A Hierarchical Reinforcement Learning Framework with Spatial-Temporal Graph Attention for Autonomous Driving Decision-Making and Control

Wen-Te Hsiao, Jhan-Yu Liao, Yu-Chen Lin, *Senior Member*, *IEEE,* Wen-Hui Chen, Zhi-Cheng Yang

***Abstract*—This paper presents a hierarchical framework that integrates spatial-temporal graph attention network (ST-GAT) with reinforcement learning for decision and control of autonomous driving. Inspired by the principles of human cognition, the framework decomposes the driving task into two complementary levels: a high-level trajectory planning module that utilizes the soft actor-critic (SAC) algorithm within the Frenet coordinate system, and a low-level tracking control module based on the worst-case soft actor-critic (WCSAC) strategy. This hierarchical decomposition improves policy stability and sample efficiency by decoupling strategic trajectory planning from reactive control execution. Unlike previous methods, the proposed ST-GAT module enables explicit scene understanding by modeling surrounding vehicles and their interactions as a spatial-temporal graph structure. Through attention-based aggregation, the system dynamically captures road geometry and agent behaviors directly from online sensor observations, enabling mapless situational reasoning. Experimental results show that the proposed framework achieves success rates of 92.33% and 97.00% in the roundabout and five-way intersection scenarios in the CARLA simulator.**

I. Introduction

Autonomous driving (AD) technology has made substantial progress over the past decade, transitioning from academic research to industrial deployment. Despite these advances, achieving reliable and robust autonomy in complex urban environments remains a significant challenge due to three primary factors. First, urban areas exhibit highly heterogeneous and irregular road topologies, including roundabouts, multi-lane arterials, and signalized intersections, characterized by varying traffic densities and geometric configurations. Second, interactions among multiple heterogeneous road agents are inherently complex and stochastic, making explicit modeling of all possible interaction patterns computationally intractable [1]. Third, the ego vehicle must continuously execute intelligent decisions under uncertainty while balancing two often competing objectives: safety (collision avoidance) and operational efficiency (timely goal achievement).

This work was supported by National Science and Technology Council, Taiwan, R.O.C., under the Grant NSTC 114-2218-E-035-001 and 112-2628-E-035-001-MY3. (Corresponding author: Yu-Chen Lin).

Wen-Te Hsiao, Jhan-Yu Liao, Yu-Chen Lin, and Zhi-Cheng Yang are with the Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan, R.O.C (e-mail: wender910402@gmail.com, lcomebest@gmail.com, yuchlin@fcu.edu.tw, a01010508s@gmail.com).

Wen-Hui Chen is with the Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan (e-mail: whchen@ntut.edu.tw).

In recent years, deep learning has emerged as a powerful paradigm for addressing these challenges in autonomous driving systems. By learning driving policies directly from large-scale datasets, deep models alleviate the substantial engineering effort required to manually design rules for all feasible scenarios, making them particularly well-suited for high-dimensional, nonlinear, and temporally dynamic environments [2-4]. Most existing approaches rely on supervised imitation learning (IL) to efficiently extract human driving expertise from demonstration data. However, IL is susceptible to dataset bias [4] and distribution mismatch (covariate shift) [6], which constrain its robustness when encountering out-of-distribution scenarios. Deep reinforcement learning (DRL) offers an alternative paradigm, enabling agents to learn optimal policies through environment interaction and trial-and-error exploration [6-10]. Nevertheless, current DRL-based AD frameworks are predominantly designed for constrained or simplified scenarios, such as single-lane highways [8], isolated unsignalized intersections [9], or single-entry roundabouts [10], often with minimal consideration of realistic multi-agent traffic dynamics. Moreover, these methods typically lack explicit mechanisms for modeling spatial-temporal dependencies among heterogeneous road agents and interpreting dynamic scene structures from sensor observations. Consequently, their generalizability to diverse, complex urban driving contexts remains limited.

