies. The HGNN-EM consists of four key components: pre-encoders including node and edge encoders, an expert model, a relational graph attention network (R-GAT), and a policy fusion model.

1) *Pre-Encoders*: The pre-encoders consist of a node encoder and three edge encoders. The node encoder projects features of different vehicle types—AV, aggressive HVs, normal HVs, and conservative HVs—into a unified embedding space using four separate FC layers with ReLU activation. The embeddings x'_{AV} , N'_A , N'_N , and N'_C for the respective vehicle types are generated for the downstream processing.

The edge encoder uses FC layers to derive learned edge values from node and edge features to offer greater flexibility and adaptability in heterogeneous traffic, enabling the model to more accurately capture intricate mutual effects among vehicles and enhance its representation capability. Let Φ_{edge} denote the edge encoder operator, the calculation of the learned edge values can be described as follows:

$$\begin{aligned} E_A &= \Phi_{\text{edge},A} (N_{\text{cat},A}, D_A), N_{\text{cat},A} = [x'_{AV} \oplus N'_A] \\ E_N &= \Phi_{\text{edge},N} (N_{\text{cat},N}, D_N), N_{\text{cat},N} = [x'_{AV} \oplus N'_N] \\ E_C &= \Phi_{\text{edge},C} (N_{\text{cat},C}, D_C), N_{\text{cat},C} = [x'_{AV} \oplus N'_C] \end{aligned} \quad (7)$$

where $\oplus$ represents the matrix concatenation operator.

2) *Expert Model*: The expert model, designed to enhance the GRL-based driving policy with domain-specific guidance, is trained using supervised learning from interactions between expert drivers who can perform highly reasonable driving behaviors and a traffic simulator. The dataset includes the

heterogeneous state space (equation (2)) as input and discrete actions $a_{\text{EXP}} \in \mathcal{A}$ as output. Moreover, the expert model shares the same architecture as the HGNN-EM, excluding the fusion weight generator component.

Initially, the output of the dataset is transformed into a one-hot encoded vector $Y = [y_1, y_2, \dots, y_{n_a}]$, corresponding to the action space, to serve as the target value, where n_a denotes the dimension of the action space. Then, the cross-entropy loss is utilized to ensure that the Q-value table generated by the expert model continuously approximates the expert's action distribution during the training process. The loss function is defined as follows:

$$L = - \sum_{i=1}^{n_a} y_i \log(\hat{y}_i) \quad (8)$$

where $\hat{y}_i = \frac{\exp(Q_i)}{\sum_{j=1}^{n_a} \exp(Q_j)}$ represents the softmax probability of each action, computed from the Q-values predicted by the expert model. The detailed training setting is clarified in Section. IV.

3) *R-GAT*: The R-GAT [12] is the core component of the HGNN-EM, designed to aggregate heterogeneous features information to generate driving policies. The R-GAT extends the conventional graph attention network (GAT) by integrating multiple edge types, which enables more effective processing of heterogeneous graph structures.

First, attention logits c_{ij}^k are computed for each relation type $k \in \mathcal{K} = \{0 \text{ (aggressive)}, 1 \text{ (normal)}, 2 \text{ (conservative)}\}$ based on the corresponding node and edge features:

$$c_{ij}^k = [\mathbf{W}_E^k \cdot e_{ij}^k] \cdot [\mathbf{a}^k \text{LeakyReLU}(\mathbf{W}_N^k \cdot [x_i^k \| x_j^k])] \quad (9)$$

where $\|$ represents the concatenate operator. For the k_{th} relation type, x_i^k and x_j^k denote the feature of the i_{th} and j_{th} node, respectively; e_{ij}^k denote the edge value between the i_{th} and j_{th} node; $\mathbf{a}^k$ denote the learnable vector; $\mathbf{W}_N^k$ and $\mathbf{W}_E^k$ denote the learnable weight matrices for node feature and edge feature, respectively. Then, the attention values are normalized using the softmax function:

$$\alpha_{ij}^k = \frac{\exp(c_{ij}^k)}{\sum_{v \in \mathcal{V}_i} \exp(c_{iv}^k)} \quad (10)$$

