A Data-Driven Control Framework Using Deep Reinforcement Learning for Autonomous Driving

Mei-Lin Huang*, Ching-Hung Lee*, Cheng-Ting Huang**, and Hsin-Han Chiang**
*National Yang Ming Chiao Tung University, Hsinchu City, Taiwan
**National Taipei University of Technology, Taipei, Taiwan
E-mail: adnil.huang.ee11@nycu.edu.tw; chl@nycu.edu.tw; t110448045@ntut.edu.tw; hsinhan@mail.ntut.edu.tw

***Abstract*—This paper presents a data-driven control framework for autonomous driving based on deep reinforcement learning (DRL). The proposed framework addresses lane-following and car-following using a dual-actor DDPG architecture with task-specific rewards. The proposed approach improves safety, efficiency, and robustness, as demonstrated through extensive simulation, hardware-in-the-loop (HIL), and real-world testing. Comparative results show significant advantages over adaptive model predictive control (AMPC), highlighting its potential for real-world autonomous vehicle deployment.**

I. INTRODUCTION

Recent advances in autonomous driving technologies have significantly increased the demand for robust, adaptable, and data-driven vehicle control systems. Autonomous vehicles must handle lateral and longitudinal control in real time, especially in complex and dynamic traffic environments. Traditional approaches such as H_∞ control [1] and Model Predictive Control (MPC) [2][3] have demonstrated effectiveness in structured scenarios. However, their reliance on precise vehicle dynamic models and computational intensity limits their applicability in real-world and complex driving conditions.

In recent years, deep learning-based methods have gained significant traction in autonomous driving, particularly through end-to-end supervised learning frameworks that directly map raw sensory inputs to driving control commands [4][5]. While such approaches excel in extracting complex features from high-dimensional data, they are often criticized for their lack of interpretability, heavy reliance on large labeled datasets, and challenges in generalizing to unseen scenarios. As an alternative, deep reinforcement learning (DRL) has emerged as a powerful paradigm for autonomous driving, enabling agents to learn optimal control policies via continuous interaction with the environment and the use of well-designed reward functions [6][7][8]. Notably, DRL facilitates adaptive decision-making under uncertain conditions, addressing some limitations of purely supervised approaches. However, the majority of existing DRL-based solutions are designed for isolated tasks such as lane-following (LF) or car-following (CF), which [1] restricts their scalability and modularity. Moreover, these methods can suffer from unstable training dynamics, particularly in continuous action spaces and complex real-world environments [9].

To overcome the aforementioned limitations, this paper presents a deep reinforcement learning (DRL)-based control framework that emphasizes interpretable reward design to ensure safety and stability. The proposed architecture explicitly decomposes vehicle control into distinct steering and speed agents, thereby improving learning efficiency, modularity, and interpretability. Training is further accelerated and generalized by incorporating physical priors. Safety constraints are integrated directly into the DRL model, thereby enhancing the reliability and operational robustness of real-time control. Extensive evaluations are conducted in CarSim simulation, hardware-in-the-loop (HIL), and real-vehicle environments. By bridging the gap between modular task design and real-world deployment, this study contributes a scalable and interpretable DRL-based control strategy that advances the safety, comfort, and performance of autonomous vehicles.

II. MODEL DESIGN

This work proposes a modular DRL framework that separates CF and LF control into two dedicated agents. Each agent is independently trained with the reinforcement learning process, which utilizes task-specific states, actions, and rewards to enhance training stability and interpretability.

A. Speed Agent

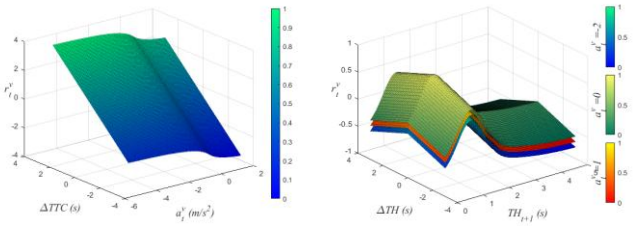
1) State and Action Spaces

The state space s_t^v includes the relative distance $d_{rel,t}$ (m) between the ego vehicle and the preceding vehicle, the relative velocity $v_{rel,t}$ of the preceding vehicle with respect to the ego vehicle, velocity of the ego vehicle v_t, as higher speeds necessitate larger following distances to ensure adequate braking time and overall safety. These features enable responsive and safe adaptation to dynamic traffic conditions. The action space consists of a bounded and smooth acceleration a_t^v to account for both safety and passenger comfort.