To address the limitations, this paper proposes a decision and control framework for autonomous driving that integrates spatial–temporal graph attention networks (ST-GAT) with Hierarchical reinforcement learning (HRL). At the spatial level, the surrounding traffic scene, including the ego vehicle and nearby agents, is represented as a graph, where each node encodes the state of an individual agent and edges capture their interactions. The ST-GAT module implicitly infers environmental intent understanding by extracting dynamic features across both time and space, enabling the system to reason about scene structures without relying on explicit map priors. Based on the extracted latent scene representations, the high-level policy determines a planned trajectory that reflects both the current environment and the ego vehicle's driving state. This trajectory serves as the target for the low-level control layer. Within the HRL architecture, the low-level controller receives the planned trajectory and converts it into relative ego-centric tracking states, which greatly simplifies the control objective. In parallel, the low-level module also receives the high-level latent scene features via feature concatenation, ensuring that reactive control remains aware of evolving environmental intent and can respond appropriately

to unexpected events. In summary, the main contributions of this research are as follows:

1) **A hierarchical architecture for mapless autonomous driving:** The framework decouples high-level path planning from low-level control. The upper layer uses ST-GAT to implicitly infer road topology and traffic interactions directly from online sensor observations **without requiring HD maps**.
2) **Reinforcement learning with Frenet-frame planning and WCSAC-based control:** The Frenet frame simplifies planning by decoupling longitudinal and lateral motion. At the lower level, Worst-Case Soft Actor-Critic (WCSAC) incorporates a safe critic to enforce collision-risk constraints, ensuring reliable, safe control execution.
3) **Demonstrated superior performance in CARLA simulation:** The method attains success rates of 92.33% and 97.00% in CARLA's roundabout and five-way intersection scenarios, showcasing stable and collision-free behavior in complex environments.

II. Related Work

A. Dynamic Object Trajectory Prediction

With the rapid advancement of artificial intelligence and deep learning, trajectory prediction has achieved significant progress in recent years. Most studies focus on accurately forecasting the future trajectories of surrounding agents in dynamic traffic environments to support autonomous driving decision-making and motion planning. Accurate trajectory prediction is essential for improving driving safety, collision avoidance, and interaction awareness in complex urban scenarios. Recently, Park *et al*. [11] employed a Long Short-Term Memory (LSTM) network combined with an Encoder–Decoder architecture for vehicle trajectory sequence prediction. Zyner *et al*. [12] used a Recurrent Neural Network (RNN) as a temporal encoder together with a Mixture Density Network (MDN) to predict multimodal intersection trajectories and driving intentions. However, traffic scenes involve not only temporal dynamics but also rich spatial interactions. To better capture spatial relationships between vehicles and their surroundings, Su *et al*. [13] utilized a Convolutional Neural Network (CNN) for feature extraction, modeling vehicles and obstacles as images to predict future trajectories. While CNN effectively capture local spatial features, their grid-structured input limits their applicability to real-world traffic scenarios with irregular node distributions and complex relationships. To address this limitation, researchers have turned to Graph Neural Network (GNN) and their variants, such as Graph Attention Network (GAT) [14], for trajectory modeling. GNN represent traffic environments as graphs composed of nodes (vehicles, pedestrians, obstacles) and edges (interaction relationships), enabling efficient modeling of dynamic interactions under irregular topologies. Furthermore, Huang *et al*. [15] further combined raw features with those processed by a GAT module and used temporal encoding to learn spatiotemporal dependencies, achieving more accurate trajectory prediction results.

B. Autonomous Vehicle Decision-Making and Control

Deep learning has proven highly effective in addressing decision-making problems in high-dimensional, dynamic, and nonlinear environments. Many studies adopt Imitation Learning (IL) to acquire driving knowledge by learning from expert behavior; however, IL suffers from data bias and poor generalization. To improve generalization, Codevilla *et al*. [5] introduced residual structures, but their supervised behavior cloning approach remained limited by inefficient data augmentation and diminishing returns. To mitigate bias and reduce reliance on expert data, later works combined IL with Reinforcement Learning (RL), leveraging RL's trial-and-error capability to learn policies autonomously. Cai *et al*. [7] used IL to establish an environment model and initial policy, then applied RL for optimization, accelerating early training convergence. Cao *et al*. [16] proposed a Hierarchical Architecture, where the high-level module (based on RL) handles decision-making, and the low-level module (trained with IL) manages control, combining both advantages. Although this approach reduced decision bias, the control layer still relied on expert data. In RL algorithm development, [17] improved the traditional DQN by introducing a Dueling Network structure that separates the state-value and advantage functions to enhance value estimation accuracy. [18] proposed Deep Deterministic Policy Gradient (DDPG), combining DQN and Policy Gradient methods for stable policy learning in continuous action spaces.