Considering all the relation types within the intersection, the feature aggregation process of the R-GAT can be expressed as:

$$h_i^{(t)} = \sigma \left(\sum_{k \in \mathcal{K}} \sum_{v \in \mathcal{V}_i} \alpha_{iv}^k \mathbf{W}_N^k h_v^{(t-1)} \right) \quad (11)$$

where $h_i^{(t)}$ denotes the newly generated feature embedding of the i_{th} node, derived from the features of other nodes $h_v^{(t-1)}$ at the preceding time step. σ denotes the activation function. Finally, multi-head attention is introduced to further enhance the model learning capacity:

$$h_i^{(t)}(P) = \parallel_{p=1}^P h_{i,p}^{(t)} \quad (12)$$

where $h_i^{(t),p}$ denotes the feature embedding of the p_{th} head.

Algorithm 1 The computing procedure of the HGNN-EM

Input: Original heterogeneous graphical state space $G = [x_{AV}, N_A, N_N, N_C, D_A, D_N, D_C]$.

Output: Fused Q-values for AVs Q_{FIN} .

• **Pre-processing for the original input**

- 1: Calculate the fixed-dimensional node embeddings x'_{AV}, N'_A, N'_N , and N'_C through the node encoder.
- 2: Generate the learned edge values by equation (7).

• **Processing of the R-GAT**

- 3: Calculate attention logits c_{ij}^k for each type of relations by equation (9).
- 4: Calculate normalized attention values α_{ij}^k for each type of relations by equation (10).
- 5: Calculate the aggregated feature embedding $h_i^{(t)}$ for each node by equation (11).
- 6: Calculate the multi-head feature embedding $h_i^{(t)}$ for each node by equation (12).

• **Processing of the policy fusion model**

- 7: Calculate the fusion weight β through the fusion weight generator.
- 8: Calculate the Q_{GRL} through the policy network, taking the multi-head feature embedding as input.
- 9: Calculate the Q_{EXP} through the well-trained expert model, taking the heterogeneous graphical state space G as input.
- 10: Calculate the final Q values for AV by equation (13).

Return: Q_{FIN} .

4) *Policy Fusion Model*: The policy fusion model aims to integrate the GRL-based policy Q_{GRL} with the expert policy Q_{EXP} to enhance decision-making performance. It comprises two main components: a fusion weight generator and a policy network.

The fusion weight generator produces a learnable weight value $\beta \in [0, 1]$ based on the feature embedding generated by the R-GAT. This mechanism allows the weight values to be dynamically adjusted in response to the feature embeddings of heterogeneous traffic, thereby enhancing the adaptability and flexibility of the fused policy. The policy network further processes the feature embedding from the R-GAT to generate GRL-based policy. The final driving policy can be represented as follows:

$$Q_{\text{FIN}} = \beta \text{Softmax}(Q_{\text{GRL}}) + (1 - \beta) \text{Softmax}(Q_{\text{EXP}}) \quad (13)$$

where Q_{FIN} denote the final policy. The role of the softmax function is to ensure that the Q_{GRL} and Q_{EXP} are scaled consistently before policy fusion. The complete feature extraction procedure of the HGNN-EM is illustrated in Algorithm 1.

C. Policy Optimization

The Double DQN (DDQN) [13] algorithm is utilized as the DRL model to train the decision-making model. Two instances of HGNN-EM are initialized: the current Q-network θ for action value generation, and the target Q-network $\bar{\theta}$ for guiding policy updates. It is worth noting that the parameter of the well-trained expert model is fixed during training.