2) Reward Formulation

This work is supported by NSTC under Grant 113-2221-E-027-063-MY2.

979-8-3315-4206-8/25/$31.00 ©2025 IEEE



(a) (b)

Fig. 1. Illustration of r_t^v (a) under critical condition ($TTC_t < 2s$) and (b) under regular driving condition ($TTC_t \geq 2s$).

The reward function r_t^v aims to strike a balance between safety and driving efficiency using two temporal indicators, i.e., time headway (TH) and time-to-collision (TTC). A specific reward scheme is applied when a potential collision is imminent. In particular, once the TTC falls below a critical threshold (e.g., $TTC_t < 2s$), a continuous and bounded penalization (1) is employed to discourage unsafe behavior.

$$r_t^v = \Delta TTC - \frac{a_t^v}{1 + |a_t^v|} \tag{1}$$

where the temporal difference $\Delta TTC = TTC_{t+1} - TTC_t$ is physically constrained in simulation or environment dynamics, $\Delta TTC \in [\Delta TTC_{min}, \Delta TTC_{max}]$, and the second term is bounded within $[-1, 1]$. As illustrated in Fig. 1(a), the full reward under the high-risk condition remains bounded, $r_t^v \in [\Delta TTC_{min} - 1, \Delta TTC_{max} + 1]$. By saturating extreme acceleration values, the non-linear transformation stabilizes reward signals and mitigates gradient-related instabilities in DRL.

During regular driving conditions ($TTC_t \geq 2s$), the reward is decomposed into two interpretable rewards r_1 and r_2. r_1 utilizes a dual-Gaussian shaping function centered around the desired TH, TH_{des}, providing a dense and smooth reward gradient near the optimal range. As shown in (2), r_1 is analytically designed and bounded within $\left[-r_{th}, 1/\sqrt{2\pi{\sigma_1}^2} + 1/\sqrt{2\pi{\sigma_2}^2} - r_{th}\right]$, where $r_{th} \in \mathbb{R}^+$.

$$r_1 = \sum_{i=1}^{2} \frac{1}{\sqrt{2\pi{\sigma_i}^2}} exp\left(-\frac{(TH_{t+1} - TH_{des})^2}{2{\sigma_i}^2}\right) - r_{th} \tag{2}$$

Subsequently, r_2 represents the penalties over TH variation and action magnitude as given by:

$$r_2 = -w_1 \times |\Delta TH| - w_2 \times |a_t^v| \tag{3}$$

Assuming bounded physical limits $|\Delta TH| \leq \Delta TH_{max}$ and $|a_t^v| \leq a_{max}$, r_2 is bounded within $[-(w_1 \times |\Delta TH| + w_2 \times |a_t^v|), 0]$. This overall boundedness and smoothness of the reward function, $r_t^v = r_1 + r_2$, ensures that gradient signals for both the networks remain well-behaved throughout training, as shown in Fig. 1(b).

B. Steering Agent

1) State and Action Spaces

The state vector s_t^{sw} consists of spatial, dynamic, and preview-based features designed to support robust lateral control. Key elements include lateral deviation l_t and heading angle error ψ_t, which quantify tracking accuracy and orientation alignment. Ego vehicle speed v_t is included due to its impact on lateral dynamics. To enhance anticipatory control, three preview points d_t^i (i = 1, 2, 3) are extracted in the vehicle's local coordinate frame, with their temporal variations $\Delta d_i = d_t^i - d_{t-1}^i$ capturing curvature transitions along the path. Additionally, longitudinal acceleration $a_{x,t}$ and yaw rate change $\Delta\omega_t$ are accounted for transient dynamics affecting lateral response. Collectively, the features $\{l_t, \psi_t, v_t, \Delta d_1, \Delta d_2, \Delta d_3, a_{x,t}, \Delta\omega_t\}$ provide a rich, context-aware state representation for robust and adaptive steering control. The action space a_t^{sw} is the steering wheel angle bounded by the physical limits of the steering system within $a_t^{sw} \in [\delta_{min}, \delta_{max}]$.