Inspired by Cognitive Science and Theory of Mind, hierarchical design is now viewed as an effective means to simplify learning objectives and improve decision precision. Qiao *et al*. [19] modeled autonomous driving within a HRL framework, where the high-level "option network" selects strategies and the low-level "action network" executes control actions, using state attention and a custom reward function to address sparse rewards. Cai *et al*. [20] integrated Graph Neural Networks (GNNs) with RL to encode latent features of vehicle nodes in a bird's-eye view, where the RL model outputs target speeds and a PID controller generates control commands. Yoo *et al*. [21] employed a Graph Convolutional Network (GCN) for high-level environment understanding and a low-level action network for control, demonstrating a clearer hierarchical structure. Overall, combining hierarchical architectures with reinforcement learning effectively models the human "thinking–acting" process, enhancing decision stability in complex environments. Building on this concept, this paper proposes an autonomous driving decision and control system based on ST-GAT and HRL.

III. Proposed Method

The proposed autonomous driving decision-making and control system integrates spatial–temporal attention mechanisms with hierarchical reinforcement learning (HRL). As illustrated in Fig. 1, the overall framework is composed of three core modules. First, the spatial-temporal graph attention-based (ST-GAT) dynamic object trajectory prediction module captures the spatial interactions and temporal dependencies of surrounding agents to forecast their future states. Following this, the collision risk–aware constraint module evaluates potential hazards based on these

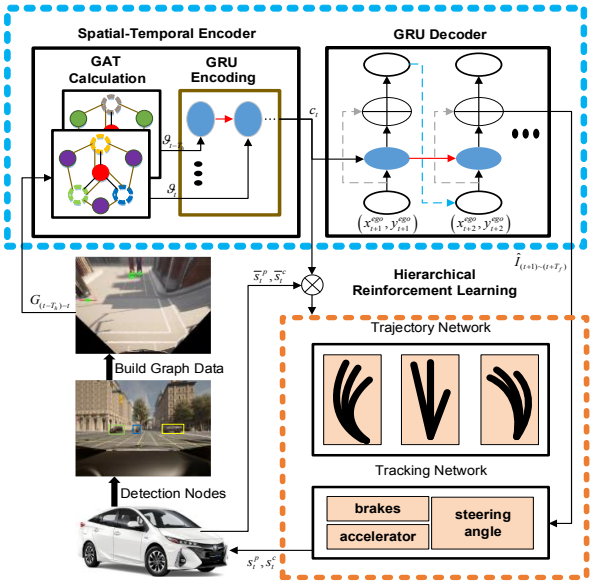


Figure 1. System architecture of the proposed autonomous driving decision-making and control framework based on ST-GAT and HRL.

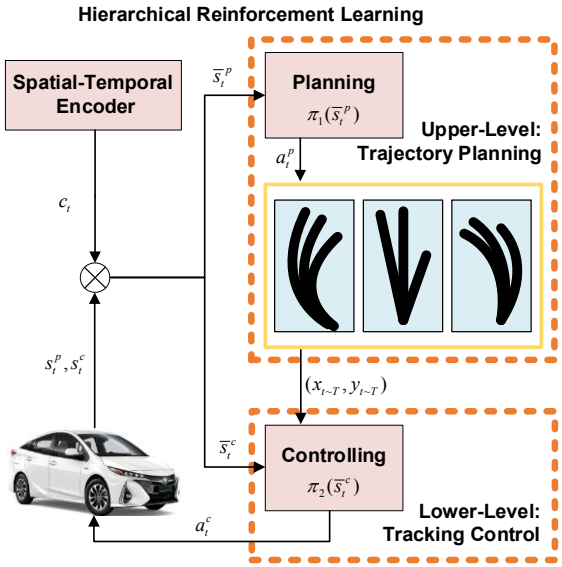


Figure 2. Architecture of HRL-based decision-making and control design

predictions, formulating safety boundaries to prevent unsafe actions. Finally, the ego-vehicle decision and control module executes hierarchical decision-making to optimize driving policies within these safety constraints. This modular design enables effective perception, risk assessment, and hierarchical decision-making in complex traffic scenarios.