Initially, a T_r -step random exploration phase is implemented to broaden the exploration space of AV. Then, AV interacts with the heterogeneous traffic based on the linear decay ϵ — greedy strategy. The value of ϵ at each time step is determined using the following equation:

$$\epsilon_t = -\frac{(\epsilon_i - \epsilon_f)t}{T_e} + \epsilon_i \quad (14)$$

where ϵ_i is the initial exploration rate, ϵ_f is the final exploration rate, and T_e is the max exploration step. Then, the action-selecting strategy can be expressed as:

$$a_t = \begin{cases} \text{random}(\mathcal{A}), & P = \epsilon_t \\ \text{argmax}_a Q_{\text{FIN}}(s_t, a; \theta), & P = 1 - \epsilon_t \end{cases} \quad (15)$$

After performing the actions, the interaction trajectory (s_t, a_t, r_t, s_{t+1}) is recorded in a replay memory pool $\mathcal{D}$ with capacity $N_{\mathcal{D}}$. During the training phase, a random minibatch B comprising transitions (s_J, a_J, r_J, s_{J+1}) is sampled from $\mathcal{D}$. Subsequently, the training loss can be calculated as follows:

$$\begin{cases} \hat{y}_J = r_J + \gamma Q_{\text{FIN}}(s_{J+1}, \text{argmax}_a Q_{\text{FIN}}(s_{J+1}, a; \theta); \hat{\theta}) \\ L(\theta) = (\hat{y}_J - Q_{\text{FIN}}(s_J, a_J; \theta))^2 \end{cases} \quad (16)$$

Finally, the parameters of the two HGNN-EM instances are updated, thereby solving the optimal driving strategies:

$$\begin{cases} \theta \leftarrow \theta - \alpha \nabla_{\theta} L(\theta), & t \bmod T_c = 0 \\ \hat{\theta} \leftarrow \theta, & t \bmod T_t = 0 \end{cases} \quad (17)$$

where T_c and T_t denote the model update interval of the current Q-network and target Q-network, respectively. α is the learning rate.

IV. CASE STUDY

The proposed method is evaluated with a case study on a typical four-way intersection. It describes the scenario construction, basic settings, and experimental results.

A. Scenario Construction

Fig. 4 shows a four-way unsigned intersection constructed based on the SUMO [14] simulator with six HVs, evenly distributed across three driving styles: aggressive, normal, and conservative. HVs travel straight from designated starting areas to the opposite side. The AV's goal is to safely and efficiently navigate the intersection, accounting for varied HV behaviors. Furthermore, a set of key rules is established: 1) AV can obtain information on all HVs through its onboard sensors. 2) AV can directly acquire the driving characteristics of HVs.

The intelligent driver model (IDM) [15] and the MOBIL model [16] are utilized for longitudinal and lateral controls after decision-making, respectively. Parameters are customized to reflect varied driving styles among HVs, as detailed in Table I. Furthermore, the distinct behavioral patterns of HVs are incorporated to reflect varying levels of interactions at the intersection. Specifically:

- The aggressive HVs exhibit higher speed and disregard the right-of-way rules within the intersection, prioritizing speed and assertiveness over safety in conflict situations.
- The normal HVs adhere to the right-of-way rules within the intersection, striking a balance between safety and efficiency when conflicts exist. Their behaviors provide a baseline for interaction with the AV and other HVs.

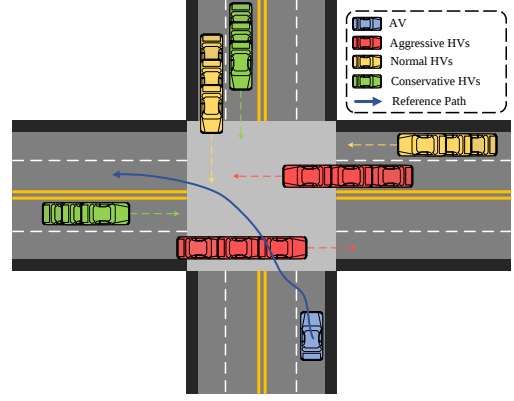


Fig. 4. The schematic of the typical four-way intersection.