2) Reward Formulation

The steering reward function $r_t^{sw} = r_l + r_d$ comprises two components, namely, the lateral deviation reward r_l and deviation trend reward r_d. r_l penalizes deviation from the lane center, defined as a non-linear function of future deviation l_{t+1}, with hyperparameters k_1, k_2, and k_3 controlling sensitivity, as expressed by

$$r_l = \frac{k_1}{k_2 + |l_{t+1}|} - k_3 \tag{4}$$

This formulation ensures that rewards are maximized when the vehicle closely follows the lane center, while progressively penalizing larger deviations. In addition to maintaining alignment with the lane center, human-like corrective behavior is encouraged by promoting convergence toward the centerline. Thus, the deviation trend reward r_d is designed to be dependent on whether the lateral deviation is decreasing or increasing. As formulated in (5), when the vehicle is converging toward the centerline, i.e., $\Delta l = |l_{t+1}| - |l_t| < 0$.

$$r_d = \frac{-\Delta l}{\max(\eta, |\psi_{t+1}|)} \tag{5}$$

$\eta > 0$ is a small constant introduced to avoid division by zero. Conversely, if the vehicle diverges from the centerline ($\Delta l > 0$), the reward uses (6) to penalize both increased deviation and poor orientation.

$$r_d = -\Delta l \times |\psi_{t+1}| \tag{6}$$

To confront the robustness in real vehicle actuation conditions with steering mechanism delay, a compensatory term ϵ is incorporated into the reward function. When the lateral acceleration indicates a corrective steering effort, but the resulting lateral deviation l_{t+1} has yet to reflect improvement due to inertial or latency, the reward is adjusted as:

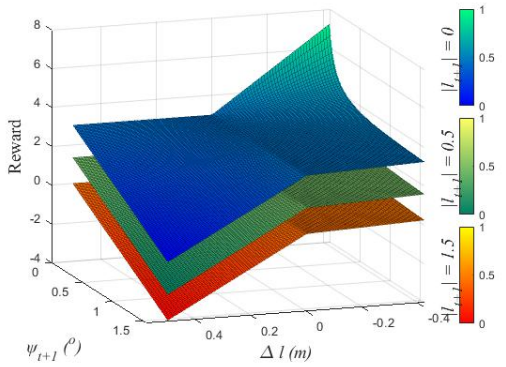


Fig. 2. Illustration of r_t^{sw}.

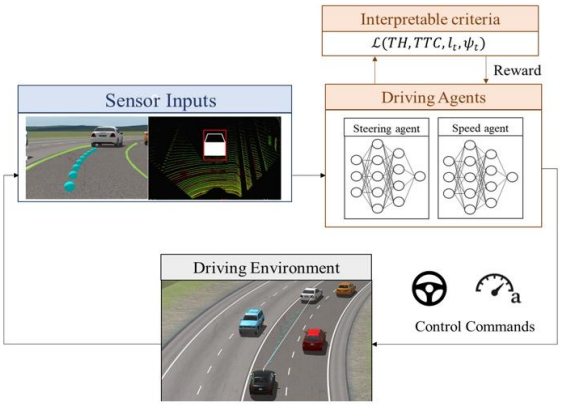


Fig. 3. Block diagram of the proposed modular DRL architecture for autonomous driving agents.

$$r_t^{sw} \leftarrow r_t^{sw} + \epsilon \tag{7}$$

This mechanism acknowledges delayed responses and reinforces beneficial actions even before observable outcomes.

The complete reward formulation offers several advantages: 1) the nonlinear shaping of r_l promotes precise tracking, 2) r_d captures behavioral correction trends for smoother recovery, and 3) the inclusion of ϵ supports learning stability in the presence of system delays and uncertainties inherent to real-world driving. Fig. 2 illustrates the overall reward function r_t^{sw}. Each surface corresponds to a distinct level of lateral deviation magnitude $|l_{t+1}|$, showing how the total reward varies under different driving conditions. The reward increases when $\Delta l < 0$ and maintains small ψ_{t+1}, indicating effective and smooth correction toward the lane center. In contrast, large values of both $\Delta l > 0$ and $|\psi_{t+1}|$ signify divergence with misalignment, resulting in significant penalties.

