Risk-Constrained Hierarchical Reinforcement Learning for Safe Decision-Making and Control in Autonomous Driving

Jhan-Yu Liao, Chien-Wen Sun, Yu-Chen Lin, *Senior Member*, *IEEE*, Ting-Jia Yeh

Abstract—**Autonomous driving in complex dynamic traffic scenarios is challenging. The ego vehicle must navigate these conditions by engaging in complex interactions and negotiations with other road users, such as yielding, merging into traffic, taking turns, and changing lanes, to ensure both safe and efficient driving. In this paper, a framework for decision-making and control in self-driving by combining driving risk constraint and hierarchical reinforcement learning (HRL) is proposed, which is divided into three layers: (1) High-level: lane-changing decision and path planning, (2) Mid-level: path-following control decision, and (3) Low-level: risk-constrained emergency braking control decision. First, use bird's-eye view and graph attention methods with global semantic segmentation to observe. Then, decision-making and control in self-driving are implemented through a hierarchical dueling double DQN (D3QN) and worst case soft actor-critic (WCSAC), which includes path planning and path tracking. This paper utilizes the CARLA Simulator for test environment construction, simulation validation, and analysis. Two test scenarios, a roundabout intersection and a five-way intersection, were constructed, achieving self-driving success rates of 90.23% and 95%, respectively.**

I. Introduction

With the rapid development of Intelligent Transportation Systems (ITS) [1], autonomous driving technology has become a key innovation driving the progress of the transportation industry. A typical autonomous driving technology framework includes layers such as perception, communication, decision-making, and control, among which the decision-making and control layers are central to ensuring driving performance. The decision-making layer is responsible for path planning, including generating global paths based on environmental information and updating local paths in dynamic environments. The control layer implements precise tracking of planned paths through longitudinal and lateral control modules.

Path planning and tracking control are two fundamental components of autonomous driving systems. Traditional path planning methods [1]-[3] often rely on graph search and optimization techniques. For example, the Dijkstra algorithm is widely used for finding globally optimal paths in static environments, but its adaptability to dynamic environments is limited. Meanwhile, tracking control algorithms, including Pure-Pursuit (PP), Proportional-Integral-Derivative (PID) control, and Model Predictive Control (MPC), ensure accurate execution of planned trajectories. While these methods have been widely applied, they each have limitations, leading to the development of various improvements and extensions [4]-[6].

To address the challenges posed by dynamic and high-dimensional environments, researchers have increasingly explored intelligent control strategies that integrate Machine Learning (ML), Imitation Learning (IL), and Reinforcement Learning (RL) [7]-[14]. In [7], Liu *et al.* skillfully combined heuristic learning with RL to achieve high-speed highway lane-change decision-making. Studies [8]-[13] applied imitation learning to control decision-making, leveraging expert demonstrations to enhance performance and efficiency. In [14], RL was integrated with physical controllers to combine adaptability with robustness, addressing challenges in dynamic environments. While these approaches have achieved notable advancements, their standalone application often struggles in addressing the complexity of dynamic and high-dimensional environments, particularly when managing driving risks such as collision avoidance and dynamic interactions with other road users. Studies [18]-[19] adopted HRL, which has emerged as an effective strategy to address these challenges by dividing decision-making and control into multiple layers. This layered approach not only simplifies the learning process but also allows for the integration of driving risk constraints at different levels, ensuring safe and efficient decision-making and control in self-driving systems.

In this paper, we aim to develop a novel framework for decision-making and control in self-driving by integrating driving risk constraints with HRL. Inspired by studies [15]-[19], we first design a graph attention-based network utilizing semantic segmentation bird's-eye view (BEV) maps to process heterogeneous traffic information, enhancing adaptability across diverse scenarios. Building on this, we construct a hierarchical reinforcement learning (HRL) framework comprising three components-path planning, tracking control, and risk constraints-to simplify task complexity and enhance training efficiency. For path planning and risk constraint decision-making, which involve relatively discrete actions, we adopt Dueling Double Deep Q-Network (D3QN) to optimize decision-making efficiency. In contrast, for the precise and continuous control required for tracking,

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).