A. ST-GAT Dynamic Object Trajectory Prediction Module

Inspired by human perceptual cognition, the system performs structured environment understanding by modeling surrounding traffic participants and their interactions as a spatial–temporal graph, following [15]. Each detected vehicle is represented as a graph node, with spatial and temporal relationships encoded as edges. An attention-based aggregation mechanism adaptively weights neighboring vehicles, enabling effective modeling of multi-agent interactions and dynamic traffic patterns. First, each object's feature vector $\vec{h} = \left[x, y, \varphi, v^x, v^y, a^x, a^y, \omega^x, \omega^y, p\right]^T$ includes its world coordinates (x, y), yaw angle φ, velocity (v^x, v^y), acceleration (a^x, a^y), angular velocity (ω^x, ω^y), and driving direction p. By encoding latent spatial and temporal interactions among all graph nodes, the system attains an understanding of the surrounding environment. Spatial dependencies are modeled using a graph attention network (GAT), while temporal dynamics are captured via gated recurrent units (GRU). In the ST-GAT module shown in Fig. 1, the network input is $G_{(t-T_h)\sim(t)}$, which consists of object feature vector $\vec{h}$, defined as

$$G_{(t-T_h)\sim(t)} = \begin{Bmatrix} \vec{h}^{\,1}_{t-T_h} & \cdots & \vec{h}^{\,m}_{t} \\ \vdots & \ddots & \vdots \\ \vec{h}^{\,1}_{t-T_h} & \cdots & \vec{h}^{\,m}_{t} \end{Bmatrix}, \; m = 1, \ldots, N \tag{1}$$

where t denotes the current time, $t - T_h$ corresponds to the past observation period. Subsequently, the GAT extracts spatial state features distributed among nodes, as detailed below.

$$\vartheta^i_{(t-T_h)\sim(t)} = f_{gat}\left(G_{(t-T_h)\sim(t)}\right) \tag{2}$$

where $\vartheta^i_{(t-T_h)\sim(t)}$ represents the attention weights at every node i from time $t - T_h$ to the current time t; f_{gat} denotes the graph attention function, which computes the attention weights. Afterwards, these features are passed through a GRU to capture temporal dependencies and obtain temporal feature encoding. The temporal relationship can be expressed as follows.

$$c_t = f_{enc}\left(\vartheta^i_{(t-T_h)\sim(t)}\right) \tag{3}$$

where c_t denotes the spatial-temporal feature representation, which serves as the latent representation of the environment and f_{enc} represents the temporal encoder function. The extracted spatial-temporal features are subsequently decoded to predict future trajectories, as detailed below.

$$\hat{I}_{(t+1)\sim(t+T_f)} = f_{dec}\left(c_t\right) \tag{4}$$

where f_{dec} denotes the temporal decoder function, and $\hat{I}_{(t+1)\sim(t+T_f)}$ represents the predicted future trajectory, which also reflects the anticipated state of the environment.

B. HRL-Based Ego-Vehicle Decision and Control

A hierarchical reinforcement learning (HRL) framework is employed for autonomous vehicle decision-making and control, where the overall reward is decomposed into an upper-level reward for path planning and a lower-level reward for trajectory tracking. This hierarchical formulation for trajectory tracking. This hierarchical formulation decouples learning objectives and mitigates the reward sparsity issue

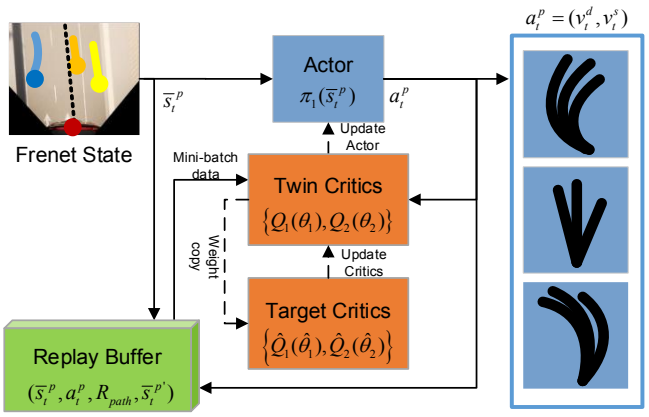


Figure 3. Flowchart of SAC-Based Actor-Critic Updates for Path Planning (Upper-Level)