C. DRL Training Architecture

The proposed control framework, as illustrated in Fig. 3, adopts a modular DRL architecture with two dedicated agents for CF and LF control. Sensor inputs, including camera and radar data, are processed and supplied to each agent, which operates independently using the DDPG algorithm. Both agents employ an actor-critic structure, with separate neural networks for the actor and critic. The critic network is trained by minimizing the Bellman error between predicted and target Q-values, while the actor network updates its policy to maximize the expected Q-value as evaluated by the critic. The off-policy nature of DDPG enables efficient exploration through the use of a replay buffer, while target networks are updated via a soft update mechanism to stabilize training. Interpretable reward criteria, as introduced in the previous section, are integrated to guide learning towards safe and efficient driving behavior. The resulting control commands for steering and speed are executed in the simulated or real driving environment, closing the control loop as depicted in the block diagram. This dual-agent design supports continuous control and coordinated learning for autonomous driving with complex decision making.

III. Simulation Setup

Simulation experiments were performed in CarSim, a widely recognized vehicle dynamics simulator, to evaluate the proposed control algorithms across diverse driving scenarios.

A. Implementation Details

The experience replay buffer size M was set to 10,000, with a mini-batch size $M_{mini} = 100$ for training. Initially, the network remains untrained while the buffer is populated with diverse transitions. Data accumulation halts if the cross-track error exceeds 2m (on a 3.5m wide road), the ego vehicle collides with the preceding vehicle, or the lead vehicle exits the sensor range. In such cases, the ego vehicle is reset to the centerline, and data collection resumes until the buffer exceeds M_{mini}. For steering, episodes continue despite deviation to allow learning of recovery behaviors. For speed control, episodes terminate upon collision or loss of the lead vehicle. To accelerate early training, the agents are initially trained separately: the steering agent operates under constant speed, while the speed agent is trained on a straight path. Once both agents can complete episodes reliably, joint training is performed. The training process employs the DDPG algorithm with tailored hyperparameters for learning rate, reward shaping, and temporal safety margins, as detailed in Table I.

TABLE I. Parameters Adopted in The DRL Algorithm

Parameters	Meaning	Value
γ	Reward discount factor	0.9
τ	Soft replacement factor	0.1
w_1	Ratio for TH variation of speed model reward	0.1
w_2	Ratio for action magnitude of speed model reward	0.1
σ_1	Variance of the first normal distribution of speed reward	0.3
σ_2	Variance of the second normal distribution of speed reward	1.5
TH_{des}	TH that returns the maximum reward	1.5
k_1, k_2, k_3	Tunable parameters for the reward function of the steering agent	5, 1, 2
ϵ	Small positive reward for compensating delayed but corrective steering response	0.01
η	A small positive constant ensuring bounded denominator when $\|\psi_{t+1}\|$ approaches zero.	0.01

TABLE II. TRAINING AND TESTING ROAD SPECIFICATION

Training			Testing		
Direction	**Length**	**Radius**	**Direction**	**Length**	**Radius**
Ahead	50m	-	Ahead	150m	-
Right	300m	150m	Right	300m	150m
Ahead	50m	-	Ahead	50m	-
Left	300m	150m	Left	300m	150m
Ahead	50m	-	Ahead	50m	-
-	-	-	Right	250m	200m
-	-	-	Ahead	80m	-
-	-	-	Left	250m	200m
-	-	-	Ahead	100m	-