Jhan-Yu Liao, Yu-Chen Lin and Ting-Jia Yeh are with the Department of Automatic Control Engineering, Feng Chia University, Taichung 40724, Taiwan, R.O.C (e-mail: lcomebest@gmail.com, yuchlin@fcu.edu.tw, spring31212@yahoo.com.tw).

Chien-Wen Sun is with the Institute of Electrical and Control Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: cwsun.ee12@nycu.edu.tw).

we employ Worst Case Soft Actor-Critic (WCSAC) to ensure smooth and accurate maneuverability. Simulation verification and quantitative analysis are conducted using the CARLA simulator. Experimental results show that the proposed decision-making and control algorithm achieves success rates of 90.23% and 95.00% in roundabout and five-way intersection scenarios, respectively, demonstrating its effectiveness in complex traffic environments. The main contributions are summarized as follows:

1) **A risk-aware hierarchical reinforcement learning framework** for autonomous driving is proposed, integrating driving risk constraints into a three-layer architecture: high-level lane-changing and path planning, mid-level path-following control, and low-level emergency braking control. The framework further incorporates D3QN, WCSAC, and an Active Emergency Braking (AEB) mechanism to enhance driving safety and robustness.
2) **An HRL-based task decomposition strategy** is introduced to divide autonomous driving into multiple subtasks with distinct objectives, effectively alleviating reward sparsity and improving learning efficiency. Experimental results show a 3.32% improvement in success rate in roundabout scenarios compared with baseline methods.
3) **Extensive simulations** are conducted in the CARLA simulator to validate the proposed framework. The proposed method achieves success rates of 90.23% in roundabout scenarios and 95.00% in five-way intersection scenarios, demonstrating its effectiveness in complex traffic environments.

II. RELATED WORK

A. Semantic BEV Mask and Spatial Feature Extraction

Although autonomous driving technology has advanced significantly in recent years, urban driving environments remain challenging due to complex traffic dynamics and dense interactions among road users. Autonomous vehicles must accurately perceive surrounding environments and assess driving safety under diverse scenarios. To achieve human-like environmental perception, multiple cameras and semantic segmentation techniques are widely adopted. Semantic segmentation classifies each image pixel into categories such as vehicles, pedestrians, traffic signs, and road markings, enabling fine-grained scene understanding. Previous studies [15–17] demonstrated that deep learning and multi-sensor fusion-based semantic segmentation can effectively improve road scene understanding and obstacle perception in complex urban environments.

B. Hierarchical Autonomous Vehicle

With the increasing complexity of autonomous driving tasks, hierarchical architectures have emerged as an effective solution to enhance decision-making and control efficiency. Hierarchical autonomous driving approaches typically divide the driving task into multiple layers, improving task management and learning efficiency. Cao *et al.* [18] propose a hierarchical reinforcement and imitation learning (H-ReIL) approach, where the low-level policies, learned via imitation learning (IL), manage discrete driving modes, while the high-level policy, trained with RL, determines the switching between these modes. Zhu *et al.* [19] present a hierarchical deep reinforcement learning (DRL) framework with high sampling efficiency and strong sim-to-real transfer. The low-level DRL policy guides the robot toward the target while maintaining a safe distance from obstacles, whereas the high-level DRL policy further enhances navigation safety.

Building on these hierarchical approaches, our work further integrates a risk-aware mechanism into the decision-making process. Specifically, we incorporate a structured framework that jointly considers path planning, control, and risk constraints, enabling safer and more adaptive autonomous driving in complex environments with dynamic interactions and varying safety requirements.

C. Risk Constraint

Risk constraints are essential in autonomous driving to ensure safety by incorporating multiple levels of risk assessment into decision-making. In our framework, we introduce a two-layer risk constraint mechanism to enhance safety and stability. The first layer of risk constraint is imposed through the cost function in WCSAC, which explicitly penalizes unsafe actions and encourages safer behavior. The second layer of risk constraint is introduced at the high-level policy, following Zhu *et al.*[19]. This policy dynamically adjusts decision-making beyond immediate cost-based constraints by considering broader navigation safety. By incorporating high-level risk awareness, the system can proactively avoid hazardous situations that may not be captured solely by low-level cost penalties. By integrating these two levels of risk constraints, our approach enhances risk-aware decision-making, balancing local safety enforcement (via cost functions) and global safety optimization (via hierarchical policy adjustments).