TABLE I
PARAMETERS OF IDM FOR DIFFERENT STYLES OF HVs

Symbols	Description	Value		
		Agg.	Nor.	Con.
v (m/s)	Longitudinal speed	Variable		
Δv (m/s)	Relative speed	Variable		
s (m)	Distance from leader	Variable		
$a_{\max}$ (m/s ²)	Maximum acceleration	4.5	3.5	2.5
δ	Acceleration exponent	5	4	4
v^* (m/s)	Desired speed	20	16	12
s_0 (m)	Minimum gap from leader	1.2	1.6	2.0
T (s)	Safety time gap	1.0	1.5	2.0
b_{comf} (m/s ²)	Comfortable deceleration	2.0	2.0	2.0

- The conservative HVs prioritize safety and respond more cautiously to other vehicles, tending to wait for other vehicles to pass through the intersection first.

B. Experiment Setting

1) *Implementation Details*: The initial speed of all vehicles is set to 0 m/s. The observation time length T for the average acceleration is set to 5 time steps. The maximum permissible longitudinal speed $v_{\max}$ for AV is set to 20 m/s. The hyper-parameters for model training are listed in Table II.

2) *Training Details of the Expert Model*: A keyboard interface has been integrated into SUMO to allow expert drivers to maneuver AVs through intersections, generating a dataset of (G^i, a_{EXP}^i) pairs representing heterogeneous states and expert actions. The dataset consists of 5.5×10^5 samples, with 70% for training and 30% for testing.

The hyper-parameters for training the expert model are listed in Table III. During testing, the model is tested for 50,000 times. The accuracy of action generation is calculated, achieving a value of 99.23%, which underscores the effectiveness and reliability of the trained expert model in replicating expert-level decision-making.

3) *Evaluation Metrics*: For each episode, several evaluation metrics are introduced to evaluate the performance of the AV, specifically:

- **Reward**: The reward for each episode is recorded as the effective evaluation of the overall performance of the AV.

TABLE II
HYPER-PARAMETERS FOR MODEL TRAINING

Symbols	Description	Value
—	Simulation time step	0.1 s
M	Max training episodes	150
N_{test}	Number of testing episodes	50
T_M	Max time steps for each episode	300
T_r	Random exploration step	9000
T_c	Max exploration step	10000
ϵ_i	Initial exploration rate	0.5
ϵ_f	Final exploration rate	0.01
N_D	Replay memory pool capacity	1×10^6
B	Minibatch size	64
α	Learning rate	0.0005
T_c	Update interval for current Q-network	50
T_t	Update interval for target Q-network	5000
γ	Discount factor	0.99

TABLE III
HYPER-PARAMETERS FOR EXPERT MODEL TRAINING

Description	Value
Number of training epochs	500
Training batch size	32
Max training batches for each epoch	100
Optimizer	Adam
Learning rate	0.0005

TABLE IV
BASELINES FOR COMPARATIVE EXPERIMENT

Model	Graph Representation	Backbone
GCQ [6]	Homogeneous graph	GCN
HGNN	Heterogeneous graph	HGNN
STGCN [10, 17]	Homogeneous graph	GCN, TCN
Proposed	Heterogeneous graph	HGNN-EM

- **Collision number:** The count of collisions performed by the ego AV is recorded as a safety assessment.
- **Average speed:** The average speed (m/s) of AV during driving is computed to evaluate driving efficiency.
- **Traveling time:** The cumulative time (s) for the AV to exit the intersection is recorded to evaluate the efficiency of task accomplishment.

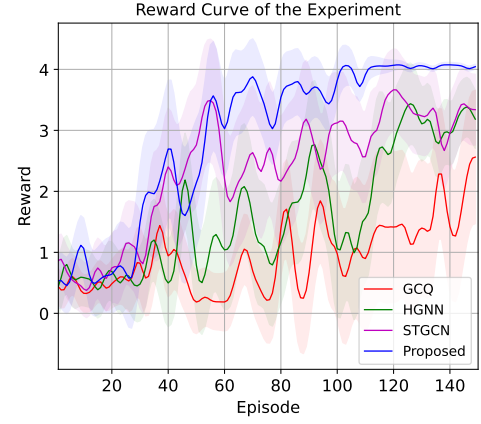
For model testing, one metric is additionally established, specifically:

- **Computational efficiency:** The average computing time (ms) per simulation step is measured to assess the computational efficiency. The test device is a laptop with 32GB RAM, i9-10980HK, RTX4070s.