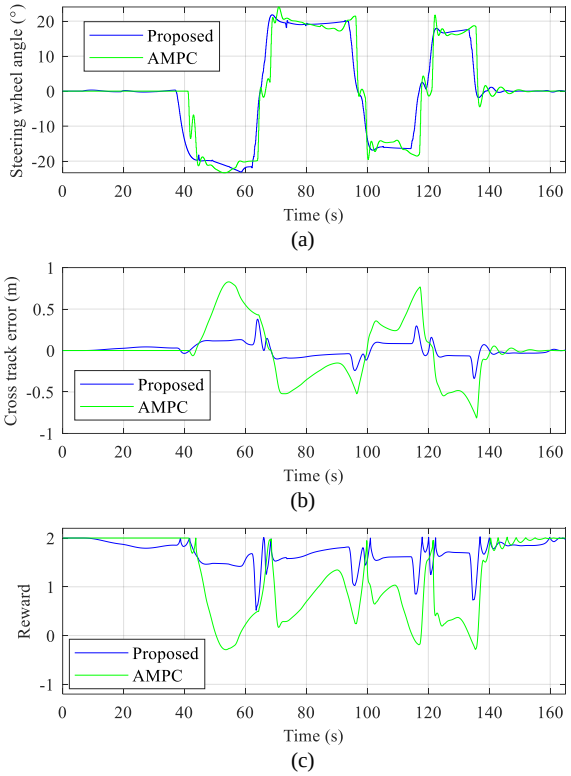


Fig. 4. Comparison of LF performance between AMPC and the proposed method in simulation. (a) Steering wheel angle response. (b) Cross-track error. (c) r_t^{sw} during the test.

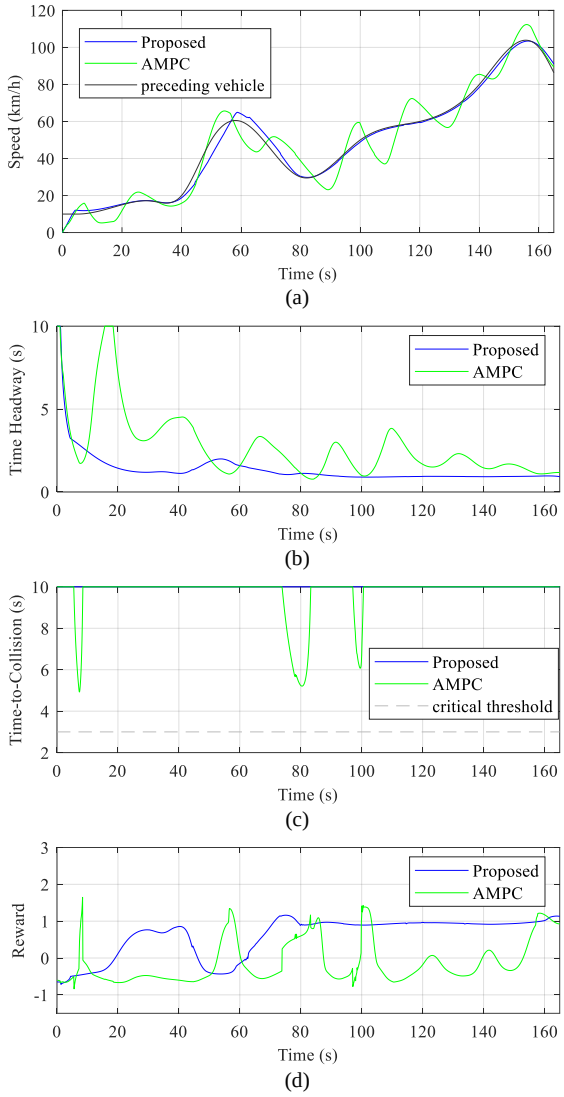


Fig. 5. Comparison of CF performance between AMPC and the proposed method in simulation. (a) Velocity of ego vehicle and the preceding vehicle. (b) TH and (c) TTC performance. (d) r_t^v during the test.

B. *Evaluation Results*

A comparative evaluation between the proposed control strategy and AMPC [10] is conducted on a 1530m test track featuring both left and right curves. Road specifications are detailed in Table II. To enhance scenario diversity, the preceding vehicle's speed was varied between 10-100 km/h, with acceleration rates from -1.5 to 1 m/s^2. As shown in Fig. 4(a), the integration of the deviation trend reward r_d enables the proposed model to produce smoother and more stable steering responses compared to AMPC, especially during sharp turns and on untrained paths. During critical intervals (40–60s and 120–140s), our approach demonstrates superior trajectory adaptation with minimal overcorrection. From the observation of cross-track error in Fig. 4(b), the proposed method consistently maintains deviation within ± 0.5m, outperforming AMPC with deviation exceeds ± 1m in continuously curved road segments. Additionally, in Fig. 4 (c), reward r_t^{sw} for our approach shows higher and more stable average values, indicating improved consistency and robustness. Such results confirm that the proposed method achieves precise path-following and stable performance, even on high-curvature roads.