III. PROPOSED SYSTEM

We develop a novel framework for decision-making and control in self-driving by integrating driving risk constraints with HRL, as depicted in Fig. 1. This architecture integrates spatial feature extraction and HRL, to enhance the efficiency and safety of decision-making and control in self-driving in urban traffic scenarios involving dense traffic flow and complex vehicle interactions with heterogeneous road layouts and uncertain surrounding behaviors while maintaining stable control performance under real-time operational constraints. The following sections detail each module within the architecture and describe the information flow between different hierarchical levels to clarify the overall system operation and the coordination among perception, decision, and control components.

A. Spatial Attention Feature Extraction

1) ResNet for Image Feature Extraction: In this architecture, the proposed method employs a global semantic segmentation BEV map to extract the current road semantic information around the autonomous vehicle, including drivable areas, lane line positions, and dynamic object information. The BEV representation provides a unified top-down spatial layout, which effectively reduces

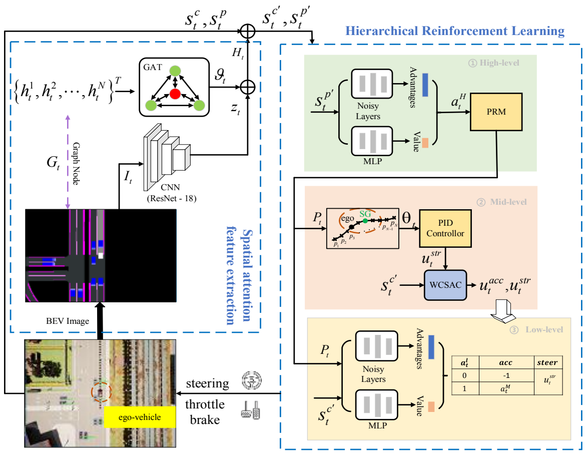


Figure 1. The architecture of HRL

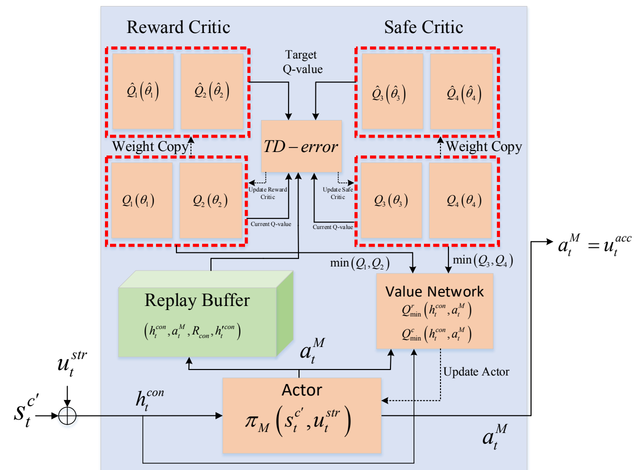


Figure 2. The architecture of WCSAC algorithm

perspective distortion and facilitates spatial reasoning in complex traffic scenarios by aligning visual perception with road-level semantics. Additionally, ResNet-18 [21] is utilized for image feature extraction to enhance the representation of road structures and surrounding environments. By leveraging deep convolutional layers, ResNet-18 captures multi-scale visual features such as lane boundaries, road geometry, and contextual cues from nearby vehicles. These extracted features preserve both local details and global semantic information, enabling robust perception under varying lighting and traffic conditions and improving adaptability to diverse urban environments with different road configurations. The resulting image feature embedding serves as a compact yet informative representation of the surrounding driving environment for subsequent interaction modeling and downstream decision-making modules.