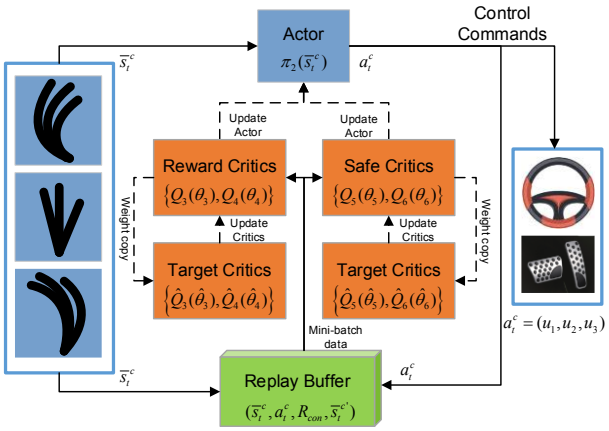


Figure 4. Flowchart of Actor and Critic Updates in the WCSAC-based Tracking Control Network (Lower-Level)

encountered in conventional reinforcement learning. Inspired by human reasoning, the upper layer performs path planning in the Frenet coordinate system using latent environmental features and ego-vehicle states to generate a reference trajectory, while the lower layer tracks this trajectory by minimizing deviations from target waypoints. The HRL architecture is illustrated in Fig. 2.

The input states for the path-planning and tracking-control layers are denoted as s_t^p and s_t^c, respectively, and are expressed as follows.

$$s_t^p = \left[x_t^s,\ y_t^d,\ v_t^{ego},\ a_t^{ego},\ \varphi_t^{ego},\ \upsilon_t^1,\ \upsilon_t^2,\ \upsilon_t^3 \right]^T \tag{5}$$

For the ego vehicle, the path-planning state includes its position in the Frenet coordinate system $(x_t^s,\ y_t^d)$, the current velocity v_t^{ego}, acceleration a_t^{ego}, yaw angle φ_t^{ego}, turn signals for left υ_t^1 and right υ_t^2 and the route ID υ_t^3. Subsequently, the tracking-control state s_t^c incorporates the current velocity v_t^{ego}, as well as the current and predicted values for lateral distance $\left(D_t^{ego}, D_t^{pre}\right)$ and yaw deviation $\left(\varphi_t^{ego}, \varphi_t^{pre}\right)$, defined as

$$s_t^c = \left[v_t^{ego},\ D_t^{ego},\ \varphi_t^{ego},\ D_t^{pre},\ \varphi_t^{pre} \right]^T \tag{6}$$

The original input data are combined with the latent feature c_t to enhance environmental understanding for both the decision and control modules. The fused feature representations $\overline{s}_t^p$ and $\overline{s}_t^c$ are defined as follows:

$$\overline{s}_t^p = concate\left(s_t^p,\ c_t\right),\ \ \overline{s}_t^c = concate\left(s_t^c,\ c_t\right) \tag{7}$$

where $\otimes$ refers to the vector concatenation operator in Fig. 2.

Based on the fused feature representation $\overline{s}_t^p$, the path-planning process is initiated. As detailed in Fig. 3, the upper-level policy generation produces an output defined as $a_t^p = \left(v_t^d,\ v_t^s\right)$, representing the predicted lateral endpoint position v_t^d and the longitudinal target velocity v_t^s. This can be expressed as:

$$\left(v_t^d, v_t^s\right) = \pi_1\left(\overline{s}_t^p\right) \tag{8}$$

where π_1 denotes the policy network of the path-planning layer. Using v_t^d and v_t^s, the predicted path $\left(x_{t\sim T},\ y_{t\sim T}\right)$ can be generated.

Ultimately, the policy network predicts the future terminal position v_t^d and target velocity v_t^s based on the current state, thereby generating a complete planned trajectory $\left(x_{t\sim T},\ y_{t\sim T}\right)$.