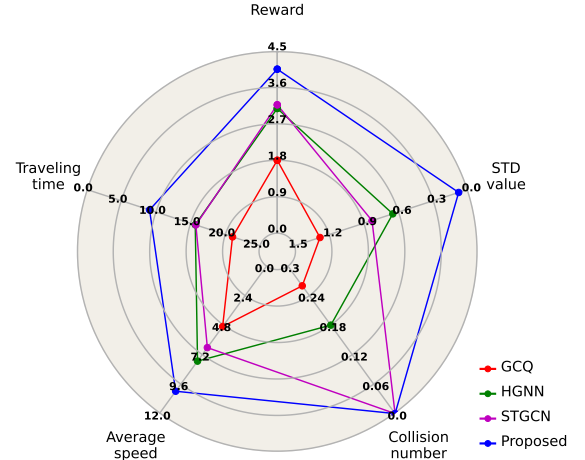
C. Experimental Results and Analysis

1) *Baselines:* To assess the performance of our proposed method, we compare it with several state-of-the-art baseline models, all calibrated to match the decision-making problem formulation in Fig. 2. Details of these models are summarized in Table IV.

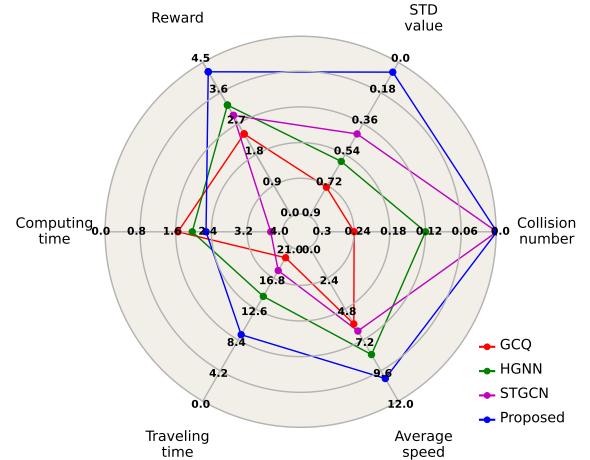
2) *Results and Analysis:* During training, reward curves for three random seeds are plotted in Fig. 5(a), with final results averaged after convergence (around 120 episodes). Additionally, the convergent reward trajectories are recorded to compute the standard deviation (STD) to reflect the model's



(a) Reward curves of all models.



(b) Training results of all models.



(c) Testing results of all models.

Fig. 5. The results of the experiment. The shaded areas in the reward figure show the standard deviation for 3 random seeds.

stability. These training results are displayed in Fig. 5(b). During testing, each model yields three decision-making instances, and the averaged testing results are summarized in Fig. 5(c).

As shown in Fig. 5(a), all models finally converge to relatively stable reward intervals, signifying the effectiveness of the training process. As seen in Fig. 5(b), the traveling times for all models remain within the max time steps limitation

($T_M = 300$ steps = 30 s) after achieving convergence. This demonstrates that all models are capable of successfully navigating the intersection. However, the GCQ model exhibits inferior performance across all evaluation metrics, underscoring the limitations of relying solely on graphic techniques without additional enhancements in heterogeneous traffic scenarios.

The HGNN model exhibits superior overall performance compared to the GCN model, particularly evident in the lower STD value. This emphasizes that the incorporation of heterogeneous technologies into the graph representation and the GNN significantly optimizes the decision-making performance and model adaptability in heterogeneous traffic. Nonetheless, there is room for improvement in the driving safety aspect of the HGNN model. Compared to the HGNN model, the STGCN model performs better in STD value and driving safety (zero collisions). This highlights the importance of incorporating temporal networks, facilitating more consistent and stable behaviors, thereby enhancing driving safety and model stability in heterogeneous traffic.

Notably, the proposed method achieves zero collisions and outperforms other models in terms of reward, average speed, and traveling time, showcasing its strong ability to generate safe and efficient driving strategies in heterogeneous traffic. The lower STD value further underscores its adaptability to the varied driving styles of HVs in heterogeneous traffic.