2) GAT for Vehicle Interaction Modeling: Subsequently, a Graph Attention Network (GAT) is utilized to model the interaction relationships among dynamic objects with respect to the ego vehicle. The input to GAT consists of node features $h_t^i = [\Delta x_i, \Delta y_i, \ldots]^T$ denote surrounding vehicles and dynamic objects, where t represents the current time step, and $i = 1, \ldots, N$ denotes the index of each node, representing dynamic objects interacting with the ego vehicle. The node features include: relative world coordinates in the Cartesian coordinate system $(\Delta x_i, \Delta y_i)$, relative longitudinal distance Δs_i and lateral displacement d_i in the Frenet coordinate system, heading angle θ_i, velocity components (v_{x_i}, v_{y_i}), acceleration components (acc_{x_i}, acc_{y_i}), and vehicle indicator p_i which indicates whether the vehicle is in front of the ego vehicle.

B. Hierarchical Reinforcement Learning (HRL)

The path state s_t^p includes: the Frenet coordinates (s_t^{ego}, d_t^{ego}), the current vehicle speed v_t^{ego}, the current vehicle acceleration a_t^{ego}, the current vehicle yaw angle φ_t^{ego}, the left turn signal υ_t^1, the right turn signal υ_t^2, and the intersection signal υ_t^3. The control state is $s_t^c = [v_t^{ego}, d_t^{ego}, \theta_t^{ego}, s_t^{pre}, d_t^{pre}]^T$, where v_t^{ego} and d_t^{ego} are the same as those in s_t^p. In addition, θ_t^{ego} represents the steering angle required for the ego vehicle to reach the selected target point, whereas s_t^{pre} and d_t^{pre} denote the coordinates of the selected target point in the Frenet coordinate system.

1) High-level: lane-changing decision and path planning. At the high-level, a D3QN [22] is employed to determine whether the ego vehicle should perform a lane change under the current traffic conditions and interaction constraints with surrounding vehicles and road semantics. If a lane change is required, a cubic spline interpolation method is used to generate the vehicle's future trajectory, ensuring smooth, dynamically feasible, and comfortable transitions between lanes while respecting lane geometry and collision avoidance requirements.

2) Mid-level: path-following control decision. At the mid-level, the system generates path-following control commands based on the planned trajectory. WCSAC is an RL algorithm for continuous control that determines the vehicle's acceleration output based on the input data s_t^c and steering control signal u_t^{str}, as illustrated in Fig. 2. In addition, a PID controller is utilized to generate the steering commands, ensuring stable and accurate trajectory tracking under dynamic driving conditions. The loss function of actor is

$$L_{\pi_M}(\phi_M) = E_{h_t^{con} \sim D_c, a_t^M \sim \pi_M}[\beta \pi_{\phi_M}(a_t^M \mid h_t^{com}) - Q_{\min}^r(h_t^{con}, a_t^M) + \sigma Q_{\min}^c(h_t^{con}, a_t^M)] \tag{1}$$

where β denotes the entropy weight, which is introduced to enhance policy stochasticity and improve adaptability to diverse environments and tasks. The comprehensive state feature h_t^{com} is defined as the concatenation of the combined tracking control state s_t^c and the PID-generated steering command u_t^{str} to serve as the input for the actor network. D_c represents the replay buffer of the tracking control network, while σ is the safety weighting factor used to minimize safety-related losses. The loss function of critic is

$$L_{Q_n^r}\left(\theta_n^r\right)=\left(y_n^r-Q_n\left(h_t^r,a_t^r;\theta_n^r\right)\min(\rho_{\pi_M}(h_t^r,a_t^r),\sigma)\right)^2 \quad (2)$$

where ρ_{π_M} denotes the probability of selecting action a_t^r under policy π_M given the state h_t^r. This term can be regarded as an importance weight, with σ regulating the degree of emphasis assigned to critical constraints.

$$y_n^r = R_{con} + V_{\min} + V_{\max} \quad (3)$$

and R_{con} denotes the overall reward for tracking control. $V_{\min}$ and $V_{\max}$ are the minimum and maximum value functions, which represent the lowest and highest expected values obtained by selecting specific actions in the current state. Specifically, $V_{\min}$ and $V_{\max}$ are modulated by a threshold parameter $\delta\in[0,1]$ alongside the discount factor γ_c, where a higher δ encourages exploration and a lower δ promotes conservative driving.