With the reference trajectory established, the system proceeds to the lower-level execution phase. As illustrated in Fig. 4, the tracking-control layer output is given by $a_t^c = \left(u_t^1,\ u_t^2,\ u_t^3\right)$, where u_t^1, u_t^2, u_t^3 denote the steering angle, throttle, and turn signal, respectively. The corresponding formulation is defined as

$$\left(u_t^1,\ u_t^2,\ u_t^3\right) = \pi_2\left(\overline{s}_t^c\right) \tag{9}$$

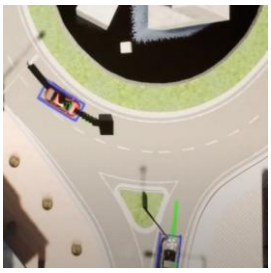
where, π_2 represents the policy network of the tracking-control layer. Trained via the reward function, the network is optimized to generate precise control commands. Finally, the generated steering angle u_t^1, throttle u_t^2, and turn signal u_t^3 are applied to the vehicle, completing the closed-loop interaction with the environment and thereby establishing an integrated autonomous driving decision and control framework.

This paper selects WCSAC for the tracking-control module primarily because, unlike the path-planning layer where constraints can be explicitly defined, the operational constraints at the control level are often difficult to formulate deterministically. Consequently, WCSAC is employed to regulate tracking behavior through a 'soft' constraint mechanism by incorporating estimated risk values, rather than relying on rigid rules. To explicitly enforce these safety bounds, the WCSAC algorithm maintains two distinct critics: a Reward Critic Q_{reward} and a Safe Critic Q_{safe}. The policy network is updated by maximizing the expected return while

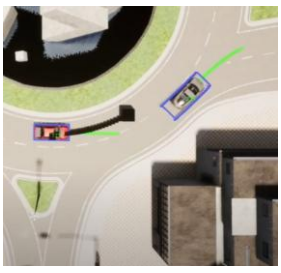
(a)

(b)

Figure 5. The ego-vehicle traversing the five-way intersection



(a)



(b)

Figure 6. The ego-vehicle traversing the roundabout

strictly satisfying a cost threshold limit. Specifically, the policy loss function is formulated as:

$$L(\pi) = E_{s\sim D}\left[\alpha \log \pi(a \mid s) - \min_{i=1,2} Q_{reward,i} + \beta \max(0, \min_{j=1,2} Q_{safe,j} \text{-d})\right] \quad (10)$$

where α is the entropy temperature regulating exploration, β is a weight multiplier for the safety constraint, and d is the predefined risk threshold. Furthermore, the collision risk-aware constraint module provides an explicit safety signal to the tracking controller. By utilizing the predicted trajectories from the ST-GAT module, the system continuously checks for future spatial intersections between the bounding box of the ego-vehicle (B_{ego}) and those of neighboring vehicles ($B_{neighbor}$). If a potential intersection is detected ($B_{ego} \cap B_{neighbor} \neq \theta$), an immediate risk penalty proportional to the inverse of the relative distance is assigned to the Safe Critic. This mechanism enables the low-level controller to proactively execute emergency braking or evasive maneuvers before a physical collision occurs, bridging the gap between high-level intent understanding and low-level safe execution.

C. Reward Function Design

The reward function in this paper is divided into two parts, corresponding to the path-planning layer and the tracking-control layer. The reward function for the path-planning layer is first introduced, as follows.

$$R_{path} = r_{lan1} + r_{lane2} + r_{lane3} + r_{lane4} + r_{dis} \quad (11)$$

where R_{path} represents the overall reward for the path-planning layer. The term r_{lane1} denotes the reward for successful lane keeping, while r_{lane2} rewards necessary lane changes, determined by the variation in vehicle speed. The term r_{lane3} evaluates the safety of a lane change, assessing whether the maneuver might lead to potential collisions or excessive deviations from the current or planned route. The term r_{lane4} serves as a penalty for frequent lane changes, since excessive changes may increase the likelihood of accidents. In addition, to maintain smooth driving behavior, this penalty is weighted accordingly. Finally, r_{dis} represents the distance-tracking reward, which encourages the vehicle to follow the planned trajectory accurately by minimizing the positional deviation. The fundamental justification for this upper-level reward formulation is to closely mimic human driving cognitive processes. Specifically, the speed variation threshold within r_{lane2} is designed to prioritize stable lane-keeping when the surrounding traffic flows smoothly, thereby actively preventing unnecessary and risky lane-change maneuvers. Furthermore, the safety evaluation term r_{lane3} serves as a critical hard constraint, applying massive penalties to strictly prohibit the high-level planner from generating any trajectories that would lead to unavoidable collisions or kinematically infeasible paths. Next, the reward function for the tracking-control layer is introduced, as follows.