Fig. 5 demonstrates the superior performance of the proposed CF strategy in dynamic driving scenarios. Despite fluctuations in the preceding vehicle's acceleration, as shown in Figs. 5(a) and 5(b), the model consistently maintains a stable time TH near 0.9s and keeps the TTC above 2s. This indicates not only safety but also adaptability in various conditions. In contrast, AMPC exhibits frequent oscillations and unsafe drops below critical thresholds. Moreover, Fig. 5(c) confirms that the proposed agent sustains a TTC greater than 2s throughout the testing period, while AMPC often approaches a dangerously low 3-second margin, significantly increasing collision risk. Additionally, Fig. 5(d) shows that our model achieves smoother and higher cumulative rewards, particularly during the early

and mid-episodes (20–80s), unlike the erratic performance observed with AMPC. These advancements are driven by the reward design r_t^v, which incorporates the temporal safety metrics TTC and TH through their exponents, enabling the agent to learn anticipatory and stable speed control policies. Due to a lack of such temporal sensitivity, AMPC yields erratic and less safe behaviors. Overall, the proposed CF strategy demonstrates superior safety, stability, and reward consistency in dynamic driving environments.

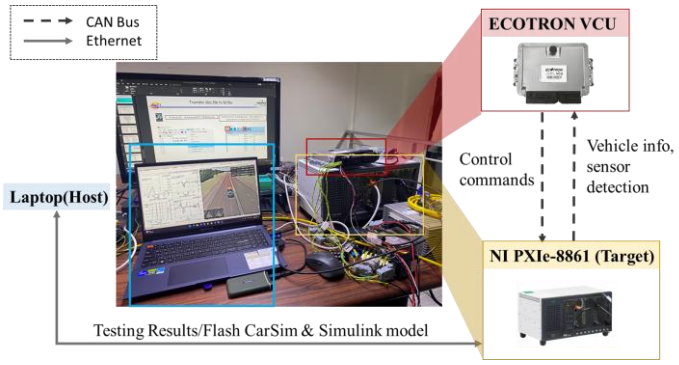


Fig. 6. Hardware architecture of HIL simulation.

IV. Test With HIL System

A. Hardware Setup

As shown in Fig. 6, the HIL simulation system includes a host laptop, an NI PXIe-8861 real-time target, and an Ecotron VCU. The laptop sets up the NI-RT framework and NI VeriStand project, deploying CarSim-RT and Simulink-based control models through Ethernet to the PXIe-8861 for real-time operation. The PXIe communicates with the VCU via a CAN bus, allowing real-time exchange of sensor data and control signals. The VCU simulates vehicle actuators—steering, throttle, and brake—creating a closed loop with the PXIe to replicate real vehicle responses. Although the CAN interface introduces minor resolution-related inaccuracies, it effectively mimics real-world signal transmission conditions.

B. Testing Results

Compared to the simulation response in Fig. 4(a), the HIL results in Fig. 7(a) display slight oscillations and signal jitter in steering control, particularly between 40–60s, due to CAN bus latency and resolution limits. Despite these effects, the steering trajectory remains aligned with the desired path, and the cross-track error in Fig. 7(b) stays within ± 0.5, confirming reliable lateral control under real-time constraints. The observed response delays highlight practical limitations of physical systems, such as communication bottlenecks and reduced signal fidelity. For CF operation, Fig. 8 demonstrates the robustness of longitudinal control. The ego vehicle effectively tracks speed changes of the preceding vehicle (Fig. 8(a)), while maintaining a safe TH of 1s (Fig. 8(b)) and TTC consistently exceeding 10s, which are capped at 10s in Fig. 8(c) for visualization purposes. Although responses are less smooth than in SIL, the HIL setup successfully emulates real-world dynamics, validating the proposed driving agent performance under hardware limitations.