3) Low-level: risk-constrained emergency braking control decision. At the low level, a risk-constrained AEB mechanism is introduced to handle high-risk situations. The AEB mechanism incorporates time to collision (TTC) estimation as a key factor in the reward function design, while a D3QN network is employed to generate braking commands. This mechanism ensures rapid response to potential collisions, improving vehicle safety in hazardous scenarios while avoiding unnecessary braking under normal driving conditions.

IV. EXPERIMENTS

A. Environment Settings and State space

The urban simulation scenario used in this paper is based on the "Town03" map from the open-source autonomous driving simulation software CARLA simulator. The state representation follows the definitions in Section III-B, where the path state s_t^p and control state s_t^c are concatenated with the spatial attention feature to form the to enhanced states $s_t^{p'}$ and $s_t^{c'}$. These updated states are then utilized as inputs for HRL. The final representations $s_t^{p'}$ and $s_t^{c'}$ include Frenet coordinates, velocity, orientation, and other key motion attributes, along with spatial attention information to better capture surrounding traffic context and dynamic interaction characteristics.

B. HRL Structure for Decision Making

The HRL framework in this study divides the decision-making process into three levels: high, mid and low, each responsible for different levels of decision tasks. The high-level module focuses on lane-changing decisions, specifically determining whether the ego vehicle should change lanes or remain in its current lane. The high-level action space consists of three options: lane keeping, left lane change, and right lane change. The decision from the lane-changing network is then passed to the Path Replanning Module (PRM) module. Following the network's output, a lateral shift is applied to the ego vehicle's path, starting from the next waypoint and ending at the final point of the current trajectory. The shift can be left (negative), right (positive), or none, with a fixed lateral displacement of 3.5 meters, corresponding to the lane centerline distance in the CARLA simulator. Finally, cubic spline interpolation and a vehicle dynamics-based curvature equation are used to refine the reference path. The mid-level focuses on path-following decisions, ensuring that the vehicle stays on the intended path while safely tracking it. Its action space can be represented as an acceleration set $a^M=\{-1\sim 1\}$, where values less than 0 correspond to braking, and values greater than 0 correspond to throttle input. Meanwhile, the steering command is computed by a PID controller based on the current vehicle position and the next target point. The low-level focuses on risk-constrained decisions, specifically determining how the ego vehicle should respond in unexpected situations. The low-level action space is defined as: $a^l=\{0;1\}$, where 0 represents an emergency requiring forced braking, and 1 indicates a normal condition, where the control command mapped from the mid-level is executed.

Finally, the control commands, including throttle, brake, and steering inputs, are fed into the autonomous vehicle, driving it to the next state. This process enables dynamic interaction between the vehicle and its environment, while ensuring safe driving considerations throughout the training of the entire algorithm under various traffic and risk conditions and maintaining stable closed-loop control behavior across different driving situations.

C. Rewards

The HRL architecture allows each sub-task network to be trained independently, enabling it to focus on learning the objectives specific to its assigned task. This improves both learning efficiency and overall performance while reducing interference between different control objectives. Therefore, we designed three distinct reward functions, corresponding to the high, mid, and low levels. During the training phase, these rewards function independently, with each reward signal exclusively used to update the neural network parameters of its respective layer. This hierarchical reward design also alleviates reward sparsity by providing dense and task-relevant feedback at different control layers, which is especially important for long-horizon urban driving scenarios. By decomposing complex driving behaviors into multiple reward-driven subproblems, each policy can receive more informative supervision signals during training. As a result, the learning process becomes more stable and sample-efficient, even when episodes terminate early due to safety violations or unexpected traffic interactions.