$$R_{con} = r_{suc} + r_v + r_{steer} + r_{yaw} + r_{lat1} + r_{lat2} + r_{lat3} \quad (12)$$

where R_{con} represents the overall reward for the tracking-control layer. The term r_{suc} denotes the reward for completing the trajectory, encouraging the vehicle to follow the planned path. The term r_v corresponds to the speed-related reward, which promotes maintaining an appropriate and stable velocity rather than abrupt stops or excessive acceleration. The term r_{steer} relates to the steering angle, while r_{yaw} reflects the predicted yaw angle error; together, they encourage smooth and stable steering control. The term r_{lat1} represents the reward associated with lateral velocity, r_{lat2} and r_{lat3} correspond to the lateral distance deviation and its rate of change, respectively. These three lateral-related reward components are designed primarily to reduce vehicle oscillations and ensure stable trajectory tracking performance. The rationale behind decomposing the lateral tracking reward into lateral velocity (r_{lat1}), position error (r_{lat2}), and the rate of change of lateral distance (r_{lat3}) is to effectively suppress the 'ping-pong' oscillation effect commonly observed in naive path-following agents. By heavily penalizing the rapid fluctuation of the lateral distance, the controller is forced to ensure not only accurate waypoint tracking but also ride smoothness and passenger comfort through gradual, stable steering corrections. Additionally, the collision impact penalty allows the model to learn the severity of different failure states, pushing the policy towards safer regions during the WCSAC training phase.

I. Experiments and Results

This study evaluates the efficacy of the proposed autonomous navigation system in complex traffic environments. The primary performance metric is the success rate, defined as the ratio of collision-free episodes to the total number of episodes. Experiments were conducted in roundabout and five-way intersection scenarios using the CARLA Simulator (version 0.9.11).

To demonstrate the system's capability in handling dynamic interactions, several driving maneuvers were analyzed. As illustrated in Fig. 5(a), while traversing the five-way intersection, the ego-vehicle detects a nearby vehicle and executes an emergency braking command, successfully clearing the intersection as shown in Fig. 5(b). Similarly, in the roundabout scenario shown in Fig. 6(a), the ego-vehicle proactively decelerates upon detecting another vehicle entering the roundabout to avoid a potential collision. Results from 300 test episodes show that the five-way intersection scenario achieved a higher success rate than the roundabout scenario, mainly due to its simpler single-lane initialization. Overall, the proposed framework achieved competitive performance in both scenarios. Detailed comparative results are presented in Table 1.

TABLE 1.
COMPARISON RESULTS WITH OTHER STUDIES AT ROUNDABOUTS AND FIVE-WAY INTERSECTIONS

Models	**Roundabout Success Rate (%)**	**5-way Success Rate (%)**
H-REIL[8]	28.33%	85.00%
IL-Conser[12]	55.67%	80.00%
IL-Aggr[12]	26.67%	83.00%
DQ-GAT [20]	87.33%	99.67%
FSM-TTC[14]	87.67%	98.67%
GAT-HRL(Ours)	92.33%	97.00%

To further examine the limitations of the proposed framework, we analyzed the unsuccessful episodes observed during the simulation experiments. Blind-spot collisions caused most failures. Since the proposed system relies on a single front-facing camera, its perception range is limited by the restricted field of view. As a result, the ego vehicle may fail to detect vehicles rapidly approaching from the rear or in lateral blind spots, especially during high-speed merging in roundabouts. This limitation explains why the roundabout's success rate is slightly lower than that of the five-way intersection scenario.