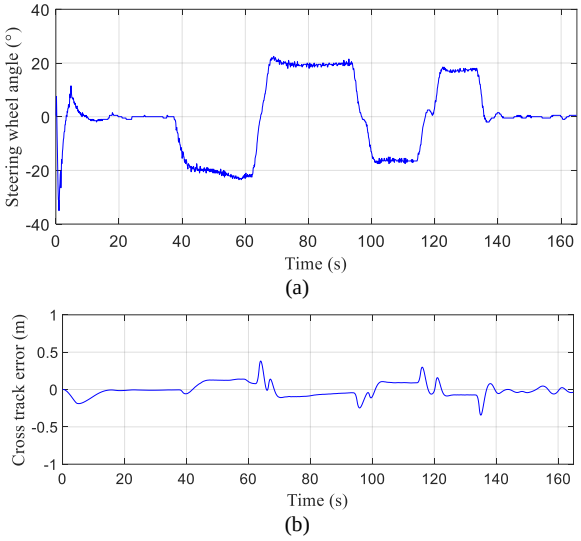


Fig. 7. Testing result of LF in HIL simulation. (a) Steering wheel angle response. (b) Cross-track error.

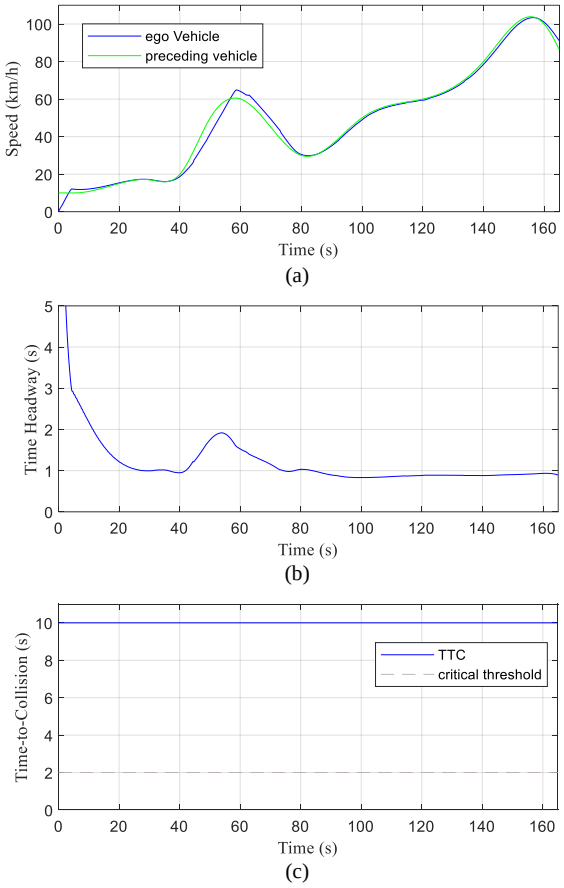


Fig. 8. Testing result of CF in HIL simulation (a) Velocity of the host car and the preceding car. (b) TH and (c) TTC performance.

V. On-Road Experiments

To assess the generalization and real-time performance of the proposed control system, experiments were conducted on a full-scale Toyota Corolla Cross, as shown in Fig. 9. The vehicle

is equipped with a camera, millimeter-wave radar, prototyping VCU, and away. Sensor fusion provided relative speed and distance to the preceding vehicle. Real-time vehicle states were acquired through the CAN bus, with model inference executed at a 20Hz sampling rate to meet stringent real-time requirements. The experiments are carried out on an expressway under actual vehicular traffic conditions, providing representative real-world evaluations.

Fig. 9. The testbed with equipped sensors.