1) High-level reward

The high-level path reward is defined as

$$R_{path} = r_{lane} + r_{inv} + r_{lc} \quad (4)$$

where r_{lane} represents the base reward for the selected path, while r_{inv} is the reward function that evaluates the safety of the chosen path. $r_{lc} = w_D \cdot dis_{ahead}$ is the lane-changing incentive reward, designed to prevent the vehicle from persistently choosing the middle lane solely to maximize rewards. By incorporating the relative distance to the preceding vehicle, this term encourages more balanced lane usage and smoother traffic flow. dis_{ahead} denotes the relative distance between the preceding vehicle and the ego vehicle, with a weight of w_D.

2) Mid-level reward

The reward of WCSAC is defined as

$$R_{con} = r_{lon} + r_{lat} - J \tag{5}$$

where $r_{lon} = w_1 \cdot v_t^{ego}$ represents the longitudinal reward, as this paper aims to maximize the ego vehicle's speed under normal conditions. Meanwhile, $r_{lat} = w_2 \cdot d_t^{ego}$ represents the lateral reward, designed to ensure that the ego vehicle maintains its speed while minimizing deviations from the intended path.

To balance driving efficiency and safety, a cost function is introduced to constrain risky behaviors during path-following control. Define the WCSAC cost function

$$J \; = \; v_t^{ego}\left(dis_{safe} - dis_{ahead}\right) + w_3\left(h_{TTC} - TTC\right) \tag{6}$$

where TTC denotes the time-to-collision. dis_{safe} denotes the safe distance to the preceding vehicle. In this work, TTC is bounded by a predefined maximum value and computed based on the ego vehicle's speed to prevent excessive acceleration in potentially hazardous situations and to constrain risky behaviors during close vehicle interactions. h_{TTC} denotes the maximum human reaction time for collision avoidance, which is typically considered to be 2 seconds. To balance the priorities of different objectives, we assigned distinct weights $w_1 \sim w_3$ to each reward component according to their relative importance in safety and tracking performance.

3) Low-level reward

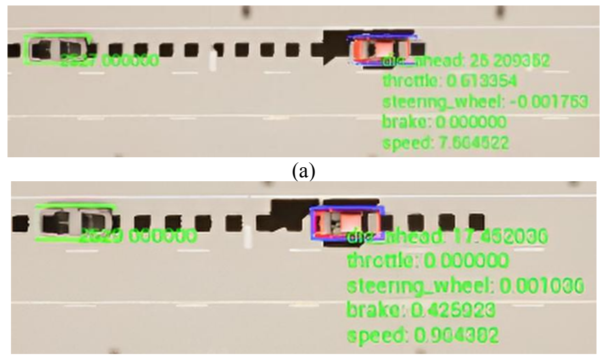
The reward of low-level is

$$\mathrm{R}_{safe} = -J^{\max}\mathbb{I}(C) + \mathrm{v}^{ego}(1 - \mathbb{I}(C)) \tag{7}$$

where C is a Boolean function indicating whether a collision has occurred. $\mathbb{I}(C)$ is Identity function. $J_{\max}$ represents the maximum value of the cost function. According to the design of this paper, the ego vehicle will decelerate in hazardous situations. However, under normal conditions, unnecessary braking is undesirable. Therefore, the reward function provides incentives based on the ego vehicle's speed during regular driving and applies a severe penalty only when a collision occurs. As a result, unnecessary emergency braking under normal driving conditions is avoided, preserving driving efficiency and ride comfort. In addition, this design encourages the ego vehicle to maintain smooth and stable driving behavior throughout the navigation process.

D. Results

Fig. 3 and Fig. 4 illustrate the vehicle's behavior during control tracking when encountering an obstacle on a straight path and in a roundabout, respectively. It can be observed that when the obstacle is too close or a potential collision is detected, the ego vehicle executes a deceleration maneuver to maintain a safe distance from the preceding vehicle. In both scenarios, the proposed controller adapts smoothly to the surrounding environment while preserving stable trajectory tracking. The scenario in Fig. 5 shows that the system determines that a lane-change maneuver can be executed on a straight path, where the ego vehicle performs a smooth lateral transition without abrupt steering or speed fluctuations.