A smaller number of failures were associated with rule-induced deadlocks. For instance, when a preceding vehicle illegally stopped and completely blocked the lane, the high-level planner tended to remain stopped because the available bypass maneuver would require crossing a solid lane marking. Although this behavior reflects strict rule compliance, it also reduces navigation flexibility in abnormal traffic situations. These results indicate that future work should incorporate exception-handling mechanisms, such as risk-aware temporary bypass decisions, better to balance traffic-rule compliance, safety, and task completion.

II. CONCLUSION

Overall, this paper establishes an HRL framework that separates decision-making and control into two layers, thereby simplifying the learning objectives and improving overall system performance. The decision module in the upper layer focuses on path planning, while the control module in the lower layer executes throttle, brake, and steering actions according to the planned trajectory. The proposed method does not rely on any map information, making decisions solely based on road rules and the ego-vehicle's state. The required input features can be easily obtained from a single monocular camera, giving the system a low-cost and practically deployable advantage. Experimental results in the CARLA simulator verify the effectiveness of the proposed framework in complex traffic scenarios.

Future work will optimize the model for embedded deployment and real vehicle testing. To overcome the limited field of view of a single front camera, a panoramic vision system will monitor surrounding environments. In addition, a risk-constrained module will improve the control layer's response to anomalies, enhancing safety and reliability.

REFERENCES

[1] Y. Chen, C. Liu, B. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robot. Autom. Lett.*, vol. 5, pp. 2754–2761, 2020.

[2] P. Cai, S. Wang, Y. Sun, and M. Liu, "Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion," *IEEE Robot. Autom. Lett.*, vol. 5, pp. 4218–4224, 2020.

[3] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Proc. 3rd Conf. Robot Learn.*, 2019.

[4] F. Codevilla, *et al.*, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1–9.

[5] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9329–9338.

[6] P. Cai, *et al.* "Vision-based autonomous car racing using deep imitative reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7262–7269, 2021.

[7] P. Cai, *et al.*, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1247–1254, 2020.

[8] A. Kendall *et al.*, "Learning to drive in a day," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 8248–8254.

[9] E. Leurent and J. P. Mercat, "Social attention for autonomous decision making in dense traffic," arXiv:1911.12250, 2019.

[10] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *Proc. IEEE Int. Trans. Syst. Conf.*, 2019, pp. 2765–2771.

[11] S. H. Park, *et al.*, "Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture," in *Proc. IEEE Int. Conf. Vehicles Symposium*, Changshu, China, 2018, pp. 1672-1678.

[12] A. Zyner, S. Worrall and E. Nebot, "Naturalistic Driver Intention and Path Prediction Using Recurrent Neural Networks," *IEEE Trans. Intell. Trans. Syst.*, vol. 21, no. 4, pp. 1584-1594, April 2020.

[13] Z. Su, *et al.*, "Convolutions for Spatial Interaction Modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. and Pattern Recognit.*, New Orleans, LA, USA, pp. 6583-6592, 2022.

[14] P. Veličković, *et al.*, "Graph Attention Networks," *arXiv preprint* arXiv:1710.10903, 2018.

[15] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Korea (South), pp. 6271-6280, 2019.

[16] Z. Cao, *et al.*, "Reinforcement Learning based Control of Imitative Policies for Near-Accident Driving," *arXiv preprint* arXiv:2007.00178, pp. 1-10, 2020.

[17] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, Phoenix, Arizona, USA, Vol. 30, No. 1, 2016.

[18] T. P. Lillicrap, *et al.,* "Continuous Control with Deep Reinforcement Learning," *arXiv preprint* arXiv:1509.02971, 2019.

[19] Z. Qiao, *et al.*, "Hierarchical Reinforcement Learning Method for Autonomous Vehicle Behavior Planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst*., Las Vegas, NV, USA, pp. 6084-6089, 2020.

[20] P. Cai, *et al.*, "DQ-GAT: Towards Safe and Efficient Autonomous Driving with Deep Q-Learning and Graph Attention Networks," *IEEE Trans. Intell. Trans. Syst.*, vol. 23, no. 11, pp. 21102-21112, 2022

[21] Y. S.-Wook, *et al.* "GIN: Graph-based Interaction-aware Constraint Policy Optimization for Autonomous Driving," *IEEE Robo. and Autom. Lett.*, vol. 8, no. 2, pp. 464-471, 2022.