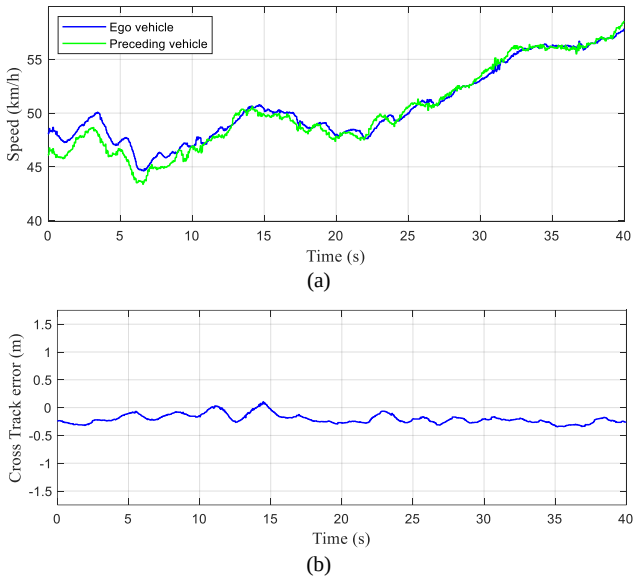


Fig. 10. Testing result of CF and LF on the testbed under real road conditions. (a) Velocity of the host car and the preceding car. (b) Cross-track error.

Fig. 10 summarizes the experimental performance. As shown in Fig. 10(a), the ego vehicle maintains a stable TH of approximately 0.9s and consistently safe TTC while tracking the preceding vehicle. In Fig. 10(b), the cross-track error obtained from camera-based lane detection indicates that the lateral deviation remains within ±0.35m, with minor steering oscillations caused by actuation delay in the real vehicle. These results demonstrate the generalization capability and real-time operational feasibility for simultaneous LF and CF under real-world conditions.

VI. Conclusion

This paper introduces a DRL framework for autonomous driving that includes task-specific agent decomposition, interpretable reward design, and integrated safety constraints. Through extensive experiments in simulation, HIL simulation, and real-world testing, the proposed method shows notable improvements in tracking accuracy, stability, and robustness compared to typical AMPC. The explicit separation of speed and steering agents, combined with physical priors, speeds up policy convergence and improves generalization to unseen driving conditions. These findings demonstrate the practical viability of the data-driven control framework for autonomous vehicles, advancing both safety and operational performance. Future work investigates more complex traffic scenarios, multi-agent interactions, and real-time adaptation to diverse and dynamic environments.

References

[1] F. Rekabi, F. A. Shirazi, M. J. Sadigh, et al., "Distributed output feedback nonlinear H_∞ formation control algorithm for heterogeneous aerial robotic teams," *Robotics and Autonomous Systems*, vol. 136, p. 103689, 2021.

[2] D. Jeong and S. B. Choi, "Efficient trajectory planning for autonomous vehicles using quadratic programming with weak duality," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 2878-2892, 2024.

[3] X. Sun, Y. Zhang, Z. He, Y. Liu, and K. Li, "Integrated path planning and tracking control for autonomous vehicles: A review of the state of the art and future perspectives," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 3786–3805, 2023.

[4] X. Hu, B. Tang, L. Chen, S. Song, and X. Tong, "Learning a deep cascaded neural network for multiple motion commands prediction in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7585-7596, Dec. 2021.

[5] S.-H. Chung, S.-H. Kong, S. Cho, and I. M. A. Nahrendra, "Segmented encoding for sim2real of RL-based end-to-end autonomous driving," *2022 IEEE Intelligent Vehicles Symposium (IV)*, Aachen, Germany, 2022, pp. 1290-1296.

[6] Y. Tian, X. Cao, K. Huang, C. Fei, Z. Zheng, and X. Ji, "Learning to drive like human beings: A method based on deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6357-6367, July 2022.

[7] Y. Du, J. Chen, C. Zhao, F. Liao, and M. Zhu, "A hierarchical framework for improving ride comfort of autonomous vehicles via deep reinforcement learning with external knowledge," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, pp. 1059–1078, 2023.

[8] C.-J. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine, and M. J. Kochenderfer, "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving," in *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 294-305, June 2020.

[9] T. P. Lillicrap, et al., "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[10] MathWorks, "Adaptive MPC," MathWorks Documentation. Available: https://www.mathworks.com/help/mpc/ug/adaptive-mpc. html. [Accessed: Jan. 25, 2025].