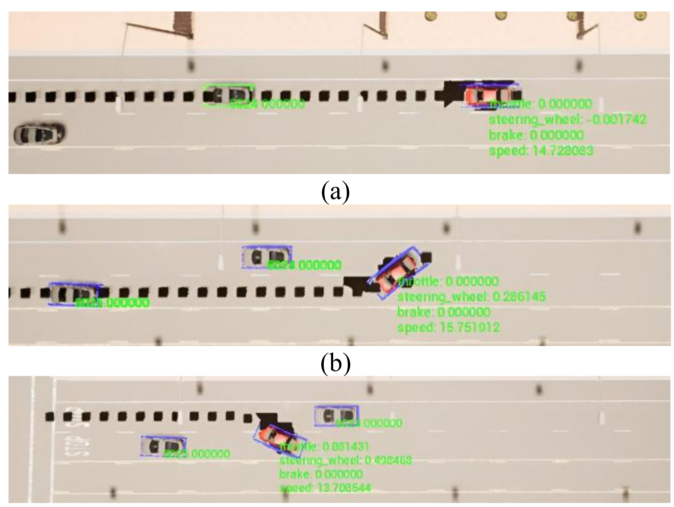
(a)

(b)

Figure 3. Control tracking when encountering an obstacle on a straight path


(a) (b)

Figure 4. Control tracking when encountering an obstacle in a roundabout


(a)

(b)

(c)

Figure 5. Control tracking of lane-change decision on a straight path

To comprehensively assess the performance of different models in autonomous driving scenarios involving urban maneuvers, we selected a commonly used metric for comparison:

Success Rate (%): The success rate denotes the proportion of successful path completions in urban traffic scenarios. A higher success rate indicates that the model performs more reliably and stably in complex urban environments with dynamic interactions and varying traffic densities. From Table I, it can be observed that our proposed model achieves the highest success rate in the roundabout scenario, outperforming all baseline methods. In contrast to Table I, the success rate of all models in Table II shows improvement. This is because the scenario starts with a single-lane design, which effectively reduces traffic complexity when vehicles enter the intersection, thereby lowering the risk of collisions. Although DQ-GAT achieves a higher success rate in the five-way intersection scenario, this study adopts a single-worker training approach, which requires relatively fewer computational resources. In contrast, DQ-GAT utilizes a multi-worker parallel training strategy, offering superior efficiency and sample utilization. However, such a framework requires scenario-specific adjustments to balance training efficiency and stability, whereas our objective is to achieve greater generalizability across diverse urban driving scenarios.

TABLE I.
QUANTITATIVE COMPARISON OF CONTROL STRATEGIES FOR THE EGO VEHICLE IN THE ROUNDABOUT SCENARIO

Model	Roundabout Success Rate (%)
IL-Aggr [16]	26.67
H-REIL [18]	28.33
IL-Conser [16]	55.67
DQ-GAT [16]	87.33
FSM-TTC[20]	87.67
Ours	**90.23**

TABLE II.
QUANTITATIVE COMPARISON OF CONTROL STRATEGIES FOR THE EGO VEHICLE IN THE FIVE-WAY INTERSECTION SCENARIO

Model	5-way Success Rate (%)
IL-Aggr [16]	83.00
H-REIL [18]	85.00
IL-Conser [16]	80.00
DQ-GAT [16]	99.67
FSM-TTC[20]	98.67
Ours	**95.00**

V. CONCLUSION

This paper proposes a hierarchical framework that encompasses high-level lane-change decision-making and path planning, mid-level path-following control, and low-level risk-constrained emergency braking control. By adopting a hierarchical task design, the proposed approach not only significantly improves learning efficiency but also provides a clear description of the algorithm's operational mechanism in autonomous driving tasks. Experimental results show the effectiveness of the proposed framework in achieving a high success rate in complex scenarios. In addition, the proposed framework demonstrates strong robustness and adaptability under diverse traffic environments and dynamic driving conditions, further highlighting its practical potential for real-world autonomous driving applications. In the future, we plan to implement the proposed algorithm in a real vehicle for validation. To achieve this, reducing the number of parameters and lowering the computational complexity of the model will be essential challenges that need to be addressed.