As seen in Fig. 5(c), the testing results align closely with the converged training outcomes depicted in Fig. 5(b), further validating the effectiveness of the model training process. The STGCN model exhibits a noticeable increase in computing time, highlighting that the integration of spatial-temporal techniques significantly elevates data processing demands, thereby reducing computational efficiency. In contrast, the computing time of the proposed method remains comparable to that of the GCQ and HGNN models while delivering superior overall decision-making performance. This emphasizes that the proposed method effectively balances performance gains with efficiency consideration.

V. CONCLUSION

This paper introduces a combined approach of heterogeneous GRL and an expert system to enhance AVs' decision-making in heterogeneous traffic. Specifically, a heterogeneous graph representation is proposed to capture intricate interactions, while an HGNN-EM is established to encode the heterogeneous information and generate expert-level driving behaviors. Experiments on a typical four-way intersection demonstrate the superior performance of the proposed method. Future work will focus on predicting the driving styles of HVs and validating our approach across a broader range of traffic scenarios with diverse settings, thereby further enhancing the adaptability and robustness of the proposed method.

REFERENCES

- [1] S. Bae, D. Isele, A. Nakhaei, P. Xu, A. M. Añon, C. Choi, K. Fujimura, and S. Moura, "Lane-change in dense traffic with model predictive control and neural networks," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 2, pp. 646–659, 2023.
- [2] Q. Liu, X. Li, S. Yuan, and Z. Li, "Decision-making technology for autonomous vehicles: Learning-based methods, applications and future

- outlook," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 30–37.
- [3] S. Fang, P. Hang, C. Wei, Y. Xing, and J. Sun, "Cooperative driving of connected autonomous vehicles in heterogeneous mixed traffic: A game theoretic approach," *IEEE Transactions on Intelligent Vehicles*, pp. 1–15, 2024.
- [4] K. Chen, B. Li, R. Zhang, and X. Cheng, "Autonomous intersection management with heterogeneous vehicles: A multi-agent reinforcement learning approach," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 2255–2260.
- [5] J. Liu, W. Zhao, C. Wang, C. Xu, L. Li, Q. Chen, and Y. Lian, "Eco-friendly on-ramp merging strategy for connected and automated vehicles in heterogeneous traffic," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 11, pp. 13 888–13 900, 2023.
- [6] S. Chen, J. Dong, P. Ha, Y. Li, and S. Labi, "Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 7, pp. 838–857, 2021.
- [7] D. Xu, P. Liu, H. Li, H. Guo, Z. Xie, and Q. Xuan, "Multi-view graph convolution network reinforcement learning for cavs cooperative control in highway mixed traffic," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2588–2599, 2024.
- [8] P. Cai, H. Wang, Y. Sun, and M. Liu, "Dq-gat: Towards safe and efficient autonomous driving with deep q-learning and graph attention networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [9] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.
- [10] J. Li, D. Isele, K. Lee, J. Park, K. Fujimura, and M. J. Kochenderfer, "Interactive autonomous navigation with internal state inference and interactivity estimation," *IEEE Transactions on Robotics*, vol. 40, pp. 2932–2949, 2024.
- [11] Z. Li, J. Gong, Z. Zhang, C. Lu, V. L. Knoop, and M. Wang, "Interactive behavior modeling for vulnerable road users with risk-taking styles in urban scenarios: A heterogeneous graph learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 8, pp. 8538–8555, 2024.
- [12] M. Chen, Y. Zhang, X. Kou, Y. Li, and Y. Zhang, "R-gat: relational graph attention network for multi-relational graphs," *arXiv preprint arXiv:2109.05922*, 2021.
- [13] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
- [14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 2575–2582.
- [15] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug 2000. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.62.1805>
- [16] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.
- [17] Q. Liu, Y. Tang, X. Li, F. Yang, X. Gao, and Z. Li, "Sif-stgdan: A social interaction force spatial-temporal graph dynamic attention network for decision-making of connected and autonomous vehicles," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 376–383.