REFERENCES

[1] E. W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[2] C. K. The, W. K. Wong and T. S. Min, “Extended Dijkstra Algorithm in Path Planning for Vision Based Patrol Robot,” *Int. Conf. Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, Malaysia, pp. 184-189, Jun 2021.

[3] P. E. Hart, N. J. Nilsson and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Trans. Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, Jul 1968.

[4] F. Qing, *et al*., “An Improved Pure Pursuit Algorithm for Tractor Automatic Navigation”, *IEEE 4th Int. Conf. Information Systems and Computer Aided Education (ICISCAE)*, Dalian, China, Sep 2021.

[5] H. Zhong, *et al*., “Adaptive PID Controller Design Method based on PSO,” *Chinese Control and Decision Conf. (CCDC)*, Hefei, China, pp. 4501-4506, Aug 2020.

[6] G. Williams, *et al*., “Aggressive Driving with Model Predictive Path Integral Control,” *IEEE Int. Conf. Robotics and Automation (ICRA)*, Stockholm, Sweden, pp. 1433-1440 May 2016.

[7] T. Liu, *et al*., “Heuristics-oriented Overtaking Decision Making for Autonomous Vehicles Using Reinforcement Learning,” *Electrical Systems in Transportation (IET)*, pp. 417-424, 2020.

[8] L. Gao, *et al.*, “Learn to Design the Heuristics for Vehicle Routing Problem,” *arXiv:2002.08539*, pp. 1-10, Feb. 2020.

[9] P. Cai, *et al*., “Probabilistic End-to-End Vehicle Navigation in Complex Dynamic Environments with Multimodal Sensor Fusion,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 3, pp. 4218-4224, 2020

[10] D. Cheng, *et al*., “Learning by Cheating,” *Conf. Robot Learning (CoRL)*, London, USA, pp. 66-75, 2020.

[11] E. O.-Bar, *et al*., “Learning Situational Driving,” *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 11293-11302, 2020.

[12] F. Codevilla, E. Santana, A. Lopez, and A. Gaidon, “Exploring the Limitations of Behavior Cloning for Autonomous Driving,” *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Long, Beach, CA, USA, pp. 9329-9338, 2019.

[13] P. Cai, *et al*., “Vision-Based Autonomous Car Racing Using Deep Imitative Reinforcement Learning,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 4, pp. 7262-7269, 2021.

[14] G. Williams, *et al*., “Information Theoretic MPC for Model-Based Reinforcement Learning,” *IEEE Int. Conf. Robotics and Automation (ICRA)*, Singapore, pp. 1714-1421, May 2017.

[15] S. Teng, *et al*., “Hierarchical Interpretable Imitation Learning for End-to-End Autonomous Driving,” *IEEE Trans. Intelligent Vehicles*, vol. 8, pp. 673-683, November 2022.

[16] 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 Trans. Intelligent Transportation Systems (T-ITS)*, vol. 23, no. 11, pp. 21102-21112, 2022.

[17] Z. Zhang, *et al*., “End-to-End Urban Driving by Imitating a Reinforcement Learning Coach,” *International Conf. Computer Vision (ICCV)*, Montreal Canada, pp. 15202-15212, 2021.

[18] Z. Cao, *et al*., “Reinforcement Learning based Control of Imitative Policies for Near-Accident Driving,” *arXiv:2007.00178*, 2020.

[19] W. Zhu and M. Hayashibe, “A Hierarchical Deep Reinforcement Learning Framework with High Efficiency and Generalization for Fast and Safe Navigation,” *IEEE Trans. Industrial Electronics*, vol. 70, no. 5, pp. 4962-4971, May 2023.

[20] A. Cosgun, *et al*., “Towards Full Automated Drive in Urban Environments: A Demonstration in GoMentum Station, California,” *IEEE Intelligent Vehicles Symposium (IV)*, Los Angeles, CA, USA, 2017, pp. 1811-1818.

[21] X. Ou, *et al*., "Moving Object Detection Method via ResNet-18 With Encoder–Decoder Structure in Complex Scenes," *IEEE Access*, vol. 7, pp. 108152-108160, 2019.

[22] K. Arulkumaran, *et al*., "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26-38, Nov. 2017,